THE UNIVERSITY OF CHICAGO

DEEP APPROXIMATE BAYESIAN INFERENCE

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE UNIVERSITY OF CHICAGO
BOOTH SCHOOL OF BUSINESS
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

BY
YUEXI WANG

CHICAGO, ILLINOIS
JUNE 2023

Dedicated to my parents, Keqing and Zhigang.

Stay Hungry, Stay Foolish, Stay Curious.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES ix

ACKNOWLEDGMENTS x

ABSTRACT xii

1 INTRODUCTION 1
1.1 Approximate Bayesian Computation 1
1.2 Bayesian Deep Learning 3
1.3 Variable Selection 5

2 ABC VIA CLASSIFICATION 6
2.1 Introduction 6
2.2 ABC without Summary Statistics 10
2.2.1 ABC with KL Divergence 11
2.2.2 Connection to Robust Bayesian Inference 15
2.2.3 Estimating KL Divergence via Classification 16
2.2.4 Other KL Estimators 17
2.3 Frequentist Analysis of the ABC Posterior 19
2.3.1 Convergence Rate of Estimation Errors 20
2.3.2 Posterior Concentration Rate 24
2.3.3 Shape of the Limiting ABC Posterior Distribution 27
2.3.4 ABC Kernels and Model Misspecification 29
2.4 Simulations 30
2.4.1 M/G/1-Queuing Model 32
2.4.2 Lotka-Volterra Model 34
2.5 Empirical Analysis 36
2.5.1 Synthetic Data 39
2.5.2 Real Data Analysis 40
2.6 Discussion 42
2.7 Appendix 43
2.7.1 Convergence Rate of Estimation Errors with Neural Network Sieves . 43
2.7.2 Frequentist's Analysis on the Exponential Kernel 47
2.7.3 Proofs 49

3 ADVERSARIAL BAYESIAN SIMULATION 60
3.1 ABC and Beyond 60
3.2 Adversarial Bayes 64
3.2.1 Vanilla GANs 65
3.2.2 Conditional GANs 66

3.2.3 Bayesian GANs . . . 69
3.2.4 Two-step Refinement . . . 73
3.3 Adversarial Variational Bayes . . . 77
3.4 Theory . . . 81
3.5 Performance Evaluation . . . 86
3.5.1 Lotka-Volterra Model . . . 86
3.5.2 Simple Recruitment, Boom and Bust . . . 88
3.6 Discussion . . . 91
3.7 Appendix . . . 92
3.7.1 Proofs from Section 3.4 . . . 92

4 UNCERTAINTY QUANTIFICATION FOR SPARSE DEEP LEARNING . . . 106
4.1 Introduction . . . 106
4.2 Deep ReLU Networks . . . 111
4.2.1 Spike-and-Slab Priors . . . 112
4.2.2 A Connection between Deep ReLUs and Trees . . . 114
4.2.3 Posterior Concentration . . . 117
4.3 Semi-parametric BvM's . . . 118
4.3.1 Linear Functionals . . . 120
4.3.2 Squared L^2-norm Functional . . . 122
4.4 Adaptive Priors . . . 122
4.5 Discussion . . . 125
4.6 Appendix . . . 126
4.6.1 Rudiments . . . 126
4.6.2 Posterior Concentration Rate . . . 127
4.6.3 Preparations for Main Theorems . . . 135
4.6.4 Proof of Theorem 29 . . . 137
4.6.5 Proof of Theorem 31 . . . 139
4.6.6 Proof of Theorem 32 . . . 145

5 DATA AUGMENTATION FOR BAYESIAN DEEP LEARNING . . . 149
5.1 Introduction . . . 149
5.2 Bayesian Deep Learning . . . 153
5.2.1 Bayesian Simulation and Regularization Duality . . . 154
5.2.2 A Stochastic Top Layer . . . 156
5.3 Data Augmentation for Deep Learning . . . 157
5.3.1 MCMC with J-copies . . . 162
5.3.2 Connection to Diffusion Theory . . . 164
5.4 Applications . . . 166
5.4.1 Gaussian Regression . . . 167
5.4.2 Support Vector Machines (SVMs) . . . 169
5.4.3 Logistic Regression . . . 170
5.5 Experiments . . . 172
5.5.1 Friedman Data . . . 174

5.5.2 Boston Housing Data . . . 175
5.5.3 Wine Quality Data Set . . . 177
5.5.4 Airbnb Data Set . . . 179
5.5.5 Summary of Experiment Results . . . 180
5.6 Discussion . . . 181

6 VARIABLE SELECTION WITH ABC BAYESIAN FORESTS . . . 184
6.1 Perspectives on Non-parametric Variable Selection . . . 185
6.2 Bayesian Subset Selection with Trees . . . 188
6.2.1 Trees with Spike-and-Slab Regularization . . . 189
6.3 ABC for Variable Selection . . . 192
6.3.1 Naive ABC Implementation . . . 193
6.3.2 ABC Bayesian Forests . . . 195
6.3.3 ABC Bayesian Forests in Action . . . 200
6.4 Model-Free Variable Selection Consistency . . . 202
6.4.1 The Case of Known α . . . 202
6.4.2 The Case of Unknown α . . . 210
6.4.3 Variable Selection Consistency with Bayesian Forests . . . 212
6.5 Simulation Study . . . 214
6.6 HIV Data . . . 219
6.7 Discussion . . . 222
6.8 Appendix . . . 223
6.8.1 Theory . . . 223
6.8.2 Spike-and-Forests: MCMC Variant . . . 238

REFERENCES . . . 241

LIST OF FIGURES

2.1 ABC posterior under misspecified models. 31
2.2 ABC reconstructed posteriors under M/G/1-queuing model 33
2.3 Estimated $\hat{K}(\mathbf{X}, \widetilde{\mathbf{X}^{\theta}})$ over a grid of (θ_1, θ_4) values 36
2.4 ABC posteriors for the Lotka-Volterra model with $\theta_0 = (0.01, 0.5, 1, 0.01)'$ 37
2.5 Closing prices time series for three choices of σ_{12} ($\mu_1 = \mu_2 = 0, \sigma_{11} = \sigma_{22} = 1$). . 39
2.6 Posterior densities on simulated volatility data ($d = 2$) 40
2.7 Posterior densities estimated from log-prices of BA and PG 41

3.1 Approximated posteriors given by B-GAN, SNL, SS, and W2 for the toy example. 72
3.2 Posterior densities under the Gaussian model 75
3.3 MMD between the true posteriors and the approximated posteriors 76
3.4 Approximate posterior densities under the Lotka-Volterra Model 87
3.5 Approximate posterior densities under the Boom-and-Bust Model 90

4.1 Visualization of the two examples 115
4.2 Network Construction 127

5.1 J-copies network architecture 163
5.2 Quartiles of out-of-sample MSEs under the Friedman setup 175
5.3 Computation time under the Friedman setup 176
5.4 Out-of-sample MSEs for the Boston housing dataset 177
5.5 Binary classifications on the wine quality dataset 178
5.6 Binary classifications on the Airbnb booking dataset 180

6.1 (Left) Dynamic ABC plots for evolving inclusion probabilities as ϵ gets smaller. (Right) Plot of $\pi_j(\epsilon)$ obtained with ABC Bayesian Forests (ϵ is the 5% quantile of ϵ_m's) and the variable importance measure from Random Forests (rescaled to have a maximum at 1). 201
6.2 Average variable selection performance under equicorrelation $\rho_{ij} = 0.5$ over 20 simulations 215
6.3 Average variable selection performance under autocorrelation $\rho_{ij} = 0.9^{|i-j|}$ over 10 simulations 216
6.4 Barplots of ordered importance measures for each of the $p = 201$ mutations for the drug APV 219
6.5 (a) The number of true discoveries using an adaptive cut-off; (b) The number of true (red) and false (blue) discoveries using an automated cut-off; (c) The AUC of each method. 221

LIST OF TABLES

2.1 ABC performance on the M/G/1 queuing model over 10 repetitions 34
2.2 ABC performance evaluated on the Lotka-Volterra Model, averaged over 10 repetitions 38
2.3 Performance on the stock volatility estimation example, averaged over 10 repetitions 41
2.4 Posterior estimates on analysis of BA and PG 42

3.1 Summary statistics of the approximated posteriors under the Lotka-Volterra model, averaged over 10 repetitions 89
3.2 Summary statistics of the approximated posteriors under the Boom-and-Bust model, averaged over 10 repetitions 91

5.1 Data augmentation strategies 159
5.2 Frequencies of different wine ratings 177
5.3 Percentage of each class (#obs = 21 3451) 179

6.1 Average out-of-sample mean squared prediction error over 20 independent validation datasets 218

ACKNOWLEDGMENTS

First and foremost, I want to thank my advisors, Profs. Veronika Ročková and Nicholas Gerald Polson, who have been my academic parents since I was a master's student. I am forever indebted to Veronika, who has been an excellent advisor and a cherished friend, and has always gone above and beyond to support me. Her work ethic, passion, and dedication to research inspired me to pursue a career in academia. I am also grateful to her for inviting me over on Thanksgivings, which brought me great joy during pandemic. Nick introduced me to the world of Bayesian statistics and deep learning. He has been an invaluable advisor to me. I enjoyed countless tea times with him and benefited greatly from his knowledge and insights into deep learning. His humor repeatedly saved me from the occasional deep hollows in PhD life.

I would like to thank Prof. Sanjog Misra for all the enlightening conversations that opened up new avenues of my research. I thank Prof. Tengyuan Liang for his encouragement and support during my job market journey. I am also grateful to my committee member, Prof. Chao Gao. His wonderful lectures were instrumental in understanding the essence of van der Vaart's bibles. I also want to express my thanks to my co-authors, Yi Liu and Prof. Tetsuya Kaji, who brought fresh perspectives into our collaborations.

My wonderful PhD adventure could not happen without the generous financial support from Booth School of Business. I am grateful to our PhD office for their amazing support and care. I am fortunate to spend so much time with my fellow doctoral students, Jianeng Xu, Jingyu He, Percy Zhai, Sixun Tang, Meichen Qian, and Zizhe Xia. Thanks to my marketing buddies, Yuxiao Li, Ningyin Xu, Xinyao Kong, Vanessa Alwan, Kevin Lee, Quoc Dang Hung Ho, Tim Schwieg, and Juan Mejalenko, for the fun dinners and poker nights. I want to thank Rui Da and Wanrong Zhu for being the best tennis partners and the most caring friends.

Special thanks to Walter Zhang, my nearest office neighbor and my biggest supporter, for making the worst optimal transport plans, listening to all my random thoughts, celebrating

my every little achievement, and being there for me when things are not that well.

Finally, my deepest gratitude goes to my mother, Keqing Huang, and my father, Zhigang Wang, who granted me the maximum freedom possible to explore my interests, and supported me all the way. Without their unconditional love, I would not be who I am.

ABSTRACT

The strength of the Bayesian paradigm lies in its flexibility through hierarchical modeling and its ability to provide coherent uncertainty quantification. However, the computation costs of classical Bayesian procedures like Markov Chain Monte Carlo (MCMC) can be daunting when confronting big data challenges (large p or large n problems). This thesis innovates Bayesian methodology and theory with the help of modern machine learning techniques, to bring together the best of both worlds.

In Chapter 1, I provide a general introduction of the advancement of machine learning methods, challenges in conducting inference with black-box machine learning methods, and an overview of the Bayesian methodology covered in this thesis.

In Chapter 2 and Chapter 3, I investigate the integration of machine learning techniques and Bayesian computation in case where the likelihood is implicit or intractable. I developed two summary-free Approximate Bayesian Computation (ABC) approaches. The first approach adopts the "classification trick" to estimate the KL divergence between the simulated and observed data. The second approach directly targets at the posterior distribution by matching the joint distribution of the parameter and the data via conditional generative adversarial networks (cGANs).

I study the theoretical guarantees as well as methodology of Bayesian neural networks in Chapter 4 and Chapter 5. Their expressiveness and generalizability has motivated me to deploy deep neural networks inside Bayesian algorithms. This combination not only benefits from the power of neural networks but also retains the inferential potential of the Bayesian probabilistic structure.

I tackle the classical problem of variable selection in the context of ensemble tree-based regression in Chapter 6. To encourage more competition among variables, we place a spike-and-slab wrapper outside the sum-of-trees prior and propose to solve the computation with ABC techniques.

CHAPTER 1

INTRODUCTION

Innovations in machine learning methods and the availability of big data have encouraged numerous advancements in high-dimensional statistical modeling. The general belief in the machine learning community is that with enough data, machine learning models will be able to learn inherent structures in the true underlying data generating process and researchers need to place few assumptions on the model. This success has been witnessed in natural language models, image recognition, game strategies, and various other fields. In particular, universal approximation methods like forests and neural networks has gained popularity because of their exceptional predictive power. Many physical, natural and social science applications have benefited from their expressiveness and flexibility in prediction. However, it is challenging for practitioners to conduct meaningful inference with a fully nonparametric model. It is no secret that interpreting the machine learning methods like forests and neural networks is challenging in itself because the model has no interpretable structure. In addition, it is not obvious whether the model will obey the theory that has been proved to empirically informative by experts, when the data size is limited. A natural remedy to the problem lies in the Bayesian paradigm, which provides automated *uncertainty quantification*. For this thesis, I address some current challenges in statistical inference in the area of deep learning and Bayesian statistics.

1.1 Approximate Bayesian Computation

The first part proposes new methodology in simulation-based inference, where we focus on models with known data generating processes (DGPs) but suffer from computational issues from intractable likelihood functions. One of the driving forces of simulation-based inference is Approximate Bayesian Computation (ABC). The intuition behind ABC is that if the simulated data looks "similar" to the observed data, then the proposed parameters

should be "close" to the true values. A lot of the research in this field has been focused on how to rigorously quantify the closeness between two datasets and what are theoretical guarantees for the approximated posteriors. Very often, ABC methods construct a kernel-type approximation to the posterior distribution through an Accept/Reject mechanism that compares summary statistics of observed and simulated data. Despite the simplicity of usage, they depend on external expert knowledge of the model to specify useful summary statistics. As a result, the corresponding posterior shape highly relies on the choice of summary statistics and the (often ad-hoc) Accept/Reject threshold.

Motivated by these two issues, I propose two summary-free ABC methods which are both inspired by the framework of Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014). Essentially a GAN is an interplay between two neural networks: the discriminator network and the generator network. While the generator tries to learn the underlying representation of the observed dataset and produces fake datasets that looks similar to the real dataset, the discriminator tries to distinguish the fake dataset from the real dataset and provides updates to the generator on how it can improve itself.

In Chapter 2, we use the DGP as the generator and mainly exploit the expressive power of the discriminator. We adopt a classification-based Kullback-Liebler (KL) divergence estimator as the data discrepancy used in the Accept-Reject ABC procedure. The estimator directly quantifies how close the empirical distributions of real and fake datasets are and obviates the need to specify summary statistics. Furthermore, inspired by the connection between the KL divergence and likelihood ratio, we propose an exponential weighting strategy that requires no ad-hoc thresholding or scaling. This connection further allows us to provide theoretical justifications for the approximated posterior distribution.

While our method in Chapter 2 alleviates the two major issues of ABC, it is unclear how to construct a reasonable classifier when the observed dataset has a dependent structure. Further, the need to train a new classifier for each ABC sample makes the computation cost

somewhat heavy. To overcome these limitations, we propose an alternative Bayesian sampler in Chapter 3, where we harness the approximability of the generator network. We build our sampler using conditional GAN (Mirza and Osindero, 2014), which takes conditioned information as part of the input for both the generator and the discriminator. The generator then learns the conditional distribution of the dataset given the conditional information. We further borrow the idea to utilize the architecture to approximate the conditional distribution of the parameter given input dataset. Once a mature, trained generator is obtained, we can use it to retrieve draws from approximated posteriors by plugging in the observed dataset as the condition and passing random latent variables to the generator. The training process is then executed completely on simulated pairs of parameters and datasets, which enables us to conduct inference for dependent dataset.

1.2 Bayesian Deep Learning

In the second part of this thesis, I propose to place Bayesian probabilistic structures on the parameters in a neural network. Specifically, I examine into deep rectified linear unit (ReLU) networks, given their sparsity-inducing nature, superior representation ability, and well-documented empirical successes. However, uncertainty quantification remains a challenge and is largely unexplored. A natural approach to the problem of uncertainty assessment lies in Bayesian hierarchical modeling. I investigate the combination of Bayesian hierarchical modeling and deep ReLU networks from both theoretical and practical perspective.

In Chapter 4, I consider deep ReLU networks with spike-and-slab priors on its weights, and study the limiting behavior of the posterior distribution of the deep sparse ReLU networks. Deep sparse ReLU networks have been shown to be capable of approximating smooth functions and their compositions at the optimal rate (Schmidt-Hieber, 2020). Polson and Ročková (2018) show that, with the spike-and-slab priors, the sparse deep ReLU networks attain the near-optimal speed of posterior contraction. Building on these previous works, I

investigated the semi-parametric Bernstein-von Mises phenomenon. The squashing nature of the ReLU function enables deep ReLU networks to partition the predictor space similarly to trees/forests. Using this connection, I conducted the analysis by holding the deep architecture fixed and performing the change of measure on the final layer, which preserves the same partition of the predictor space and shifts the regression functions only locally. I obtained asymptotic normality results for linear and squared L2-norm functionals for deep sparse ReLU networks. These findings provide a foundation for inference such as testing of the exceedance of a level, constructing confidence balls, and potentially casual inference.

While my previous work in Chapter 4 focuses on the theoretical aspect of Bayesian deep neural networks, Bayesian implementation of deep learning remains practically challenging. In Chapter 5, I propose a new way to update the weights in the neural network via sampling instead of optimization that exploits the duality between regularized optimization and the *maximum a posterior* (MAP) estimator. In particular, by utilizing data augmentation strategies, the objective functions of many commonly-used activation functions, like ReLU and Logit, can be represented as mixtures of Gaussians. While the most straightforward implementation for our data augmentation strategy is to model the neural network as a Bayesian hierarchical model and conduct full Bayesian inference, the number of parameters (neural network weights) can be prohibitively large, making it computationally infeasible to update all of them with conditional Gibbs sampling.

To strike a balance between the inferential potential of Bayesian sampling and the computational efficiency of optimization, I propose an intermediate solution that updates only the last layer of the neural network using data augmentation schemes while still updating the rest of the weights through stochastic gradient descent. I iteratively update the two during the training process to achieve a tradeoff between exploration and exploitation. I provide two solutions to the data augmentation strategies, one approximation approach based on Expectation-Maximization (EM) and another exploratory approach based on MCMC.

I illustrate the usage in three applications: Gaussian regression, Logistic regression, and Support Vector Machine (SVM). The new updating scheme helps neural networks converge faster with significant efficiency gains, and this is validated in our empirical analysis.

1.3 Variable Selection

In the last part of this thesis, I look into a different family of machinery: trees and forests. Similar to other ensemble methods, forests combine several week learners (trees) in order to produce one predictive model. In particular, we consider Bayesian Additive Regression Trees (BART) (Chipman et al., 2010), which have demonstrated great efficacy in various applications. However, its variable screening procedure is rather ad-hoc and does not encourage enough sparsity in high-dimensional settings.

In Chapter 6, we initially proposed a natural solution, which places a spike-and-slab wrapper outside the sum-of-trees prior. However, the fully Bayesian computation then demands a stochastic reversible jump search (Green, 1995) where the dimensionality of the predictor space for each tree can change. This is inherently difficult and computationally burdensome when combined with internal tree splitting moves. As an alternative to the MCMC, we tackled the problem with ABC, which provides a new avenue towards approximating the median probability model in non-parametric setups where the marginal likelihood is intractable. The innovation of the method lies in the two-step data splitting strategy where (1) a random subset of data is used to come up with a proposal draw and (2) the rest of the data is used for ABC acceptance. We made contributions to both model-free variable selection and ABC literatures. We generalize ABC methods to a nonparametric regression setting and use model-fit criteria, such as mean squared error on validation set, instead of summary statistics to quantify the distance between the proposed model and optimal model. We demonstrate the effectiveness of our methods through several simulated examples and an analysis of the HIV data.

CHAPTER 2

ABC VIA CLASSIFICATION

Approximate Bayesian Computation (ABC) enables statistical inference in simulator-based models whose likelihoods are difficult to calculate but easy to simulate from. ABC constructs a kernel-type approximation to the posterior distribution through an Accept/Reject mechanism which compares summary statistics of real and simulated data. To obviate the need for summary statistics, we directly compare empirical distributions with a Kullback-Leibler (KL) divergence estimator obtained via contrastive learning. In particular, we blend flexible machine learning classifiers within ABC to automate fake/real data comparisons. We consider the traditional Accept/Reject kernel as well as an exponential weighting scheme which does not require the ABC acceptance threshold. Our theoretical results show that the rate at which our ABC posterior distributions concentrate around the true parameter depends on the estimation error of the classifier. We derive limiting posterior shape results and find that, with a properly scaled exponential kernel, asymptotic normality holds. We demonstrate the usefulness of our approach on simulated examples as well as real data in the context of stock volatility estimation.

2.1 Introduction

We consider a collection of i.i.d. observations $\boldsymbol{X} = (X_1, \ldots, X_n)'$ where each $X_i \in \mathcal{X}$ is realized from a parametric model $\{P_{\theta_0} : \theta_0 \in \Theta \subset \mathbb{R}^d\}$. We assume that P_θ, for each $\theta \in \Theta$, admits a density p_θ. We are interested in Bayesian inference about θ_0 based on the posterior distribution

$$\pi_n(\theta \mid \boldsymbol{X}) \propto p_\theta^{(n)}(\boldsymbol{X})\pi(\theta) \tag{2.1}$$

. Adopted from Yuexi Wang, Tetsuya Kaji, and Veronika Ročková. Approximate Bayesian computation via classification. *Journal of Machine Learning Research*, 23(350):1–49, 2022a.

prescribed by the likelihood $p_\theta^{(n)}(\boldsymbol{X})$ and the prior density $\pi(\theta)$. We are particularly interested in simulator-based models whose the likelihood function cannot be directly expressed/evaluated (such as discretely observed diffusions (Sørensen, 2004) or generative models) but can be sampled from.

Simulator-based models are often called implicit models because the the likelihood function p_θ cannot be numerically evaluated (Diggle and Gratton, 1984). Fortunately, it may still be possible to simulate synthetic datasets from the model. The ability to simulate from the likelihood has opened up new opportunities for simulating from the posterior. For example, Approximate Bayesian Computation (ABC) (Pritchard et al., 1999; Beaumont et al., 2002) emerged as a default likelihood-free Bayesian inferential tool. It is an Accept/Reject posterior sampling mechanism which obviates likelihood evaluations. Each iteration proceeds by (1) simulating prior parameter guesses and fake data from the likelihood, and then (2) accepting those parameter values whose fake data were close to the observed data. A big challenge with ABC has been gauging the similitude between observed and fake data.

Measures of similarity between data sets have traditionally been based on summary statistics (see Blum et al. (2013) for an overview within the ABC context). In other words, two datasets are considered similar if their summary statistics are close. In the absence of expert knowledge, however, constructing effective summary statistics can be challenging (Joyce and Marjoram, 2008; Nunes and Balding, 2010; Blum et al., 2013) and one may need to resort to automated strategies. One possibility is regressing parameter values onto (functionals of) fake data in a pilot ABC run to train a flexible mapping which can be substituted for summary statistics (Fearnhead and Prangle, 2011; Jiang et al., 2017b; Akesson et al., 2021). Another possibility, related to indirect inference, is to construct summary statistics from an auxiliary model (Drovandi et al., 2011; Wood, 2010). One can also choose a subset of candidate summary statistics that satisfy some optimality criterion (Joyce and Marjoram, 2008; Nunes and Balding, 2010) or find an optimal projection of a set of summary statistics onto

a lower-dimensional map (Fearnhead and Prangle, 2011). Alternatively, one can directly use a discrepancy between the empirical distributions of the observed and synthetic data sets inside ABC (such as Kullback-Leibler (Jiang et al., 2018) or Wasserstein (Bernton et al., 2019) or Maximum Mean (MM) discrepancy (Park et al., 2016)) or Energy Statistics (ES) (Nguyen et al., 2020)). See Drovandi and Frazier (2022) for a nice comprehensive review of the distribution-style ABC discrepancies. Our work fortifies this ABC point of view by focusing on the Kullback-Leibler discrepancy estimated via classification.

The KL divergence is one of the most widely used discrepancy metrics. It expresses the average information per observation to discriminate between two probabilistic models (Kullback, 1958). In large deviations, for example, it characterizes the exponential decay rate at which empirical measures converge to their probabilities (see Sanov's theorem in Den Hollander (2008)) and the rate of decay of the probability of error in a binary hypothesis testing problem (see Stein's Lemma in Cover and Thomas (1991)). KL also naturally connects to maximum likelihood estimation through its interpretation as the expectation of the log-likelihood ratio. There exist many methods for estimating the KL divergence. For example, Wang et al. (2009) proposed nearest-neighbor techniques to obtain a mean-square consistent estimator. Wang et al. (2005) proposed a histogram-based KL estimator based on partitioning of the space into statistically equivalent intervals. Silva and Narayanan (2007) and Silva and Narayanan (2010) went a step further and proposed using data-driven partitions (including multivariate recursive partitioning) and formulated sufficient consistency conditions. Alternatively, Nguyen et al. (2007) proposed a variational approach by turning KL estimation into a penalized convex risk minimization problem. Our work is different from the approaches above as we adopt a KL estimator based on classification.

We suggest embedding a machine learning classifier inside ABC to determine whether or not fake and observed data are similar and, thereby, whether or not the underlying parameter value should be kept in the ABC reference table. The fundamental premise of this proposal is

as follows: parameter values that yield indistinguishable simulated datasets can be deemed close. Bayesian inference via classification has been suggested before. Kaji and Ročková (2022) developed a version of the Metropolis-Hastings algorithm, called MHC, based on classification-based estimators of likelihood ratios. Thomas et al. (2022) derived a marginal approach by contrasting two fake datasets generated from the marginal and conditional likelihoods. Gutmann et al. (2018) proposed a classification strategy related to ours using a different discrepancy metric. Our paper reframes the method of Gutmann et al. (2018) as a genuine ABC algorithm with a KL divergence discriminator and provides supporting theory which justifies its inferential potential.

In particular, we study statistical properties of the approximate posterior which, in part, depend on the properties of the KL divergence estimator. We consider the traditional Accept/Reject ABC version (with a uniform kernel) as well as an exponential kernel variant which does not require the ABC tolerance threshold. Similar to Frazier et al. (2018), we show that the choice of the ABC acceptance threshold ϵ plays a critical role in the convergence rate and in the limiting posterior shape. In practice, it is often not obvious what the optimal threshold ϵ should be. Motivated by the connections with the MHC algorithm of Kaji and Rockova (2021), we propose an exponential kernel which yields ABC posteriors that correspond to the stationary distribution of MHC. Our ABC kernel method can be thus regarded as a parallelizable counterpart to the sequential MHC sampling, targeting the same posterior approximation. The concentration and asymptotic shape behavior of the ABC posterior, which can be derived from Kaji and Ročková (2022), theoretically justify our exponential weighting scheme. Finally, our classification-based ABC approach provides a viable computational strategy for obtaining coarsened posteriors for Bayesian robust inference (Miller and Dunson, 2018). Our ABC approach leverages machine learning but does so in a perhaps more traditional way than the recent sequential neural likelihood and mixture density network approaches for learning posteriors (Papamakarios and Murray, 2016;

Papamakarios et al., 2019).

The remaining of the paper is structured as follows. In Section 2.2, we flesh out the basic idea of ABC and introduce our framework with classification. In Section 2.3, we investigate the posterior concentration and limiting shape behaviors of the ABC posteriors. Section 2.4 shows performance on simulated datasets and Section 2.5 further highlights the practical value of our approach on real data. In Section 2.6, we conclude with a discussion.

Notation. We use the shorthand notation $p_0 = p_{\theta_0}$ and $P_0 = P_{\theta_0}$. We employ the operator notation for expectation, e.g., $P_0 f = \int f dP_0$. The ϵ-bracketing number $N_{[]}(\epsilon, \mathcal{F}, d)$ of a set $\mathcal{F}$ with respect to a premetric d is the minimal number of ϵ-brackets in d needed to cover $\mathcal{F}$[1]. The δ-bracketing entropy integral of $\mathcal{F}$ with respect to d is $J_{[]}(\delta, \mathcal{F}, d) = \int_0^\delta \sqrt{1 + \log N_{[]}(\epsilon, \mathcal{F}, d)} d\epsilon$. Next, $K(f, g) = \int f \log(f/g) d\mu$ denotes the Kullback-Leibler divergence between two density functions and $V_2(f, g) = \int f \left|\log(f/g)\right|^2 d\mu$. For real-valued sequences $\{a_n\}_{n\geq 1}$ and $\{b_n\}_{n\geq 1}$, $a_n \lesssim b_n$ means that $a_n \leq C\, b_n$ for some generic constant $C > 0$, $a_n \asymp b_n$ means that $a_n \lesssim b_n \lesssim a_n$, and $a_n \gg b_n$ indicates a greater order of magnitude. For a sequence of random variables x_n, $x_n = o_P(a_n)$ if $\lim_{n\to\infty} P(|x_n/a_n| \geq C) = 0$ for every $C > 0$, and $x_n = O_P(a_n)$ if for every $C > 0$ there exists a finite $M > 0$ and a finite N such that $P(|x_n/a_n| \geq M) \leq C$ for all $n > N$. All limits are taken as $n \to \infty$. Take $\|\cdot\|$ to be the Euclidean norm.

2.2 ABC without Summary Statistics

The now default ABC method for Bayesian likelihood-free inference constructs a nested kernel-type approximation to the posterior distribution. The first approximation occurs when the data is distilled into summary statistics to obtain $\pi(\theta \,|\, S_X) \propto \pi(S_X \,|\, \theta)\pi(\theta)$, where $S_X = S(\boldsymbol{X})$ is a vector of summary statistics. The quality of this approximation depends crucially on the informativeness of S_X. The actual ABC approximation to the posterior

1. A premetric on $\mathcal{F}$ is a function $d : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ such that $d(f, f) = 0$ and $d(f, g) = d(g, f) \geq 0$.

(2.1) is then constructed via a kernel function as $\pi_{ABC}(\theta \mid S_X) = \int \pi(\theta, S \mid S_X) dS$ with $\pi(\theta, S) \propto T_\epsilon(\|S - S_Y\|)\pi(S \mid \theta)\pi(\theta)$, where $\|\cdot\|$ is a general norm to be specified later by the user and where $T_\epsilon(\|u\|) = T(\|u\|/\epsilon)$ is a standard (smoothing) kernel with a scale parameter $\epsilon > 0$. The key two challenges with ABC are (1) deriving low-dimensional summary statistics with a minimal loss of information and (2) selecting the kernel and its the tolerance level ϵ. To remediate the reliance of ABC on summary statistics, we focus on viewing the observed and fake data as empirical distributions and gauge the discrepancy between them. See Drovandi and Frazier (2022) for an overview of other discrepancy-based ABC methods. Regarding the choice of aggregation kernels, we consider the traditional uniform kernel yielding an Accept/Reject algorithm and a smoothing kernel free from ϵ-tuning.

We are interested in regimes where ABC is most effective (Jiang et al., 2017b), i.e. settings where the sample size n of $\boldsymbol{X}$ is moderately high and the dimension of Θ is low to ensure that we can hit the high ABC-posterior region with a reasonable prior probability. The number of observations n has an impact on the effectiveness of the embedded classifier. The nonparametric neural network classifiers usually demand n to be somewhat large so that the estimation errors are manageable.

2.2.1 ABC with KL Divergence

Instead of summary statistics, we use the estimated KL divergence inside the ABC algorithm. Our interest in the KL divergence as a discrepancy metric stems partially from the following connection to the generalized Bayesian inference (Bissiri et al., 2016). The posterior distribution (2.1) can be rewritten as a generalized posterior $\pi_n(\theta \mid \boldsymbol{X}) \propto \pi(\theta) \exp(-n \times KL(p_0^{(n)}, p_\theta^{(n)}))$ where the parameter θ is linked to data through the empirical Kullback-Liebler (KL) divergence $KL(p_0^{(n)}, p_\theta^{(n)}) \equiv \frac{1}{n}\sum_{i=1}^n \log(p_0/p_\theta)(X_i)$. For when the KL divergence cannot be easily evaluated, we consider various estimators in the next section. We denote a generic KL divergence estimator obtained from observed data $\boldsymbol{X} \sim P_0^{(n)}$ and

Algorithm 1: KL-ABC with Accept-Reject

For a pre-determined tolerance level $\epsilon > 0$ repeat for $j = 1, \ldots, N$:

1. Simulate θ_j from $\pi(\theta)$.
2. Simulate $\tilde{\boldsymbol{X}}^{\theta_j} = (\tilde{X}_1^{\theta_j}, \ldots, \tilde{X}_m^{\theta_j})'$ through i.i.d. sampling from the model p_{θ_j}.
3. Construct $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta_j})$ by training a classifier distinguishing $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}^{\theta_j}$ as in (2.8).
4. Accept θ_j when $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta_j}) \leq \epsilon$.

pseudo-data $\tilde{\boldsymbol{X}}^{\theta} \sim P_{\theta}^{(n)}$ as $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta})$. Adopting $\hat{K}(\cdot, \cdot)$ as the ABC discrepancy, we consider a simple Accept/Reject ABC mechanism detailed in Algorithm 1 below. While Jiang et al. (2017b) used a nearest-neighbor estimator of the KL divergence, we devise a different estimator based on classification in Section 2.2.3.

Algorithm 1 simulates pairs of parameter values and pseudo-data $\{\theta, \tilde{\boldsymbol{X}}^{\theta}\}$ from the joint posterior density

$$\hat{\pi}^{AR}\left(\theta, \tilde{\boldsymbol{X}}^{\theta} \mid \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\right) = \frac{\pi(\theta) p_{\theta}^{(n)}(\tilde{\boldsymbol{X}}^{\theta}) \mathbb{I}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\big)}{\int \pi(\theta) p_{\theta}^{(n)}(\tilde{\boldsymbol{X}}^{\theta}) \mathbb{I}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\big) \mathrm{d}\tilde{\boldsymbol{X}}^{\theta} \mathrm{d}\theta}, \tag{2.2}$$

which margins towards the following approximate (Accept/Reject) posterior density

$$\begin{aligned} \hat{\pi}_{\epsilon}^{AR}(\theta \mid \boldsymbol{X}) &= \int \hat{\pi}^{AR}\left(\theta, \tilde{\boldsymbol{X}}^{\theta} \mid \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\right) \mathrm{d}\tilde{\boldsymbol{X}}^{\theta} \\ &\equiv \frac{\pi(\theta) P_{\theta}^{(n)}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\big)}{\int \pi(\theta) P_{\theta}^{(n)}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) \leq \epsilon\big) \mathrm{d}\theta}. \end{aligned} \tag{2.3}$$

The inferential potential of the approximation (2.3) will be scrutinized theoretically later in Section 2.3.2. In particular, we will later see that the convergence rate of (2.3) around θ_0 depends on the choice of ϵ (Frazier et al., 2018) as well as the quality of the discriminator. It is interesting to note that the ABC posterior (2.3) is mathematically equivalent to the *c*-posterior proposed by Miller and Dunson (2018) for robust inference in mis-specified (tractable) models. The computation of the *c*-posteriors has relied on powered-likelihood approximations and MCMC sampling. While we instead view (2.3) as an approximate pos-

terior in models with intractable likelihoods, our ABC algorithm can be nevertheless used to compute coarsened posteriors in broader scenarios when MCMC sampling may not be available (see Remark 2.2.2 below). Algorithm 1 uses the uniform kernel which corresponds to the indicator function $T_\epsilon(\|u\|) = \mathbb{I}(\|u\| \leq \epsilon)$. In practice, it is difficult to balance out conflicting demands of smaller ϵ (yielding good approximability) and larger acceptance rates (yielding more posterior samples). As a remedy, we propose a way to aggregate the ABC samples through a scaled exponential kernel motivated by the connection between KL and the log-likelihood ratio. This ABC variant requires no ad-hoc thresholding and is summarized in Algorithm 2.

Algorithm 2 generates draws for the pair $\{\theta, \tilde{\boldsymbol{X}}^\theta\}$ from a joint posterior density

$$\hat{\pi}^{EK}(\theta, \tilde{\boldsymbol{X}}^\theta \mid \boldsymbol{X}) = \frac{\pi(\theta) p_\theta^{(n)}(\tilde{\boldsymbol{X}}^\theta) \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big)}{\int \pi(\theta) p_\theta^{(n)}(\tilde{\boldsymbol{X}}^\theta) \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big) \mathrm{d}\tilde{\boldsymbol{X}}^\theta \mathrm{d}\theta}, \tag{2.4}$$

which leads to the approximated Bayesian posterior as

$$\hat{\pi}^{EK}(\theta \mid \boldsymbol{X}) = \int \hat{\pi}^{EK}(\theta, \tilde{\boldsymbol{X}}^\theta \mid \boldsymbol{X}) \mathrm{d}\tilde{\boldsymbol{X}}^\theta = \frac{\pi(\theta) P_\theta^{(n)} \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big)}{\int \pi(\theta) P_\theta^{(n)} \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big) \mathrm{d}\theta}. \tag{2.5}$$

Remark 1 (Generating Fake Data)**.** *We assume $\tilde{\boldsymbol{X}}^\theta = g_\theta(\tilde{\boldsymbol{X}})$, where $\tilde{\boldsymbol{X}} \in \mathbb{R}^m$ are random variables arriving from $\tilde{P}^{(m)}$ and where $g_\theta : \mathbb{R}^m \to \mathbb{R}^m$ is a deterministic mapping. Generating random variable draws by passing $\tilde{\boldsymbol{X}}$ through some mapping is commonly done in practice, also known as the reparameterization trick (Kingma and Welling, 2013). For*

Algorithm 2: KL-ABC with Exponential Weighting

Repeat for $j = 1, \ldots, N$:

1. Simulate θ_j from $\pi(\theta)$.
2. Simulate $\tilde{\boldsymbol{X}}^{\theta_j} = (\tilde{X}_1^{\theta_j}, \ldots, \tilde{X}_m^{\theta_j})'$ through i.i.d. sampling from the model p_{θ_j}.
3. Construct $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta_j})$ by training a classifier distinguishing $\boldsymbol{X}$ and $\tilde{\boldsymbol{X}}^{\theta_j}$ as in (2.8).
4. Assign θ_j a weight proportional to $\exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big)$.

example, Gaussian random variables $\tilde{\boldsymbol{X}} = \{\tilde{X}_i^\theta\}_{i=1}^m$ *that follow i.i.d.* $N(\mu, \sigma^2)$ *distribution can be obtained by transforming* $\{\tilde{X}_i\}_{i=1}^m \overset{i.i.d.}{\sim} N(0,1)$ *via* $\tilde{X}_i^\theta = \mu + \sigma\tilde{X}_i$*. In other cases, one can use uniform draws* $\tilde{\boldsymbol{X}}$ *and the inverse transform sampling.*

Smooth kernels have been used inside ABC before to rescale the acceptance probability (e.g. Beaumont et al. (2002) employed the Epanechnikov kernel). Wilkinson (2013) and Sisson et al. (2018) provide a thorough overview and comparisons of the commonly used kernels. Our smoothed weights are directly interpretable due to their linkage between the KL divergence and the log-likelihood ratio. Algorithm 2 can be regarded as a version of Importance Sampling ABC (see Nguyen et al. (2020) for a variant using energy statistics and Park et al. (2016) for minimal description length ABC). We later show in Section 2.3.2 that, with the scaled exponential kernel, the ABC posterior corresponds to the stationary distribution of the MHC algorithm of Kaji and Ročková (2022) and can be regarded as a posterior under a misspecified model. Computational comparisons of the sequential MHC sampler with our parallelizable ABC sampler are performed in the Appendix C of Wang et al. (2022a) where we show benefits of the ABC strategy when convergence issues may arise for MHC due to initialization. In Section 2.3.4, we perform comparisons of the Accept/Reject (AR) and exponential kernels under model misspecification where we show that the AR kernel is far more robust.

The classifier needs to be trained for each ABC draw, which may incur additional computational cost compared to traditional ABC where the summary statistics and their distance can be computed without optimization. We provide comparisons of computation times in Appendix F of Wang et al. (2022a). Although the computation costs of our methods are higher when the data dimensionality d is relatively small, we are less disadvantageous when d is large compared to other discrepancies like Wasserstein distance or Maximum Mean Discrepancy. In addition, nonparametric discriminator classes such as neural network classifiers can efficiently benefit when there is an inherent low-dimensional structure in the data (Kaji

et al., 2020). Additionally, training can be accelerated if one initiates the training at some pre-trained neural networks.

2.2.2 Connection to Robust Bayesian Inference

Our algorithms can be regarded as "robust ABC" algorithms that estimate (relative-entropy coarsened posteriors) c-posteriors introduced in Miller and Dunson (2018). Note that our focus of robustness is different from the robustness to different distance measures, but more relevant to data perturbation or model misspecification (discussed later in Section 2.3.4). The c-posteriors yield robust inference by conditioning on the event that the observed data $\boldsymbol{X}$ is sufficiently close (in terms of the KL divergence) to the data generated by the model. A similar case as the Huber-type data contamination is considered in γ-divergence ABC Fujisawa et al. (2021). The proposed computation of c-posteriors in Miller and Dunson (2018) is made feasible only through asymptotic approximations (Section 3.1 in Miller and Dunson (2018)), e.g. with powered posteriors that are computable using conjugate priors. Our ABC methods can compute them without any approximation and for a broader class of priors. In particular, if in Algorithm 1 we draw $\epsilon \sim \text{Exp}(\alpha)$ for some $\alpha > 0$ and accept θ if $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) < \epsilon$, then our ABC posterior coincidentally approximates the relative-entropy $c-$posterior proportional to $\pi(\theta)P_{\theta}^{(n)}\mathrm{e}^{-\eta K_n}$ where $K_n = \mathbb{P}_n \log \frac{p_0}{p_\theta}$. Algorithm 2 corresponds to the case $\alpha = n$ in Miller and Dunson (2018) without any approximation. Interestingly, the degree of robustness corresponds to the acceptance rate of ABC. For example, if we let $\alpha \ll n$, the c-posterior puts larger weight on the prior and robustifies the model, which corresponds to accepting many draws (more than proportional to n) and the draws reflecting the shape of the prior. On the contrary, if we let $\alpha \gg n$, the c-posterior puts most weight on a narrow neighborhood of the observed data, which corresponds to accepting very few draws for which the Kullback–Leibler divergence is the smallest. From an ABC's perspective, probably the most interesting case is when the acceptance rate is roughly fixed throughout

$n \to \infty$, in which case α is comparable with n and our algorithms produce a correct c-posterior without utilizing the approximation in Miller and Dunson (2018) which stands on $n \gg \alpha$ or $n \ll \alpha$. In addition, another advantage of our method is that it allows us to calculate the c-posterior for different α easily. If we use an MCMC with tempering (i.e. the powered likelihood approximation), we might need to run separate MCMC chains for different α. On the other hand, our ABC-based algorithm lets us calculate the c-posteriors by filtering out independent samples of candidate draws according to various α. This may be advantageous in applications when no ex-ante preference is available on the degree of robustness and when one wants to see how the concentration of the c-posterior varies with it.

2.2.3 Estimating KL Divergence via Classification

We adopt the '$-\log D$' trick to estimate the KL divergence (Goodfellow et al., 2014). More precisely, a flexible discriminator D (such as a neural network or logistic regression) is trained to maximize

$$\mathbb{M}_{n,m}^{\theta}(D) = \mathbb{P}_n \log D + \mathbb{P}_m^{\theta} \log(1 - D), \tag{2.6}$$

where we employ the operator notation for expectation, e.g., $\mathbb{P}_n f = \frac{1}{n}\sum_{i=1}^{n} f(X_i)$ and $\mathbb{P}_m^{\theta} f = \frac{1}{m}\sum_{i=1}^{m} f(\tilde{X}_i^{\theta})$. This can also be regarded as a classification problem where we label $\{X_i\}_{i=1}^{n}$ ('real' data) with 1 and $\{\tilde{X}_i^{\theta}\}_{i=1}^{m}$ ('fake' data) with 0. The oracle maximizer to (2.6) can be shown to be (Goodfellow et al., 2014, Proposition 1)

$$D_{\theta}(X) = \frac{p_0(X)}{p_0(X) + p_{\theta}(X)}. \tag{2.7}$$

The functional form of the oracle solution in (3.31) naturally suggests the following KL estimator obtained from a trained discriminator $\hat{D}^{\theta}_{n,m}$ (Thomas et al., 2020b)

$$\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) = \mathbb{P}_n \log \frac{\hat{D}^{\theta}_{n,m}}{1 - \hat{D}^{\theta}_{n,m}} = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\hat{D}^{\theta}_{n,m}(X_i)}{1 - \hat{D}^{\theta}_{n,m}(X_i)}. \tag{2.8}$$

Later we show that our classification-based KL estimator (2.8) converges to a well-defined limit counterpart $K(p_0, p_\theta)$ under mild conditions in Section 2.3.1.

2.2.4 Other KL Estimators

Our ABC results, presented later in Section 2.3, can be extended to other types of KL estimators if similar estimation error results as in Theorem 5 can be shown. Since the rate $1/(nu^2)$ stems from the estimation error of the empirical KL divergence, the fundamental difference between our classification-based KL estimator and other KL estimators lies in the rate δ_n. One example is the k-Nearest Neighbor (kNN) estimator proposed in Pérez-Cruz (2008). Wang et al. (2009) showed that this estimator is asymptotically unbiased and mean-square consistent and they propose a data-dependent choice of k which can improve the convergence speed. Jiang et al. (2018) assess data discrepancy inside ABC with the special case of 1-nearest neighbor, which is defined as

$$\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) = \frac{d}{n} \sum_{i=1}^{n} \log \frac{\min_j \|X_i - \tilde{X}^{\theta}_j\|}{\min_{j \neq i} \|X_i - X_j\|} + \log \frac{m}{n-1}. \tag{2.9}$$

where d is the number of covariates in $\boldsymbol{X}$. Zhao and Lai (2020) provide convergence rate of the bias for this kNN estimator is bounded by $n^{-2\gamma/(d+2)} \log n$, where γ is a parameter characterizing the tail behavior of the target distribution. Note that the kNN estimator is not applicable to cases where $\boldsymbol{X}$ arises from a discrete distribution.

Another route to estimate the KL divergence is via (data-dependent) partitioning meth-

ods. Wang et al. (2005) proposed to estimate the Radon-Nikodym derivative $\mathrm{d}P_0/\mathrm{d}P_\theta$ using frequency counts on a statistically equivalent partition of $\mathbb{R}^d$. However, the computational complexity of their method is exponential in d and the estimation accuracy deteriorates quickly as the dimension increases. Silva and Narayanan (2010) further contributed to multivariate data-driven partition schemes by using a Barron-type histogram-based density estimate. They provide sufficient conditions on the partitions scheme to make the estimator strongly consistent.

Lastly, Nguyen et al. (2007) adopted a variational approach to estimate KL by reframing the estimation problem as a penalized convex risk minimization problem, where they restrict the estimate to a bounded subset of a Reproducing Kernel Hilbert Space (RKHS). Convergence rates are then obtained from empirical process theory on nonparametric M-estimators (van de Geer, 2000). In an independent contribution, Ghimire et al. (2021) used a discriminator in RKHS to estimate KL using a similar approach to ours. They showed that the estimator error bound is related to the complexity of the discriminator in RKHS. A comparison of computational complexities of these methods and our approach can be found in Appendix G of Wang et al. (2022a).

Beyond the forward KL divergence, our classification framework allows us to consider other discrepancy metrics. Alternatively to (2.8), we could instead estimate the reversed KL divergence

$$\hat{K}_{\text{reverse}}(\tilde{\boldsymbol{X}}^\theta, \boldsymbol{X}) = \frac{1}{m}\sum_{i=1}^{m} \log \frac{1-\hat{D}^\theta_{n,m}}{\hat{D}^\theta_{n,m}}(\tilde{X}^\theta_i)$$

which converges to $K(p_\theta, p_0)$ and which is still uniquely minimized at $p_\theta = p_0$. One can show that the estimation error of this reversed KL estimator is still $O_{P^*}(\delta_n)$ by following the same techniques used in Lemma 3. The reversed KL divergence is widely used in variational inference (Jordan et al., 1999; Wainwright and Jordan, 2008). Forward and reversed KL's could perform differently when the function class inside the variational approach is not rich enough. The reversed KL is zero-forcing/mode-seeking, while the forward KL is mass-

covering/mean-seeking (Bishop, 2006). Another related metric, deployed by Gutmann et al. (2018), is the classification accuracy (CA) defined as

$$\mathrm{CA}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) = \frac{1}{n+m}\left(\sum_{i=1}^{n} \hat{D}^\theta_{n,m}(X_i) + \sum_{i=1}^{m}\big(1 - \hat{D}^\theta_{n,m}(\tilde{X}^\theta_i)\big)\right). \tag{2.10}$$

Since $\hat{D}(\cdot)$ and $\log \frac{\hat{D}}{1-\hat{D}}(\cdot)$ are linked by a logistic transformation which is Lipschitz-continous, CA can be roughly regarded as a weighted average of the forward KL divergence and the reversed KL divergence. Our framework is connected to other GAN-style discrepancy metrics which are also used in ABC literature. We provide a discussion in Wang et al. (2022a, Appendix D).

2.3 Frequentist Analysis of the ABC Posterior

One way to assess the quality of the posterior distribution is through the speed at which it contracts around the truth θ_0 as $n \to \infty$. While the ABC posterior is ultimately an approximation, it might still concentrate about θ_0 at a reasonable rate. In this section, we look into theoretical properties of both Algorithm 1 and Algorithm 2. First, we develop a tail bound result quantifying how fast the classification-based estimator $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)$ converges to the true KL divergence $K(p_0, p_\theta)$ conditional on approximability of the discriminator class. The tail bound analysis is crucial in our convergence analysis. Next, we show that the convergence rate of the accept-reject ABC in Algorithm 1 is determined jointly by the accept-reject threshold ϵ_n, the estimation error δ_n (with respect to $\mathbb{P}_n \log p_0/p_\theta$), and the rate $n^{-1/2}$ of estimation between $\mathbb{P}_n \log p_0/p_\theta$ and $K(p_0, p_\theta)$. Further, the typical posterior distribution converges to a uniform distribution over an ellipse when the acceptance threshold ϵ_n dominates the other two. On the other hand, the exponentially weighted posterior in Algorithm 2 can be viewed as the posterior under a "misspecified" model. The convergence rate is then determined by the contraction rate of the true posterior and the estimation error,

where the ABC posterior is asymptotically normal around the KL projection under LAN conditions (Kaji and Ročková, 2022).

2.3.1 Convergence Rate of Estimation Errors

We assume that the set of considered classifiers $\mathcal{D}$ resides in a sieve $\mathcal{D}_n$ that expands with the sample size and its size is measured by bracketing entropy $N_{[]}(\epsilon, \mathcal{F}, d)$. Kaji et al. (2020) prove the rate of convergence of such a classifier (under assumptions reviewed later) with a Hellinger-type distance defined as

$$d_\theta(D_1, D_2) = \sqrt{h_\theta(D_1, D_2)^2 + h_\theta(1 - D, 1 - D_\theta)^2},$$

where $h_\theta(D_1, D_2) = \sqrt{(P_0 + P_\theta)(\sqrt{D}_1 - \sqrt{D}_2)^2}$.

Assumption 1. *Assume that n/m converges and that an estimator $\hat{D}^\theta_{n,m}$ exists that satisfies $\mathbb{M}_{n,m}(\hat{D}^\theta_{n,m}) \geq \mathbb{M}^\theta_{n,m}(D_\theta) - O_P(\delta_n^2)$ for a nonnegative sequence δ_n. Moreover, assume that the bracketing entropy integral satisfies $J_{[]}(\delta_n, \mathcal{D}^\theta_{n,\delta_n}, d_\theta) \lesssim \delta_n^2\sqrt{n}$ and that there exists $\alpha < 2$ such that $J_{[]}(\delta, \mathcal{D}^\theta_{n,\delta}, d_\theta)/\delta^\alpha$ has a majorant decreasing in δ. Here $\mathcal{D}^\theta_{n,\delta_n} = \{D \in \mathcal{D}_n : d_\theta(D, D_\theta) \leq \delta_n\}$.*

Assumption 1 requires three conditions. First, the synthetic data sample size m should be at least as large as the actual data size n, which can be assured. Second, the discriminator class needs to be expressive enough so we can find a sufficiently good maximizer approximating the oracle discriminator D_θ. Lastly, the entropy of the sieve should be moderate to prevent overfitting.

The following theorem states that the sequence δ_n in Assumption 1 determines the convergence rate of $\hat{D}$. The speed at which δ_n converges to 0 depends on the choice of the sieve and smoothness of the model. When a nonparametric estimator is employed, δ_n is often slower than $n^{-1/2}$. In Section 2.7.1, we give a specific expression of δ_n for a neural network

classifier and give a few examples in which δ_n vanishes faster than $n^{-1/4}$.

Lemma 2 (Kaji et al., 2020, Theorem S.1)**.** *Under Assumption 1, we have* $d_\theta(\hat{D}^\theta_{n,m}, D_\theta) = O^*_P(\delta_n)$.[2]

To establish the rate of convergence of our approximated posterior, we have the following assumption.

Assumption 2. *There exists* $\Lambda > 0$ *such that for every* $\theta \in \Theta$, $P_0(p_0/p_\theta)$ *and* $P_0(p_0/p_\theta)^2$ *are bounded by* Λ *and*

$$\sup_{D \in \mathcal{D}^\theta_{n,\delta_n}} P_0\left(\frac{D_\theta}{D}\middle|\frac{D_\theta}{D} \geq \frac{25}{16}\right) < \Lambda, \quad \sup_{D \in \mathcal{D}^\theta_{n,\delta_n}} P_0\left(\frac{1-D_\theta}{1-D}\middle|\frac{1-D_\theta}{1-D} \geq \frac{25}{16}\right) < \Lambda,$$

for δ_n *in Assumption 1. The brackets in Assumption 1 can be taken so that* $P_0(\sqrt{u/l}-1)^2 = O(d_\theta(u,l)^2)$ *and* $P_0(\sqrt{(1-l)/(1-u)}-1)^2 = o(d_\theta(u,l))$.

Assumption 2 constrains the tail behavior of the discriminator so that the residual of the cross-entropy loss in (2.6) can be circumscribed by the bracketing entropy. See Kaji and Ročková (2022) for a discussion of how this can be reasonably satisfied for logistic discriminator and neural network discriminators that use sigmoid activation functions.

The following theorem quantifies the rate of convergence of our estimator (2.8) towards the *empirical* KL divergence $\mathbb{P}_n \log \frac{p_0}{p_\theta}$.

Lemma 3 (Convergence Rate of Estimation Errors)**.** *Under Assumptions 1 and 2,*

$$\left|\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) - \mathbb{P}_n \log \frac{p_0}{p_\theta}\right| = O_{P^*}(\delta_n). \tag{2.11}$$

2. We use P^* to denote outer expectation (see Section 1.2 of van der Vaart and Wellner (1996)), here is the expectation of a "a smallest measureable function g that dominates $d_\theta(\hat{D}^\theta_{n,m}, D_\theta)$".

Proof. Since the estimation error can be rewritten as

$$\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) - \mathbb{P}_n \log \frac{p_0}{p_\theta} = -\mathbb{P}_n \bigg(\log \frac{1 - \hat{D}^{\theta}_{n,m}}{1 - D_\theta} - \log \frac{\hat{D}^{\theta}_{n,m}}{D_\theta} \bigg),$$

it follows from Kaji and Ročková (2022, Theorem 4.1).

Remark 4 (Uniform Convergence Rate)**.** *The rate of estimation in Lemma 2 and Lemma 3 is characterized as point-wise. While for each $\theta \in \Theta$ the estimation error is shrinking at the rate δ_n, the multiplication constant in front the rate could potentially depend on θ. Assuming a compact parameter support (which is not a-typical in deep learning models; see e.g. Schmidt-Hieber (2020) or Polson and Ročková (2018) and Wang and Ročková (2020)) and continuity of the multiplication constant, we can essentially regard the rate as uniform. Hereafter, we thereby abuse the notation and tacitly assume that δ_n is the worst rate over all $\theta \in \Theta \subset \mathbb{R}^d$, i.e. the rate with the largest multiplication constant.*

Next, we investigate convergence around the *actual* KL divergence $K(p_0, p_\theta)$. The next lemma will be utilized later in the proof of Theorem 7. However, it is of independent interest as it shows how the speed at which the joint error probability (accounting for randomness of both the observed and fake data $(\boldsymbol{X}, \tilde{\boldsymbol{X}})$) decays in terms of the estimation error. Below, the probability P corresponds to $P_0^{(n)} \otimes \tilde{P}^{(m)}$, where $\tilde{P}^{(m)}$ is the measure for $\tilde{\boldsymbol{X}}$ (see Remark 1).

Theorem 5. *For a given $\theta \in \Theta$, we define for $u > 0$ and $\delta_n > 0$ as in Lemma 2 and for an arbitrarily slowly increasing sequence $C_n > 0$*

$$\rho_{n,\theta}(u; C_n; \delta_n) \equiv P\bigg(\Big| \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) - K(p_0, p_\theta) \Big| > 2u, d_\theta(\hat{D}^{\theta}_{n,m}, D^{\theta}) \leq C_n \delta_n \bigg). \tag{2.12}$$

Under Assumptions 1 and 2, we then have

$$\rho_n(u; C_n; \delta_n) \equiv \sup_{\theta \in \Theta} \rho_{n,\theta}(u; C_n; \delta_n) = O\Big(\frac{C_n \delta_n}{u} + \frac{1}{n u^2} \Big).$$

The proof can be found in Section 2.7.3.1. Note that the intersecting event above has probability converging to one, according to Lemma 2.

Remark 6 (Neural Network Sieve)**.** *The results discussed in this section apply to any nonparametric sieve discriminator that satisfies the entropy conditions. To facilitate understanding of how the rate of convergence δ can be affected by the dimension of the data space d, the smoothness of the density, and the choice of the classifier, we provide a discussion on neural network sieve specifically here.*

Borrowing from the idea of Bauer and Kohler (2019), the convergence rate of the appropriately configured neural network discriminator depends only on the low underlying dimension of the oracle discriminator, however large the ostensible dimension of the data is. Intuitively, the low-dimensional structure can be described as follows. The log-likelihood ratio $\log(p_0/p_\theta)$ *takes a d-dimensional input X as an argument, which may be large. If this function admits a representation as a nested composition of smooth functions, each of which takes a possibly smaller number d^* of arguments, the neural network sieve can adapt to this underlying structure and converges faster than the traditionally proven rate.*

In particular, if $d^* < 2p$, we have $\delta_n = o_P(n^{-1/4})$, which is often the desired rate for the nonparametric estimator of a nuisance parameter. Additionally we show that one can obtain $\delta_n \lesssim n^{-2/5}$ for the binary choice model with logistic errors and δ_n arbitrarily close to $n^{-1/2}$ for the discretely sampled Brownian motion model. The detailed characterizations of the low underlying dimension d^* and two examples can be found in Section 2.7.1.

Finally, with Assumption 2, the convergence rate δ_n translates into the convergence rate of the Kullback-Leibler estimator in Lemma 3. Thus, in smooth low-dimensional hierarchical models, our Kullback-Leibler estimator converges reasonably fast even when the nominal dimension of the data is large.

2.3.2 Posterior Concentration Rate

Concentration rates are typically quantified in terms of a prior concentration (measured in terms of a combination of the KL divergence and the KL variation) and the entropy of the model. We have a similar prior mass condition (see (8.4) in Section 8.2 in Ghosal and Van der Vaart (2017)). We denote the KL-neighborhood of p_0 by

$$B_2(p_0, \epsilon) = \{\theta : K(p_0, p_\theta) < \epsilon^2\}. \tag{2.13}$$

Assumption 3 (Prior Mass)**.** *There exist some constants $\kappa > 0$ and $\xi > 0$ such that for every $0 < \epsilon < \xi$ and some constant $C > 0$, the prior probability satisfies $\Pi[B_2(p_0, \epsilon)] \geq C\epsilon^\kappa$.*

Next, we assume that the parameter θ is identifiable in the sense that the KL divergence is locally compatible with the Euclidean norm. This assumption is adopted from Assumption 3(ii) of Frazier et al. (2018).

Assumption 4 (Identification)**.** *The density function p_θ is continuous in θ and for every θ in some open neighborhood of θ_0 satisfies*

$$\|\theta - \theta_0\| \leq L \times K(p_0, p_\theta)^\alpha$$

for some $L > 0$ and $\alpha > 0$.

Similarly as in (5.1) in Kleijn and van der Vaart (2006), Assumption 4 ensures posterior concentration around θ_0 when $K(p_0, p_\theta) \to 0$. This holds for many distributions. For example, for the exponential distribution with a rate parameter θ, we have $K(p_0, p_\theta) = \frac{\theta}{\theta_0} - \log(\frac{\theta}{\theta_0}) - 1$. Since $\log(1+x) = x - \frac{x^2}{2} + o(x^2)$ when $x \to 0$, we have $K(p_0, p_\theta) \geq \frac{1}{2}\theta_0^{-2}(\theta - \theta_0)^2$. For multivariate normal distribution with a known variance Σ and an unknown location μ, we have $K(p_0, p_\theta) = \frac{1}{2}(\mu - \mu_0)\Sigma^{-1}(\mu - \mu_0) \geq \frac{1}{2}\rho(\Sigma)^{-1} \|\mu - \mu_0\|^2$, where $\rho(\Sigma)$ is the spectral radius, i.e., the largest eigenvalue, of a matrix Σ.

First, we focus on the uniform kernel $T_\epsilon(x) = \mathbb{I}(|x| \leq \epsilon)$ used in Algorithm 1. Recall (from Remark 1) that $\tilde{\boldsymbol{X}}^\theta = g_\theta(\tilde{\boldsymbol{X}})$ for some suitable mapping $g_\theta(\tilde{\boldsymbol{X}})$ where $\tilde{\boldsymbol{X}} \sim \tilde{P}^{(m)}$. The ABC joint posterior (2.2) is a weighted aggregation of uniform kernels, i.e

$$\hat{\pi}^{AR}\left(\theta, \tilde{\boldsymbol{X}} \mid \hat{K}[\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})] \leq \epsilon\right) = \frac{\tilde{\pi}(\tilde{\boldsymbol{X}})\pi(\theta)\mathbb{I}\big(\hat{K}[\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})] \leq \epsilon\big)}{\int \tilde{\pi}(\tilde{\boldsymbol{X}})\pi(\theta)\mathbb{I}\big(\hat{K}[\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})] \leq \epsilon\big)\mathrm{d}\tilde{\boldsymbol{X}}\mathrm{d}\theta}, \tag{2.14}$$

which yields the following ABC posterior distribution

$$\hat{\Pi}^{AR}_{\epsilon_n}(A \mid \boldsymbol{X}) = \frac{\int_{\theta \in A} \pi(\theta)\tilde{P}^{(m)}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n\big)\mathrm{d}\theta}{\int_\Theta \pi(\theta)\tilde{P}^{(m)}\big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n\big)\mathrm{d}\theta} \quad \text{for a Borel-measurable } A \subset \Theta. \tag{2.15}$$

The following theorem (a modification of Theorem 1 in Frazier et al. (2018)) quantifies the concentration rate in terms of the tolerance threshold ϵ_n as well as the rate at which the classification-based KL estimator can estimate $\mathbb{P}_n \log(p_0/p_\theta)$ (as formulated in (2.11)).

Theorem 7. *Let Assumptions 1, 2 and 3 hold and take δ_n as in* (2.11) *in Lemma 3. Then, as $n \to \infty$ and with $\epsilon_n = o(1)$ such that $n\epsilon_n^2 \to \infty$ and $C_n\delta_n = o(\epsilon_n)$ for some arbitrarily slowly increasing sequence $C_n > 0$ we have*

$$P_0^{(n)}\Pi[K(p_0, p_\theta) > \lambda_n \mid \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n] = o(1), \tag{2.16}$$

where $\lambda_n = \epsilon_n + M_nC_n\delta_n\epsilon_n^{-\kappa} + \sqrt{M_n}n^{-1/2}\epsilon_n^{-\kappa/2}$ for some arbitrarily slowly increasing sequence $M_n > 0$. Moreover, if Assumption 4 also holds, as $n \to \infty$, we have

$$P_0^{(n)}\Pi[\|\theta - \theta_0\| > L\lambda_n^\alpha \mid \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n] = o(1). \tag{2.17}$$

The proof of the theorem is provided in Section 2.7.3.2. Thus, the convergence rate of our ABC posterior depends on three components: the accept-reject threshold ϵ_n, the estimation error of the KL estimator δ_n and the rate of discrepancy $n^{-1/2}$ between the empirical and

true KL divergence. Since δ_n will typically be greater than the parametric rate $n^{-1/2}$, the overall convergence rate is then driven by $\lambda_n = \epsilon_n + \widetilde{M}_n \delta_n \epsilon_n^{-\kappa}$, where $\widetilde{M}_n$ is an arbitrarily slowly increasing sequence.

In practice, it is unclear how to properly choose ϵ_n. In Algorithm 2, we proposed to weight the draws using a scaled exponential kernel $\exp(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta))$. We denote the ABC posterior under the exponential kernel as $\hat{\Pi}^{EK}(\cdot \mid \boldsymbol{X})$ where

$$\hat{\Pi}^{EK}(A \mid \boldsymbol{X}) = \frac{\int_A \tilde{P}^{(m)} \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big)\pi(\theta)\mathrm{d}\theta}{\int_\Theta \tilde{P}^{(m)} \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big)\pi(\theta)\mathrm{d}\theta}. \tag{2.18}$$

To gain more insights into the ABC posterior behavior under the exponential kernel, we take a closer look at the "likelihood function" above

$$\tilde{P}^{(m)} \exp\big(-n\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)\big) = \frac{p_\theta^{(n)}}{p_0^{(n)}} \tilde{P}^{(m)} \mathrm{e}^{u_\theta},$$

where $u_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) = -n \times \Big(\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) - \mathbb{P}_n \log \frac{p_0}{p_\theta}\Big)$. From the equations above, we can write

$$\hat{\pi}^{EK}(\theta \mid \boldsymbol{X}) \propto p_\theta^{(n)}(\boldsymbol{X}) \times \mathrm{e}^{\hat{u}_\theta(\boldsymbol{X})} \times \pi(\theta) \quad \text{with} \quad \hat{u}_\theta(\boldsymbol{X}) = \log \int \mathrm{e}^{u_\theta(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)} \mathrm{d}\tilde{P}^{(m)}(\tilde{\boldsymbol{X}}). \tag{2.19}$$

When $\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta)$ is the classification-based estimator, $\hat{u}_\theta(\boldsymbol{X})$ can be related to the random generator setting of the Metropolis-Hastings MHC algorithm in Kaji and Ročková (2022) which has (2.19) as its stationary distribution. Similarly as in Appendix Section 5 of Kaji and Ročková (2022), we can regard the posterior approximation in (2.19) as a posterior $\hat{\pi}^{EK}(\theta \mid \boldsymbol{X}) \propto q_\theta^{(n)} \tilde{\pi}(\theta)$ under a misspecified likelihood

$$q_\theta^{(n)} = \frac{p_\theta^{(n)}(\boldsymbol{X}) \mathrm{e}^{\hat{u}_\theta(\boldsymbol{X})}}{C_\theta} \quad \text{where} \quad C_\theta = \int_{\mathcal{X}} p_\theta^{(n)}(\boldsymbol{X}) \mathrm{e}^{\hat{u}_\theta(\boldsymbol{X})} \mathrm{d}\boldsymbol{X} \tag{2.20}$$

and a modified prior $\tilde{\pi}(\theta) \propto \pi(\theta) C_\theta$. Since the likelihood is misspecified, the ABC posterior

concentrates around a projection point θ^* defined as

$$\theta^* = \arg\min_{\theta\in\Theta} -P_0^{(n)} \log[q_\theta^{(n)}/p_0^{(n)}], \tag{2.21}$$

which corresponds to the mis-specified model that is closest to $P_0^{(n)}$ in the KL sense (Kleijn and van der Vaart, 2006). Kaji and Ročková (2022) study posterior concentration of (2.19). Unlike in Theorem 7, the posterior concentration rate here depends both on the estimation error δ_n and the actual concentration rate of the true posterior, not the acceptance threshold.

Remark 8 (Vanishing Bias)**.** *To better understand the severity of the centering bias of the misspecified model, we note*

$$\begin{aligned} -P_0^{(n)} \log[q_{\theta^*}^{(n)}/p_0^{(n)}] \le -P_0^{(n)} \log[q_{\theta_0}^{(n)}/p_0^{(n)}] &= P_0^{(n)} \log \frac{p_0^{(n)}}{p_0^{(n)} \mathrm{e}^{\hat{u}_{\theta_0}(\boldsymbol{X})}/C_{\theta_0}} \\ &= \log C_{\theta_0} - P_0^{(n)} \hat{u}_{\theta_0}(\boldsymbol{X}) = \log P_0^{(n)} \mathrm{e}^{\hat{u}_{\theta_0}(\boldsymbol{X})} - P_0^{(n)} \hat{u}_{\theta_0}(\boldsymbol{X}). \end{aligned}$$

This is essentially the Jensen gap. If we have this Jensen gap vanishing when $n \to \infty$, then we can conclude that the centering bias is also vanishing, and the ABC posterior in (2.18) *will eventually concentrate at the right location.*

2.3.3 Shape of the Limiting ABC Posterior Distribution

We now analyze the limiting shape of $\hat{\Pi}_{\epsilon_n}^{AR}(\cdot \mid \boldsymbol{X})$ defined in (2.15). We focus on the case when $\epsilon_n \gg \delta_n^{1/(\kappa+1)}$, where κ was defined in Assumption 3, since the posterior is not guaranteed to converge when the decision threshold ϵ_n is smaller than the estimation error δ_n of the KL estimator.

Assumption 5. *Assume that for every $\varepsilon > 0$, we have $\inf_{\|\theta-\theta_0\|>\varepsilon} K(p_0, p_\theta) > 0$. In addition, assume that $\log p_\theta$ is twice differentiable with respect to θ and that, for every θ in some neighborhood of θ_0, the remainder of the second order Taylor expansion of $K(p_0, p_\theta) =$*

$P_0 \log \frac{p_0}{p_\theta}(x)$ around θ_0 is comparatively small relative to the second-order term, i.e.

$$\begin{aligned} K(p_0, p_\theta) &= \nabla_{\theta=\theta_0} K(p_0, p_\theta)(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)' \nabla^2_{\theta=\theta_0} K(p_0, p_\theta)(\theta - \theta_0)\{1 + o(1)\} \\ &= \frac{1}{2}(\theta - \theta_0)' I(\theta_0)(\theta - \theta_0)\{1 + o(1)\}, \end{aligned}$$

where $I(\theta) = \nabla^2_\theta K(p_0, p_\theta) = P_0[(\nabla_\theta \log p_\theta)^2]$ is the Fisher information matrix.

When Assumption 5 is satisfied, the condition in Assumption 4 is immediately satisfied as well and the identification of θ_0 is guaranteed. In the open-neighborhood of θ_0, since we have

$$(\theta - \theta_0)' I(\theta_0)(\theta - \theta_0) = 2K(p_0, p_\theta)\{1 + o(1)\},$$

and $I(\theta_0)$ is positive definite, the convergence in the bilinear form $(\theta - \theta_0)' I(\theta_0)(\theta - \theta_0)$ ensures the convergence of the parameters.

Theorem 9. *Assume that the prior function $\pi(\cdot)$ is continuous around θ_0. Then, under Assumptions 1, 2, 3 and 5, if $\lim_n \delta_n / \epsilon_n^{\kappa+1} \to 0$, the average posterior distribution of $\epsilon_n^{-1/2}(\theta - \theta_0)$ converges to the uniform distribution over the ellipse $\{w : w' I(\theta_0) w \leq 2\}$ where $I(\theta)$ is the Fisher information matrix defined in Assumption 5. In particular, as $n \to 0$, we have*

$$P_0^{(n)} \int f\Big(\epsilon_n^{-1/2}(\theta - \theta_0)\Big) \hat{\Pi}^{AR}_{\epsilon_n}(\theta \mid \boldsymbol{X}) \to \int_{u' I(\theta_0) u \leq 2} f(u) \mathrm{d}u \Big/ \int_{u' I(\theta_0) u \leq 2} \mathrm{d}u$$

for every continuous and bounded function $f(\cdot) : \mathcal{X} \to \mathbb{R}$.

The proof is provided in Section 2.7.3.3.

Remark 10. *Theorem 9 is adapted from the case (i) in Theorem 2 of Frazier et al. (2018). We only consider situations when $\epsilon_n \gg \delta_n^{1/(\kappa+1)}$ with the prior shrinkage parameter κ defined in Assumption 3. In other words, we assume that the ABC decision threshold ϵ_n is*

dominating both the estimation error δ_n and the asymptotic error $n^{-1/2}$ and, thereby, determines the posterior concentration rate. It is not entirely obvious how the posterior would behave when the threshold ϵ_n shrinks faster than the estimation error δ_n.

For the asymptotic behavior of the ABC posterior (2.19) induced by the exponential kernel, we resort to BvM characterizations under misspefication in LAN models (Kleijn and van der Vaart, 2012). When the posterior (2.19) concentrates around θ^* in (2.21) at the rate ϵ_n^*, one can show under a suitable LAN condition that the approximate posterior converges to a sequence of normal distributions in total variation at the rate ϵ_n^*. The centering and the asymptotic covariance matrix both depend on θ^*. The formal statement and the proof is in Kaji and Ročková (2022).

Although we only consider the case where the model is correctly specified in our paper, our results can be extended to the mis-specified model along the lines of Frazier et al. (2019).

2.3.4 ABC Kernels and Model Misspecification

Although the exponential kernel ABC (Algorithm 2) obviates the need for the threshold ϵ_n and performs very well in our examples, the Accept/Reject kernel ABC (Algorithm 1) may perform better under mis-specification (see Remark 2.2.2). When the model is mis-specified, the posterior under Algorithm 1 will converge to the "pseudo-true" value, which is the point that minimizes the distance between summary statistics within the mis-specified class. Since our summary statistic is replaced with KL divergence, this point will coincide with the KL projection in our case. The exponential kernel will also concentrate around a certain KL projection but its bias will now be compounded by the influence of both $P_0 \notin \mathcal{P} = \{\theta \in \Theta : P_\theta\}$ and the exponential tilt $e^{\hat{u}_\theta(\boldsymbol{X})}$ arising from the approximation error of the KL estimator. We would thereby expect a bigger bias from the exponential kernel when the model is misspecified.

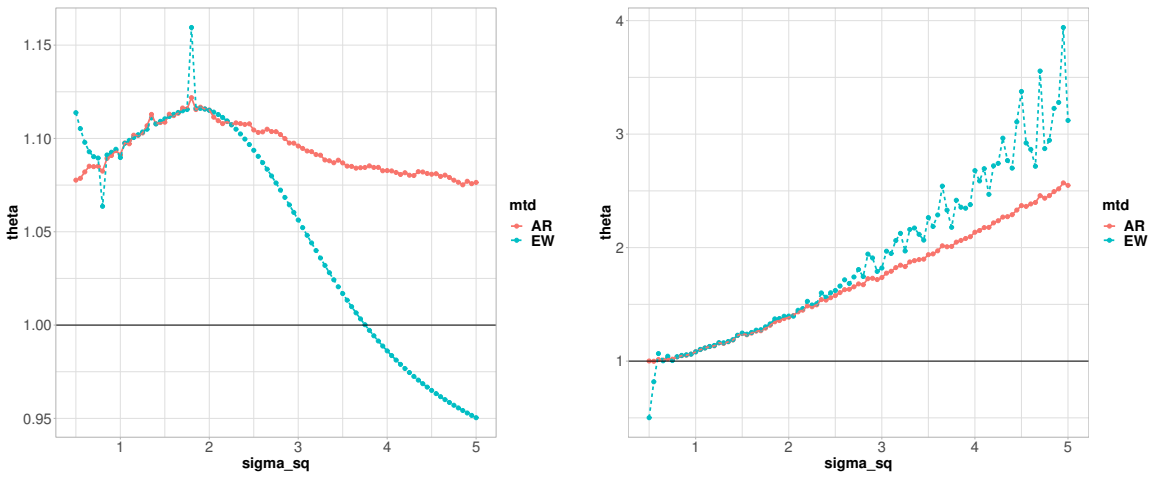
We illustrate the intuition above with a toy example. We use the simple example proposed

in Frazier et al. (2019) where the assumed data-generating process (DGP) is i.i.d. $\mathcal{N}(\theta, 1)$, but the actual DGP for $\boldsymbol{X}$ is i.i.d. $N(\theta, \sigma^2)$ for $\sigma^2 \neq 1$. In other words, the assumed DGP has an incorrect specification of the variance of the observed data. We consider the oracle logistic classifier built on $\boldsymbol{X}$ and the quadratic term $\boldsymbol{X}^2$ for our KL estimator. We fix $\theta = 1$ and simulate $\boldsymbol{X}$ (using $n = 100$) with respect to different values of σ^2, ranging from 0.5 to 5 with evenly spaced increments of 0.05. Using one common set of latent variables $\nu_i \sim \mathcal{N}(0, 1)$, the observed data is generated as $X_i = 1 + \nu_i \sigma$ for each σ^2. The prior belief is $\theta \sim \mathcal{N}(0, 25)$, and we implement ABC methods with $N = 100\,000$ pseudo datasets. The parameter draws and the latent variable datasets are the same across the different values of σ^2. To explore how the tail behavior influences the ABC bias, we also include misspecified lognormal distributions using the same setup.

Figure 2.1(a) compares the posterior mean of the accept-reject ABC (AR) and the ABC with exponential weighting (EW) across different values of σ^2. We can see that both AR and EW have a relatively small bias in estimating θ when the misspecification level in σ^2 varies. Since the logistic regression classifier on $\boldsymbol{X}$ and $\boldsymbol{X}^2$ is almost the "oracle" discriminator for gaussian distributions, the error of the KL estimator should be minimal. Nevertheless, the posterior mean of EW exhibits a downward moving trend when the level of misspecification increases. For the heavy-tailed lognormal distribution shown in Figure 2.1(b), although the posterior mean of AR does shift away from true value $\theta = 1$ as the degree of model misspecification increases, the posterior mean of EW shifts away from $\theta = 1$ at a faster speed.

2.4 Simulations

In this section, we illustrate our approach and make comparisons with other likelihood-free inference techniques. Within our KL-ABC framework, we include two types of KL estimators. One is obtained with the logit discriminator score, which we refer to as KL



(a) normal (b) lognormal

Figure 2.1: ABC posterior under misspecified models.

estimation via classification (KLC), and the other one is estimated via the kNN method (kNN) with $k = 1$ (Jiang et al., 2018). For both estimators, we aggregate ABC samples with the accept-reject kernel as in Algorithm 1 and the exponential kernel as in Algorithm 2. The latter will be denoted with a suffix 'exp', e.g. KLC-exp or kNN-exp. The discriminator used for each dataset will be specified later.

The ABC discrepancy metrics we choose for comparisons are (1) the classification accuracy (CA) (Gutmann et al., 2018) defined as (2.10); (2) the 2-Wasserstein (W2) distance under the Euclidean metric (Bernton et al., 2019) defined as $\mathrm{W2}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) = \min_\gamma [\sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} ||X_i - \tilde{X}_j^\theta||^2]^{1/2}$ s.t. $\gamma' \mathbf{1}_m = \mathbf{1}_n, \gamma' \mathbf{1}_n = \mathbf{1}_m$ with $0 \leq \gamma_{ij} \leq 1$; (3) ℓ_2-distance between summary statistics (SS) and we use the semi-automatic (SA) method (Fearnhead and Prangle, 2011) if no candidate summary statistics are given; (4) approximated posterior mean of the parameters predicted by trained deep neural network (DNN) (Jiang et al., 2017b); (5) Maximum Mean (MM) discrepancy (Park et al., 2016) defined as $MM(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) = \frac{1}{n(n-1)} \sum_{i \neq j} k(X_i, X_j) + \frac{1}{m(m-1)} \sum_{i \neq j} k(\tilde{X}_i^\theta, \tilde{X}_j^\theta) - \frac{2}{nm} \sum_{i,j} k(X_i, \tilde{X}_j^\theta)$ where $k(\cdot, \cdot)$ is a Gaussian kernel with the bandwidth being the median of $\{\|X_i - X_j\| : i \neq j\}$;(6) a V-statistic estimator of Energy Statistics (ES) proposed by Nguyen et al. (2020); (7) auxiliary

likelihood $AL = \frac{1}{m} \ln p_A(\tilde{\boldsymbol{X}}^\theta \mid \hat{\phi}(\tilde{\boldsymbol{X}}^\theta)) - \frac{1}{m} \ln p_A(\tilde{\boldsymbol{X}}^\theta \mid \hat{\phi}(\boldsymbol{X}))$ proposed by Drovandi et al. (2011), where $p_A(x \mid \phi)$ is a d-dimensional Gaussian distribution with ϕ being the sample mean and covariance. For the classification accuracy (CA), we use the same discriminator as the one in KL estimation. For the DNN approach, we deploy a 3-layer DNN with 100 neurons and hyperbolic tangent (tanh) activation on each hidden layer. The model is trained on 10^6 samples and validated on 10^5 samples, with early stopping once the validation error starts to increase. In each experiment, unless otherwise noted, we set the tolerance threshold ϵ adaptively such that 1 000 of 100 000 (i.e. the top 1%) proposed ABC samples are accepted.

2.4.1 M/G/1-Queuing Model

Because queuing models are usually easy to simulate from, but have no tractable likelihoods, they have been frequently used as test cases in the ABC literature, see e.g. Fearnhead and Prangle (2011) and Bernton et al. (2019). Here, we choose the same setup as in Jiang et al. (2017b). Each datum is a 5-dimensional vector consisting of the first five inter-departure times $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})'$. In the model, the service times u_{ik} follow a uniform distribution $U[\theta_1, \theta_2]$, and the arrival times w_{ik} are exponentially distributed with the rate θ_3. We only observe the interdeparture times x_i, given by the process $x_{ik} = u_{ik} + \max(0, \sum_{j=1}^{k} w_{ij} - \sum_{j=1}^{k-1} x_{ij})$. We perform ABC on $n = 500$ observed samples which are generated from the true parameter $\theta_0 = (1, 5, 0.2)$. The prior on $(\theta_1, \theta_2 - \theta_1, \theta_3)$ is uniform on $[0, 10]^2 \times [0, 0.5]$. [3]

Regarding the choice of the discriminator, we consider both the Random Forest (RF) classifier and a ℓ_1-penalized logistic classifier (LRD). For the former, we use the default setting in the `R` package `randomForest`. We also denote CA calculated from RF classifier as RF-CA. For the latter, we implement the discriminator with R package `glmnet`, and the model is built on degree-2 polynomials of the data, including quadratic and interaction

3. We place the uniform prior on $\theta_2 - \theta_1$ instead of θ_2, since θ_2 must be larger than θ_1. This is used in Jiang et al. (2017b).

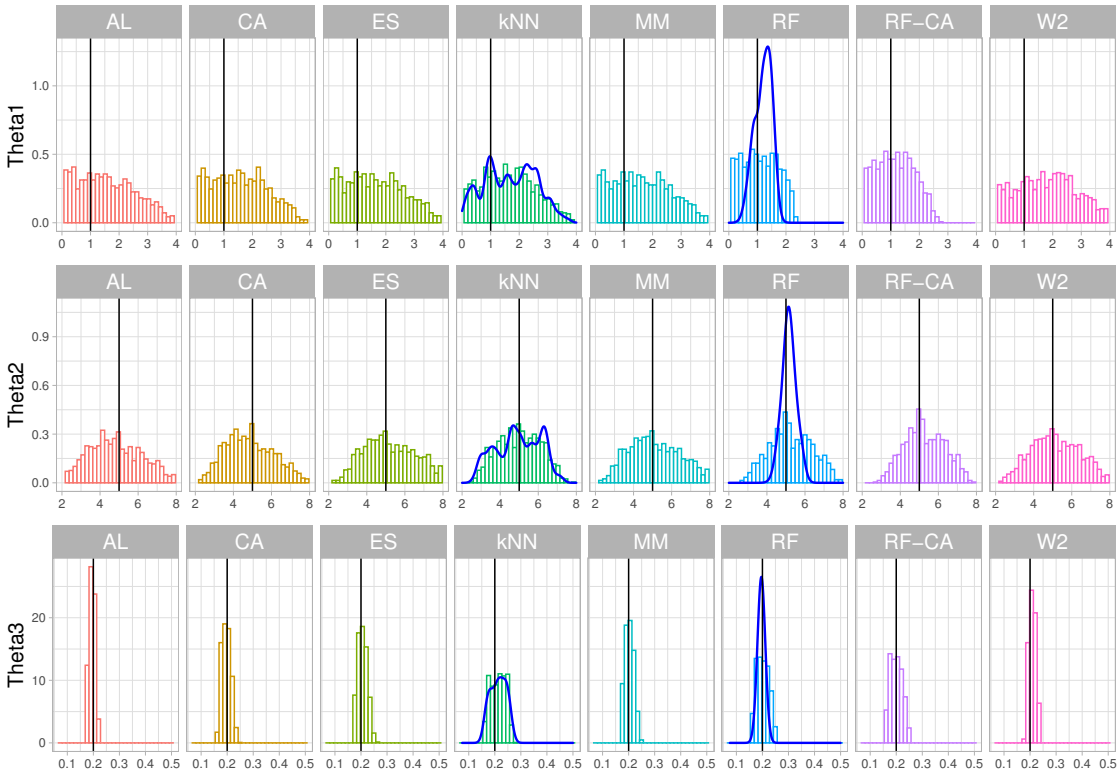


Figure 2.2: ABC reconstructed posteriors under M/G/1-queuing model with $\theta_0 = (1, 5, 0.2)'$. The vertical black lines mark the true values. The blue curves in kNN and RF boxes mark a smoothed density calculated from the exponential kernel.

terms, with penalty parameter λ is selected via 5-fold cross-validation. We find that RF outperforms LRD.

The shape of the ABC posteriors is given in Figure 2.2. The plot reveals that, among the three parameters, θ_1 is the hardest one to estimate where all methods, except for RF and its variants, gave relatively flat posterior estimates. Regarding θ_2, all methods seem to center well around the truth. RF-exp provides the tightest estimation. Regarding θ_3, all except kNN return spiky posterior. Next, we repeat the experiments on 10 different datasets, and summarize the average squared estimation errors and the width of the 95% credible intervals in Table 2.1. Overall, we see that RF and RF-exp are able to correctly identify the right locations of the parameters and outperform the kNN estimator, with RF-exp providing the tightest credible interval. The method RF-CA tends to give very similar results to RF, which

Method (scale)	$\theta_1 = 1$ $(\hat{\theta}_1 - \theta_1)^2$	95% CI width	$\theta_2 = 5$ $(\hat{\theta}_2 - \theta_2)^2$	95% CI width	$\theta_3 = 0.2$ $(\hat{\theta}_3 - \theta_3)^2$ (10^{-4})	95%CI width
RF	**0.008**	2.044	0.038	4.304	0.571	0.084
RF -exp	0.035	**0.959**	0.055	**1.473**	0.307	**0.028**
LRD	0.197	3.116	0.217	4.599	0.308	0.064
LRD-exp	0.169	2.851	0.312	3.708	**0.234**	0.030
kNN	0.525	3.135	0.106	3.986	3.659	0.094
kNN-exp	1.057	2.664 (0.8)	0.431	3.331 (0.9)	2.634	0.072
DNN	0.150	3.350	1.100	7.361	81.931	0.328
ES	0.255	3.321	0.087	5.280	1.176	0.070
CA	0.191	3.107	0.235	4.602	0.280	0.064
RF-CA	0.014	2.194	**0.036**	4.104	0.404	0.084
AL	0.259	3.423	0.057	5.651	0.575	0.044
SA	0.180	2.457	0.355	3.514	45.297	0.446
W2	0.595	3.631	0.039	4.871	3.846	0.052 (0.8)

Table 2.1: ABC performance on the M/G/1 queuing model over 10 repetitions, with top 1% ABC samples selected. Most of the 95% CIs have full coverage with the rest having their coverage marked in brackets. The bold fonts mark the best model in each metric.

is not entirely unexpected since they are derived from the same discriminators.

2.4.2 Lotka-Volterra Model

The Lotka-Volterra (LV) predator-prey model (Wilkinson, 2018) describes population evolutions in ecosystems where predators interact with prey. It is one of the classical stochastic kinetic network model examples. The state of the population is prescribed deterministically via a system of ordinary differential equations (ODEs). Inference for such models is challenging because the transition density is intractable. However, simulation from the model is possible, which makes it a natural candidate for ABC methods.

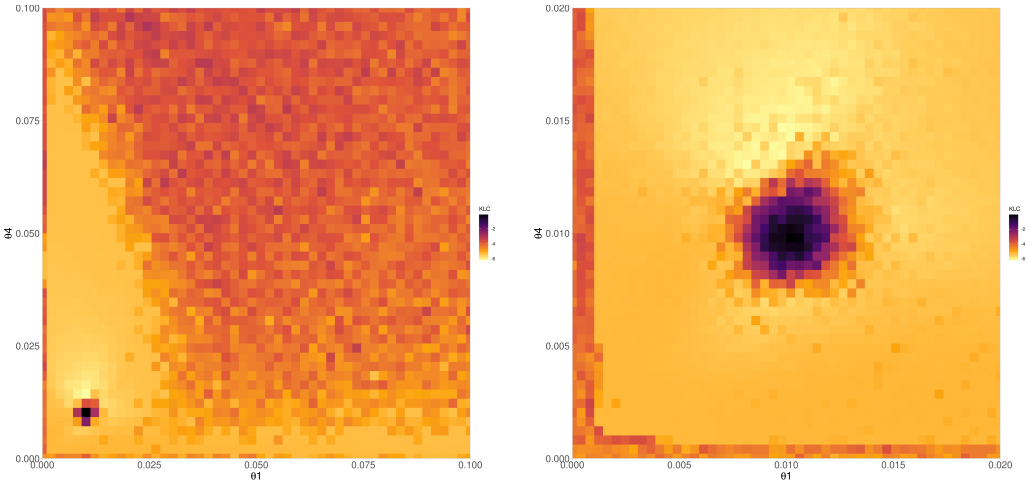
The model monitors population sizes of predators X_t and prey Y_t over time t. The changes in states are determined by four parameters $\theta = (\theta_1, \ldots, \theta_4)'$ controlling: (1) the rate $r_1^t = \theta_1 X_t Y_t$ of a predator being born; (2) the rate $r_2^t = \theta_2 X_t$ of a predator dying; (3) the rate $r_3^t = \theta_3 Y_t$ of a prey being born; (4) the rate $r_4^t = \theta_4 X_t Y_t$ of a prey dying. Given the initial population sizes (X_0, Y_0) at time $t = 0$, the dynamics can be simulated using

the Gilliespie algorithm (Gillespie, 1977). The algorithm samples times to an event from an exponential distribution (with a rate $\sum_{j=1}^{4} r_j^t$) and picks one of the four reactions with probabilities proportional to their individual rates r_j^t.

We use the same simulation setup as Kaji and Ročková (2022). Each simulation is started at $X_0 = 50$ and $Y_0 = 100$ and state observations are recorded every 0.1 time units for a period of 20 time units, resulting in a series of $T = 201$ observations each. The real data ($n = 20$ time series) are generated with true values $\theta_0 = (0.01, 0.5, 1, 0.01)'$. The predator-prey interaction dynamic is very sensitive to parameter changes. For example, Figure 4 of Kaji and Ročková (2022) shows that a slight perturbation in θ_2 leads to significant changes in the population renewal cycle. We rely on the ability of the discriminator to tell such different patterns apart. The sensitivity of the model to minor parameter changes is confirmed with heat-map plots of the estimated KL divergence as a function of $(\theta_1, \theta_4)'$ in Figure 2.3. Figure 2.3(a) provides a plot of the estimated KL over the region $[0, 0.1]^2$ where, apparently, the majority of the region is flat and uninformative with a sharp spike around the true values at $\theta_1 = \theta_4 = 0.01$. We thus narrow the investigation down to a smaller region $[0, 0.02]^2$ in Figure 2.3(b). Again, the curvature in the estimated KL around the truth is quite steep. This may pose some issues for Metropolis-Hasting algorithms, since the majority of the prior region is uninformative and improper initialization could lead to extremely slow convergence.

Previous ABC analyses of this model suggested various summary statistics including the mean, log-variance, autocorrelation (at lag 1 and 2) of each series as well as their cross-correlation (Papamakarios and Murray, 2016). For the discriminator of our method, we choose the ℓ_1-penalized (LASSO) logistic regression classifier (LRD) with $m = n$ and with a penalty λ selected via 5-fold cross-validation (as implemented in the R package `glmnet`), as well as a random forest classifier (RF).

Similar to Kaji and Ročková (2022), we use an informative prior $\theta \in U(\Xi)$ with a restricted domain $\Xi = [0, 0.1] \times [0, 1] \times [0, 2] \times [0, 0.1]$ so that the computation is more economic



(a) $(\theta_1, \theta_4) \in (0, 0.1)^2$ (b) $(\theta_1, \theta_4) \in (0, 0.02)^2$

Figure 2.3: Estimated $\hat{K}(\mathbf{X}, \widetilde{\mathbf{X}^\theta})$ (via Classification) over a grid of (θ_1, θ_4) values. The other two parameters are fixed at $\theta_2 = 0.5, \theta_3 = 1$.

and efficient. A typical snapshot of the ABC posteriors is plotted in Figure 3.4. From the plot, we can see that the difficulty in estimating different parameters varies a lot and θ_3 is the most challenging among all. While other methods give relatively flat posteriors for θ_3, our method (RF) combined with the scaled exponential kernel identifies the correct location of the parameter with a much tighter posterior. For the DNN approach, the posteriors seem to be very flat and the estimation for θ_2 is biased. Considering its heavy computation costs, we exclude this method in the repetition experiment. The average performance of ABC methods under this model is summarized in Table 2.2. We can see that RF with the exponential kernel gives the tightest CI most of the time, while maintaining relatively small estimation errors.

2.5 Empirical Analysis

We further demonstrate our approach on the nontrivial problem of estimating stock volatility using merely daily observations on high, low and closing prices. All of these price observations are typically available to investors. We use a similar data generating process as in Magdon-

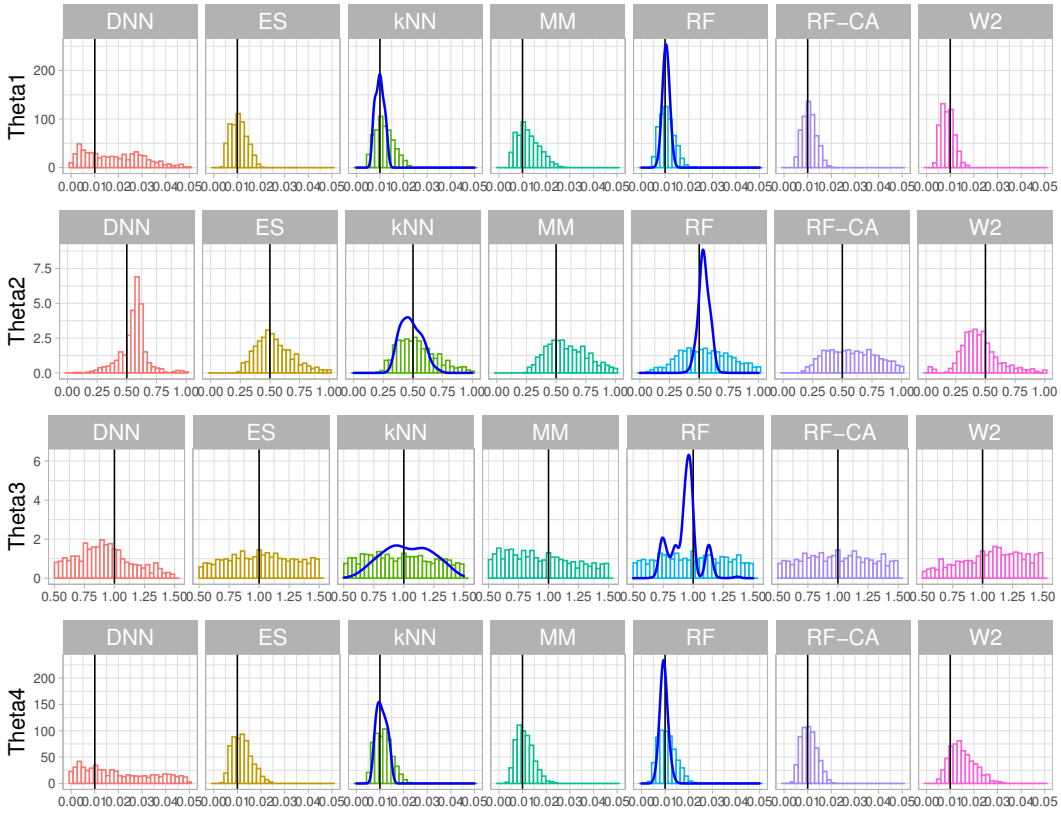


Figure 2.4: ABC posteriors for the Lotka-Volterra model with $\theta_0 = (0.01, 0.5, 1, 0.01)'$. The black vertical lines mark the true parameter values. The blue curves in kNN and RF boxes represent the smoothed density calculated from the exponential kernel. The ABC posteriors are plotted from the top 1% out of 10^5 samples.

Ismail and Atiya (2003), assuming that the assets follow a Brownian motion with a constant drift and volatility. In particular, suppose that the log-price processes $X_j(t), i = 1, \ldots, d$, are correlated Brownian motions, that is $E[X_i(s)X_j(t)] = \sigma_{ij} \min\{s, t\}$, and that the joint movement of the log-price processes $\boldsymbol{X}(t) = (X_1(t), \ldots, X_d(t))'$ follows a multivariate Brownian motion as

$$d\boldsymbol{X}(t) = \boldsymbol{\mu} dt + \Sigma d\boldsymbol{W}(t), \tag{2.22}$$

Method (scale)	$\theta_1 = 0.01$ $(\hat{\theta}_1 - \theta_1)^2$ (10^{-5})	95% CI width	$\theta_2 = 0.5$ $(\hat{\theta}_2 - \theta_2)^2$	95% CI width	$\theta_3 = 1$ $(\hat{\theta}_3 - \theta_3)^2$ (10^{-2})	95% CI width	$\theta_4 = 0.01$ $(\hat{\theta}_4 - \theta_4)^2$ (10^{-5})	95% CI width
RF	0.118	1.116	0.389	0.719	**0.037**	0.938	0.051	1.290
RF-exp	**0.015**	**0.354**	0.092	**0.191**	0.272	**0.483**	0.037	**0.493**
LRD	42.949	4.474	1.688	0.626	0.119	0.945	17.159	4.720
LRD-exp	0.066	**0.354**	0.247	0.285	0.405	0.643	**0.021**	0.626
KL	0.563	1.462	0.107	0.602	0.054	0.934	0.203	1.327
KL-exp	0.049	0.544 (0.9)	**0.044**	0.243 (0.9)	0.346	0.599 (0.9)	0.066	0.616 (0.9)
CA	36.477	4.375	1.602	0.629	0.118	0.945	15.684	4.622
RF-CA	0.141	1.061	0.699	0.700	0.042	0.933	0.056	1.181
ES	0.101	1.233	0.203	0.573	0.045	0.923	0.341	1.443
W2	1.265	1.666	0.478	0.599	0.011	0.917	4.080	2.051
SS	1.639	1.777	2.260	0.721	1.019	0.901	3.767	1.660
MM	0.824	1.727	0.959	0.601	0.414	0.932	0.074	1.294

Table 2.2: ABC performance evaluated on the Lotka-Volterra Model, averaged over 10 repetitions, with the top 1% ABC samples selected. Most of the 95% CIs have full coverage with the rest having their coverage marked in brackets. The bold fonts mark the best model in each metric.

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)'$ and $\Sigma = [\sigma_{ij}]_{1 \le i,j \le d}$ denote the drift and the volatility of the log processes, respectively. We write

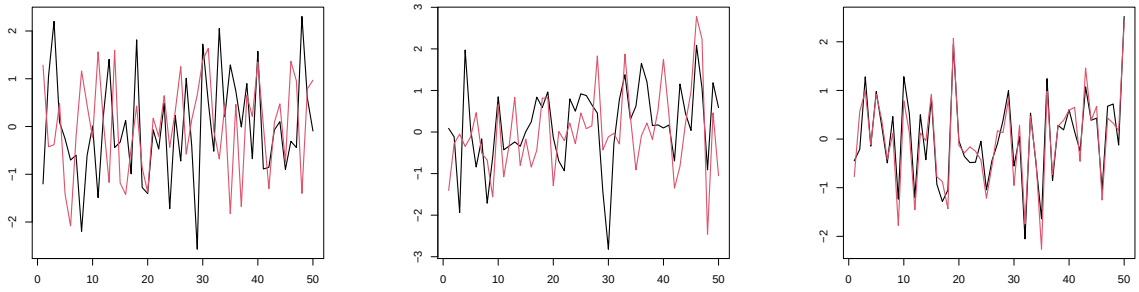
$$H_j = \max_{0 \le t \le 1} X_j(t), \qquad L_j = \min_{0 \le t \le 1} X_j(t), \qquad S_j = X_j(1),$$

for the high, low and final log price, respectively, over a fixed time interval $[0, 1]$. We want to estimate the drift $\boldsymbol{\mu}$ and the volatility matrix Σ merely from observing these three prices over a period of time.

We impose a normal-inverse-Wishart prior $(\boldsymbol{\mu}, \Sigma) \sim NIW(\boldsymbol{\mu}_0, \lambda, \Phi, \nu)$. This distribution can be sampled from in two steps: (1) sample Σ from an inverse Wishart distribution $\Sigma \mid \Phi, \nu \sim W^{-1}(\Phi, \nu)$; (2) sample $\boldsymbol{\mu}$ from a multivariate normal distribution $\boldsymbol{\mu} \mid \boldsymbol{\mu}_0, \lambda, \Sigma \sim N(\boldsymbol{\mu}_0, \frac{1}{\lambda}\Sigma)$. Since Σ is a semi-positive definite matrix, we model the parameters through its Cholesky root $\Sigma^{1/2}$. Without loss of generality, we only consider the case $W(0) = (0, \ldots, 0)'$, since the closing prices on the previous day or the opening prices of today are usually known.

2.5.1 Synthetic Data

To compare various likelihood-free estimators, we first generate synthetic data for 1 000 trading days. For each day (of length $t = 1$), we simulate the Brownian motion using 500 time steps to obtain the high, low and closing price data for each particular window. We first restrict our attention to the case of just two assets, which leaves us with 5 parameters to estimate: μ_1, μ_2 and the upper triangular root of Σ, denoted with $L = \begin{bmatrix} l_{11} & 0 \\ l_{12} & l_{22} \end{bmatrix}$. We first illustrate how the covariance parameter σ_{12} impacts the co-movement of asset prices. Holding $\mu_1 = \mu_2 = 0$ and $\sigma_{11} = \sigma_{22} = 1$ fixed, we plot time series realizations of the closing prices for three particular choices of σ_{12} in Figure 2.5. The patterns are as expected where the prices tend to co-fluctuate when σ_{12} is closer to one. The success of our method depends on how well the discriminator can tell apart these trajectories.



(a) $\sigma_{12} = 0.05$ (b) $\sigma_{12} = 0.5$ (c) $\sigma_{12} = 0.95$

Figure 2.5: Closing prices time series for three choices of σ_{12} ($\mu_1 = \mu_2 = 0, \sigma_{11} = \sigma_{22} = 1$).

For our simulation, we set $\mu = (0, 0)'$ and $\sigma_{11} = \sigma_{22} = 1, \sigma_{12} = 0.5$. We choose relatively non-informative prior hyper-parameters $\boldsymbol{\mu}_0 = (0, 0)', \lambda = 1, \Phi = I_d$ and $\nu = d$. Posterior distributions reconstructed with different ABC methods are given in Figure 2.6 and the averaged performance over 10 repetitions is summarized in Table 2.3. We explore two discriminators built on the prices and the quadratic/interaction terms of the prices: (1) a lasso classifier with penalty term λ selected from a 5-fold cross-validation (LRD); (2) a random forest classifier (RF). We find out that for the Lotka-Volterra model, the linear

classifier performs better than the nonparametric random forest, as reflected in Figure 2.6. We observe a smaller estimation bias and the computation of LRD is much shorted than RF. Thus, we only include LRD in the repetitions. It is clear that our exponential kernel methods place more mass around the true location of the parameters, and that the shape of kNN-exp is slightly less regular than LRD-exp. Although LRD-exp induces a larger bias in estimating the drift (μ_1, μ_2), it does a better job at capturing the correct location of the volatilities.

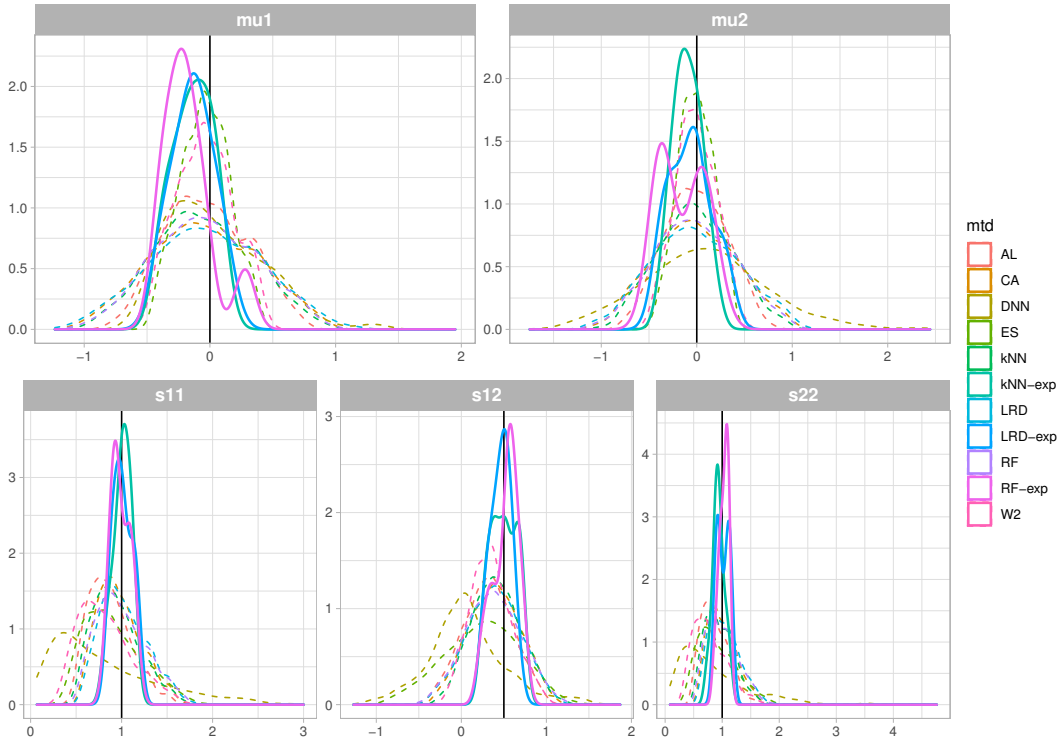


Figure 2.6: Posterior densities on simulated volatility data ($d = 2$)

2.5.2 Real Data Analysis

Following the example in Rogers and Zhou (2008), we examine a small data set of stock prices focusing on two stocks: Boeing (BA) and Proctor & Gamble (PG). The prices were obtained from NYSE (Yahoo Finance), starting from 3rd January 2011 and consisting of 1 000 trading days. Since the off-market trades follow a different mechanism than the market trade, we

		LRD	LRD-exp	kNN	kNN-exp	ES	CA	W2	AL	SS	DNN
$\mu_1 = 0$	MSE($\times 10^{-4}$)	2.004	2.113	0.638	**0.159**	1.484	1.497	0.170	0.628	1.216	1.838
	95% CI width	1.760	**0.331**	1.480	0.442	0.740	1.660	0.831	1.240	1.734	3.079
$\mu_2 = 0$	MSE($\times 10^{-4}$)	0.734	7.439	0.363	0.259	0.050	0.142	0.277	0.089	**0.021**	7.029
	95% CI width	1.628	**0.431**	1.435	0.538	0.737	1.564	0.813	1.247	1.569	2.527
$\sigma_{11} = 1$	MSE($\times 10^{-3}$)	0.311	0.111	0.167	0.460	0.834	**0.002**	2.634	1.669	1.404	3.502
	95% CI width	0.963	0.371	0.954	**0.329**	1.231	0.896	1.140	0.910	0.763	6.233
$\sigma_{12} = 0.5$	MSE($\times 10^{-3}$)	1.035	**0.103**	0.427	0.107	6.657	0.861	1.528	3.337	0.109	18.427
	95% CI width	1.193	0.355	1.092	**0.324**	1.700	1.122	0.988	1.114	0.628	5.530
$\sigma_{22} = 1$	MSE($\times 10^{-3}$)	0.416	0.304	0.124	0.018	0.776	**0.007**	4.244	1.679	3.687	1.882
	95% CI width	0.959	**0.252**	0.974	0.314	1.274	0.920	1.157	0.922	0.783	6.693

Table 2.3: Performance on the stock volatility estimation example, averaged over 10 repetitions, with top 1% selected. All 95% CIs have full coverage of the true parameters. The bold fonts mark the best model in each row.

only model price changes from the opening price to the closing price each day where the log price differences $X(t)$ are all computed based on the opening prices of that day.

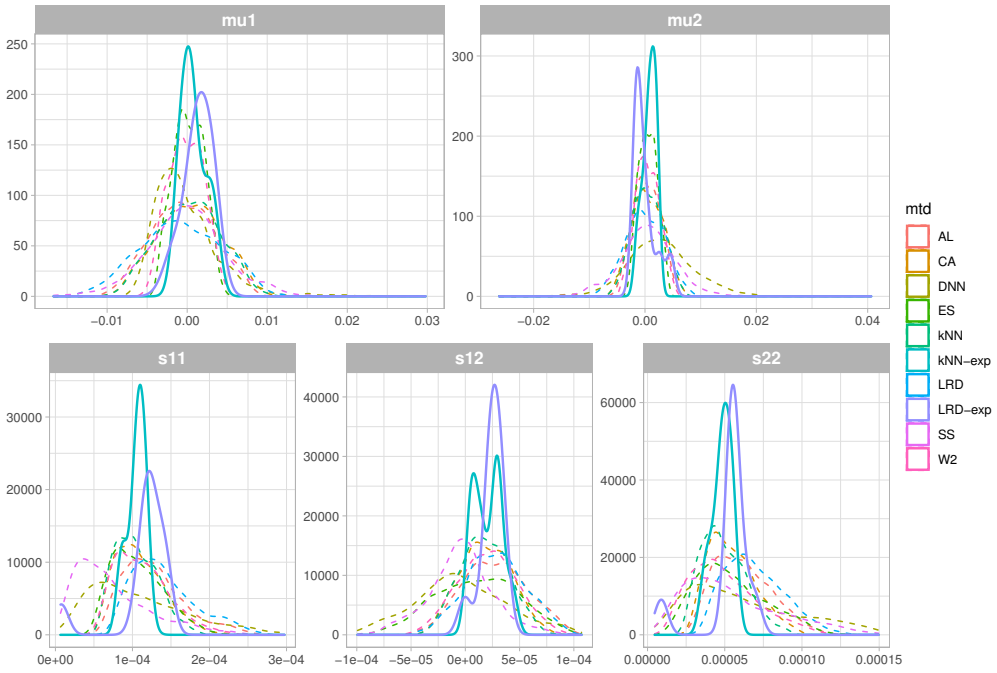


Figure 2.7: Posterior densities estimated from log-prices of BA and PG

We observe that the fluctuations in the prices are much smaller than in our simulated time series and we thereby choose the hyperparameters based on the mean and covariance of the closing prices. Figure 2.7 gives the ABC posterior distributions estimated from different

methods and the corresponding summaries of the distributions are provided in Table 2.4. We again observe that our methods with the exponential kernels return much narrower posterior distributions. The estimates on the volatilities are quite close except for the ABC with summary statistics (SS). The drifts of both BA and PG are not significantly different from zero.

		LRD	LRD-exp	kNN	kNN-exp	CA	ES	W2	AL	DNN	SS
μ_1 ($\times 10^{-3}$)	$\bar{\mu}_1$	-0.774	1.354	0.250	0.779	0.342	0.173	-0.126	-0.500	-0.091	-0.071
	l	-10.380	-1.981	-6.816	-1.145	-7.099	-3.201	-3.987	-7.603	-5.415	-9.532
	u	8.202	3.514	7.229	3.385	7.395	3.443	3.736	6.810	9.335	10.121
μ_2 ($\times 10^{-3}$)	$\bar{\mu}_2$	0.009	-0.039	0.491	0.822	0.445	0.545	0.453	0.316	2.793	0.450
	l	-6.591	-1.718	-4.329	-1.406	-4.604	-2.364	-3.491	-4.103	-8.913	-9.831
	u	6.531	4.563	5.503	2.166	5.538	3.616	4.450	4.690	14.701	10.171
$\sigma_{11}(\times 10^{-4})$	$\bar{\sigma}_{11}$	1.406	1.142	1.049	1.054	1.101	1.070	1.130	1.273	1.224	0.738
	l	0.824	1.142	0.633	0.844	0.634	0.577	0.626	0.689	0.243	0.155
	u	2.362	1.142	1.640	1.202	1.819	1.818	1.938	2.259	3.616	1.976
$\sigma_{12}(\times 10^{-5})$	$\bar{\sigma}_{12}$	2.811	2.446	1.767	2.084	1.906	1.074	2.052	2.759	-0.530	-0.376
	l	-2.379	2.446	-2.359	2.084	-2.390	-6.152	-3.137	-2.656	-25.243	-6.660
	u	8.226	2.446	6.037	2.084	6.442	8.008	7.253	8.568	23.343	6.065
$\sigma_{22}(\times 10^{-4})$	$\bar{\sigma}_{22}$	6.922	5.263	4.924	4.820	5.470	5.419	4.947	6.179	18.485	5.521
	l	4.069	5.263	2.766	4.820	3.149	2.005	1.734	3.322	1.571	1.326
	u	11.049	5.263	8.020	4.820	8.826	10.258	9.844	10.223	93.218	12.852

Table 2.4: Posterior estimates on analysis of BA and PG. For each parameter, we report three summary statistics, the posterior means, the lower limit of the 95% CI intervals (l) and the upper limit of the 95% CI intervals (u).

2.6 Discussion

This paper develops an ABC variant using a classification-based KL estimator as a discrepancy measure. By deploying a flexible classifier, the empirical KL divergence can be estimated with a vanishing error. In addition, inspired by the connection between the KL divergence and the log-likelihood ratio, we propose a scaled exponential kernel to aggregate ABC samples. This smoothing variant avoids the need for choosing the ad hoc threshold ϵ_n and fully utilizes information returned from all ABC samples. Under mild conditions, we show that the posterior concentration rate of the accept-reject ABC depends on the es-

timation error δ_n and the accept-reject threshold ϵ_n, while the rate of the smooth version depends on the estimation error δ_n and the contraction rate of the actual posterior distribution. Our methodology can also be related to many other likelihood-free inference methods, including ABC with Classification Accuracy (Gutmann et al., 2018), Wasserstein distance ABC (Bernton et al., 2019), and Generalized Posteriors (Schmon et al., 2020). Our methods coincide with the c-posterior (Miller and Dunson, 2018), which is robust to perturbations. In particular, the accept-reject ABC can be shown to be robust under model misspecification (see Section 2.3.4). In addition, the exponential kernel can be motivated as an instantiation of General Bayesian Inference (GBI) (Bissiri et al., 2016) with the KL estimator as the loss function. See Thomas and Corander (2019) and Thomas et al. (2020b) for examples of conducting robust inference using probabilistic classifiers under the generalized Bayes update setup. Along with our theoretical investigations, we demonstrate competitive performance of our methods on benchmark examples. Our theoretical analysis provides theoretical justifications for the method of Gutmann et al. (2018).

2.7 Appendix

2.7.1 Convergence Rate of Estimation Errors with Neural Network Sieves

To develop a precise definition of the low underlying dimension d^* described in Remark 6, we borrow the smoothness notion of Bauer and Kohler (2019).

Definition 1 ((p, C)-Smoothness)**.** *Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $0 < s \leq 1$. A function $m : \mathbb{R}^d \to \mathbb{R}$ is called* (p, C)-smooth *if for every $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ with $\sum_{j=1}^d \alpha_j = q$, the partial derivative $\frac{\partial^q m}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$ exists and satisfies*

$$\left| \frac{\partial^q m}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}(x) - \frac{\partial^q m}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}(z) \right| \leq C \|x - z\|^s$$

for every $x, z \in \mathbb{R}^d$ *where* $\|\cdot\|$ *denotes the Euclidean norm.*

With this, the nested composition structure is defined as follows.

Definition 2 (Generalized Hierarchical Interaction Model)**.** *Let* $d \in \mathbb{N}$, $d^* \in \{1, \dots, d\}$, *and* $m : \mathbb{R}^d \to \mathbb{R}$. *We say that the function* m *satisfies a* generalized hierarchical interaction model of order d^* and level 0, *if there exist* $a_1 \in \mathbb{R}^d, \dots, a_{d^*} \in \mathbb{R}^d$, *and* $f : \mathbb{R}^{d^*} \to \mathbb{R}$ *such that*

$$m(x) = f(a_1'x, \dots, a_{d^*}'x)$$

for every $x \in \mathbb{R}^d$. *We say that* m *satisfies a* generalized hierarchical interaction model of order d^* and level $l+1$ with K components *if there exist* $g_k : \mathbb{R}^{d^*} \to \mathbb{R}$ *and* $f_{1,k}, \dots, f_{d^*,k} : \mathbb{R}^d \to \mathbb{R}$ $(k = 1, \dots, K)$ *such that* $f_{1,k}, \dots, f_{d^*,k}$ $(k = 1, \dots, K)$ *satisfy a generalized hierarchical model of order* d^* *and level* l *and*

$$m(x) = \sum_{k=1}^{K} g_k(f_{1,k}(x), \dots, f_{d^*,k}(x))$$

for every $x \in \mathbb{R}^d$. *We say that the generalized hierarchical interaction model is* (p, C)-smooth *if all functions occurring in its definition are* (p, C)*-smooth.*

For example, a conditional binary choice model satisfies a generalized hierarchical interaction model of order $d^* \leq 3$ and level 0, irrespectively of the dimension of the covariates.

Example 1 (Binary Choice Model)**.** *Let* $y_i = \mathbb{I}\{x_i'\alpha + \varepsilon_i > 0\}$, $\varepsilon_i \sim P_\varepsilon$, *be the true DGP and* $y_i^\beta = \mathbb{I}\{x_i'\beta + \tilde{\varepsilon}_i > 0\}$, $\tilde{\varepsilon}_i \sim \tilde{P}_\varepsilon$, *be the generative model. Then,*

$$\log \frac{p_0(y, x)}{p_\theta(y, x)} = y \log \frac{1 - P_\varepsilon(-x'\alpha)}{1 - \tilde{P}_\varepsilon(-x'\beta)} + (1 - y) \log \frac{P_\varepsilon(-x'\alpha)}{\tilde{P}_\varepsilon(-x'\beta)}.$$

Therefore, we can write this as $f(a_1'z, a_2'z, a_3'z)$ *where* $z = (y, x')'$, $a_1 = (1, 0, \dots, 0)'$, $a_2 = (0, -\alpha')'$, $a_3 = (0, -\beta')'$, *and* $f(y, x_1, x_2) = y[\log(1 - P_\varepsilon(x_1)) - \log(1 - \tilde{P}_\varepsilon(x_2))] + (1 -$

$y)[\log P_\varepsilon(x_1) - \log \tilde{P}_\varepsilon(x_2)]$. *If* P_ε *and* $\tilde{P}_\varepsilon$ *are logistic distributions, then* f *is* (p, C)*-smooth for* $(p, C) = (8, 1)$ *or for* $(p, C) = (11, 10)$*, for example. Therefore, it satisfies the generalized hierarchical interaction model of order* $d^* = 3$ *and level* $l = 0$.

Example 2 (Diffusion Process)**.** *Let* $X_{i,t/d}$, $t = 1, \ldots, d$ *be discretely sampled observations of a Brownian motion* $\mathrm{d}X_{it} = \mu \mathrm{d}t + \sigma \mathrm{d}W_{it}$ *with* $X_{i0} = 0$*, where* W_{it} *is a standard Brownian motion independent across* i*. Let the generative model be* $X^\theta_{i,t/d}$, $t = 1, \ldots, d$ *from* $X^\theta_{it} = m\mathrm{d}t + s\mathrm{d}\tilde{W}_{it}$ *and* $\theta = (m, s)$*. Then, the log likelihood ratio is*

$$\log \frac{p_0(x_{1/d}, \ldots, x_1)}{p_\theta(x_{1/d}, \ldots, x_1)} = d \log \frac{s}{\sigma} - \frac{d}{2}\left(\frac{1}{\sigma^2} - \frac{1}{s^2}\right) \sum_{j=1}^{d} (x_{j/d} - x_{(j-1)/d})^2 + \left(\frac{\mu}{\sigma^2} - \frac{m}{s^2}\right) x_1 - \frac{1}{2}\left(\frac{\mu^2}{\sigma^2} - \frac{m^2}{s^2}\right).$$

Letting $z = ((x_{1/d} - x_0)^2, \ldots, (x_1 - x_{(d-1)/d})^2, x_1)'$*, we can write this as* $f(a'z)$ *where* $a = (-\frac{d}{2}(\frac{1}{\sigma^2} - \frac{1}{s^2}), \ldots, -\frac{d}{2}(\frac{1}{\sigma^2} - \frac{1}{s^2}), \frac{\mu}{\sigma^2} - \frac{m}{s^2})'$ *and* $f(y) = d \log \frac{s}{\sigma} + y - \frac{1}{2}\big(\frac{\mu^2}{\sigma^2} - \frac{m^2}{s^2}\big)$*. Then,* f *is* (p, C)*-smooth with* $(p, C) = (\infty, 1)$*, and the log likelihood ratio satisfies the hierarchical model with* $d^* = 1$ *and level* $l = 0$.

Next, we define the configuration of the neural network appropriate for estimating a generalized hierarchical interaction model.

Definition 3 (Hierarchical Neural Network)**.** *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be a* q*-admissible activation function. For* $M^* \in \mathbb{N}$, $d \in \mathbb{N}$, $d^* \in \{1, \ldots, d\}$*, and* $\alpha > 0$*, let* $\mathcal{F}_{M^*,d^*,d,\alpha}$ *be the class of functions* $f : \mathbb{R}^d \to \mathbb{R}$ *such that*

$$f(x) = \sum_{i=1}^{M^*} \mu_i \sigma\left(\sum_{j=1}^{4d^*} \lambda_{i,j} \sigma\left(\sum_{v=1}^{d} \theta_{i,j,v} x_v + \theta_{i,j,0}\right) + \lambda_{i,0}\right) + \mu_0$$

for some $\mu_i, \lambda_{i,j}, \theta_{i,j,v} \in \mathbb{R}$*, where* $|\mu_i| \le \alpha$*,* $|\lambda_{i,j}| \le \alpha$*, and* $|\theta_{i,j,v}| \le \alpha$*. For* $l = 0$*, define the set of neural networks with two hidden layers by* $\mathcal{H}^{(0)}_{M^*,d^*,d,\alpha} = \mathcal{F}_{M^*,d^*,d,\alpha}$*; for* $l > 0$*, define*

the set of neural networks with $2l+2$ *hidden layers by*

$$\mathcal{H}^{(l)}_{M^*,d^*,d,\alpha} = \left\{ h : \mathbb{R}^d \to \mathbb{R} : h(x) = \sum_{k=1}^{K} g_k(f_{1,k}(x), \ldots, f_{d^*,k}(x)),\ g_k \in \mathcal{F}_{M^*,d^*,d^*,\alpha},\ f_{j,k} \in \mathcal{H}^{(l-1)} \right\}.$$

With these definitions, the convergence rate of the neural network sieve is characterized as follows. If the log likelihood ratio satisfies the generalized hierarchical model of order d^* and the corresponding neural network sieve is used, the convergence rate of the discriminator depends only on d^*, not on d. The precise assumption is formulated as follows.

Assumption 6 (Neural Network Discriminator)**.** *Let* P_0 *and* P_θ *have subexponential tails and finite first moments.*[4] *Let* $\log(p_0/p_\theta)$ *satisfy a* (p, C)*-smooth generalized hierarchical interaction model of order* d^* *and finite level* l *with* K *components for* $p = q + s$*,* $q \in \mathbb{N}_0$*, and* $s \in (0, 1]$*. Let* $\mathcal{H}^{(l)}_{M^*,d^*,d,\alpha}$ *be the class of neural networks with the Lipschitz activation function with Lipschitz constant* 1 *for*

$$M_* = \left\lceil \binom{d^*+q}{d^*} (q+1) \left(\left\lceil \frac{(\log \delta_n)^{2(2q+3)}}{\delta_n} \right\rceil^{\frac{1}{p}} + 1 \right)^{d^*} \right\rceil,$$
$$\alpha = \left\lceil \frac{(\log \delta_n)^{2(2q+3)}}{\delta_n} \right\rceil^{\frac{d^*+p(2q+3)+1}{p}} \frac{\log n}{\delta_n^2},$$

and $\delta_n = [(\log n)^{\frac{p+2d^*(2q+3)}{p}}/n]^{\frac{p}{2p+d^*}}$*. Denote by* $\mathcal{D}_n = \{\Lambda(f) : f \in \mathcal{H}^{(l)}_{M^*,d^*,d,\alpha}\}$ *the sieve of neural network discriminators for the standard logistic cdf* Λ*.*

Assumption 6 gives a sufficient condition for the entropy condition in Assumption 1. With this, we obtain the "classification counterpart" of Bauer and Kohler (2019, Theorem 1) as below.

4. We say that P on $\mathbb{R}^d$ has *subexponential tails* if $\log P(\|X\|_\infty > a) \lesssim -a$ for large a.

Theorem 11 (Kaji et al., 2020, Proposition S.3)**.** *Suppose Assumption 6 holds. If n/m converges and an estimator $\hat{D}^{\theta}_{n,m}$ exists that satisfies $\mathbb{M}_{n,m}(\hat{D}^{\theta}_{n,m}) \geq \mathbb{M}^{\theta}_{n,m}(D_\theta) - O_P(\delta_n^2)$ for the δ_n in Assumption 6, then $d_\theta(\hat{D}^{\theta}_{n,m}, D_\theta) = O^*_P(\delta_n)$.*

This theorem, combined with the explicit expression of δ_n in Assumption 6, tells that if $d^* < 2p$, we have $\delta_n = o_P(n^{-1/4})$, which is often the desired rate for the nonparametric estimator of a nuisance parameter. For the binary choice model in Example 1, the discriminator converges much faster than $n^{-1/4}$.

Example 1 (continuing from p. 44)**.** *The binary choice model with logistic errors satisfies a (p, C)-smooth hierarchical model with $(p, C) = (8, 1)$ and $d^* = 3$. We can substitute these numbers (and $q = p-1$) into δ_n in Assumption 6 and obtain $\delta_n = \big(\frac{(\log n)^{13.75}}{n}\big)^{8/19} \lesssim n^{-2/5}$.*

Example 2 (continuing from p. 45)**.** *The discretely sampled Brownian motion model satisfies a (p, C)-smooth hierarchical model with $d^* = 1$, $C = 1$, and arbitrarily large p. Therefore, δ_n can be arbitrarily close to $n^{-1/2}$, however large the sampling frequency is.*

When there is no low-dimensional structure, Definition 2 simply reduces to the smoothness of m and d^* is equal to d. Therefore, the convergence rate in Theorem 11 reduces to the traditionally proven rate that deteriorates quickly with d.

We note that the hierarchical structure is not the only way to proving the superior adaptivity of a neural network discriminator. Similar results can possibly be deduced with other smoothness assumptions and network configurations, such as Schmidt-Hieber (2020) and Yarotsky (2017).

2.7.2 Frequentist's Analysis on the Exponential Kernel

We thus study the concentration in terms of a KL neighborhood around $Q^{(n)}_{\theta^*}$ defined as

$$B(\epsilon, Q^{(n)}_{\theta^*}; P^{(n)}_0) = \{Q^{(n)}_{\theta^*} \in \mathcal{Q}^{(n)} : \tilde{K}(\theta^*, \theta) \leq n\epsilon^2, \tilde{V}(\theta^*, \theta) \leq n\epsilon^2\}, \tag{2.23}$$

where $\tilde{K}(\theta^*,\theta) \equiv P_0^{(n)} \log \frac{q_{\theta^*}^{(n)}}{q_\theta^{(n)}}$ and $\tilde{V}(\theta^*,\theta) \equiv P_0^{(n)} \left| \log \frac{q_{\theta^*}^{(n)}}{q_\theta^{(n)}} - \tilde{K}(\theta^*,\theta) \right|^2$.

The following corollary is directly adopted from Theorem 5.1 of Kaji and Ročková (2022).

Corollary 12. *Denote with $\widetilde{Q}_\theta^{(n)}$ a measure defined through $\mathrm{d}\widetilde{Q}_\theta^{(n)} = (p_0^{(n)}/q_{\theta^*}^{(n)})\mathrm{d}P_\theta^{(n)}$ and let $d(\cdot,\cdot)$ be a semi-metric on $\mathcal{P}^{(n)}$. Suppose that there exists a sequence $\epsilon_n > 0$ satisfying $\epsilon_n \to 0$ and $n\epsilon_n^2 \to \infty$ such that for every $\epsilon > \epsilon_n$ there exists a test ϕ_n (depending on ϵ) such that for every $J \in \mathbb{N}_0$*

$$P_0^{(n)}\phi_n \lesssim e^{-n\epsilon^2/4} \quad \text{and} \quad \sup_{Q_\theta^{(n)}: d(Q_\theta^{(n)}, P_{\theta^*}^{(n)}) > J\epsilon} \widetilde{Q}_\theta^{(n)}(1-\phi_n) \le e^{-nJ^2\epsilon^2/4}. \tag{2.24}$$

Let $B(\epsilon, Q_{\theta^*}^{(n)}; P_0^{(n)})$ be as in (2.23) and let $\tilde{\Pi}_n(\theta)$ be a prior distribution with a density $\tilde{\pi}(\theta) \propto C_\theta \pi(\theta)$ with C_θ as in (2.20). Assume that there exists a constant $L > 0$ such that, for all n and $j \in \mathbb{N}$,

$$\frac{\tilde{\Pi}_n\left(\theta \in \Theta : j\epsilon_n < d(Q_\theta^{(n)}, P_{\theta^*}^{(n)}) \le (j+1)\epsilon_n\right)}{\tilde{\Pi}_n\left(B(\epsilon_n, Q_{\theta^*}^{(n)}; P_0^{(n)})\right)} \le e^{n\epsilon_n^2 j^2/8}. \tag{2.25}$$

Then for every sufficiently large constant M, as $n \to \infty$,

$$P_0^{(n)}\Pi^{EK}\left(Q_\theta^{(n)} : d(Q_\theta^{(n)}, P_{\theta^*}^{(n)}) \ge M\epsilon_n \mid X^{(n)}\right) \to 0. \tag{2.26}$$

Next, we want to show the shape of the posterior is actually asymptotically gaussian around θ^*. The following corollary follows from Theorem 2.1 of Kleijn and van der Vaart (2012) and Lemma 8.1 of Kaji and Ročková (2022).

Corollary 13. *(Bernstein von-Mises) Assume that the posterior (2.20) concentrates around*

θ^* at the rate ϵ_n^* and that for every compact $K \in \mathbb{R}^d$

$$\sup_{h\in K}\left|\log\frac{q^{(n)}_{\theta^*+\epsilon_n^* h}(\boldsymbol{X})}{q^{(n)}_{\theta^*}(\boldsymbol{X})} - h'\tilde{V}_{\theta^*}\tilde{\delta}_{n,\theta^*} - \frac{1}{2}h'\tilde{V}_{\theta^*}h\right| \to 0 \tag{2.27}$$

for some random vector $\tilde{\delta}_{n,\theta^*}$ and a non-singular matrix $\tilde{V}_{\theta^*}$. Then the approximated posterior $\Pi^{EK}(\cdot)$ converges to a sequence of normal distributions in total variation at the rate ϵ_n^*, i.e.

$$\sup_B\left|\Pi^{EK}\left(\epsilon_n^{*-1}(\theta-\theta^*\right) \in B \mid \boldsymbol{X}) - N_{\tilde{\delta}_{n,\theta^*},\tilde{V}^{-1}_{\theta^*}(B)}\right| \to 0 \quad \textit{in } P_0^{(n)} - \textit{probability.} \tag{2.28}$$

2.7.3 Proofs

2.7.3.1 Proof of Theorem 5

Denote $K_n = \mathbb{P}_n \log\frac{p_0}{p_\theta}$. Using Chebyshev's inequality and Assumption 2 as

$$\begin{aligned}
P_0^{(n)}[|K_n - K(p_0,p_\theta)| > u] &= P_0^{(n)}\Big(\left|(\mathbb{P}_n - P_0)\log\frac{p_0}{p_\theta}\right| > u\Big) \le \frac{1}{u^2}P_0^{(n)}\Big[\left|(\mathbb{P}_n - P_0)\log\frac{p_0}{p_\theta}\right|^2\Big] \\
&= \frac{1}{u^2}P_0^{(n)}\Big[\left|\mathbb{P}_n\big(\log\frac{p_0}{p_\theta} - P_0\log\frac{p_0}{p_\theta}\big)\right|^2\Big] = \frac{1}{nu^2}P_0\Big[\left|\log\frac{p_0}{p_\theta} - P_0\log\frac{p_0}{p_\theta}\right|^2\Big] \\
&\le \frac{16(2+\Lambda)h(p_0,p_\theta)}{nu^2},
\end{aligned}$$

where the last inequality follows from Lemma 2.1 (iii) of Kaji and Ročková (2022). Next, note

$$\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K_n = -\mathbb{P}_n\Big(\log\frac{1-\hat{D}^\theta_{n,m}}{1-D_\theta} - \log\frac{\hat{D}^\theta_{n,m}}{D_\theta}\Big).$$

Recall the outer expectation P^* in Lemma 2 and the set of classifiers $D^{\theta}_{n,\delta}$ defined in Assumption 1, we can bound

$$\begin{aligned}
P\Bigg(\bigg|\mathbb{P}_n\Big(\log\frac{1-\hat{D}^{\theta}_{n,m}}{1-D_\theta}-\log\frac{\hat{D}^{\theta}_{n,m}}{D_\theta}\Big)\bigg|>u, d_\theta(\hat{D}^{\theta}_{n,m},D_\theta)\le C_n\delta_n\Bigg)\\
\le P^*\Big(\sup_{D\in\mathcal{D}^{\theta}_{C_n\delta_n}}\bigg|\mathbb{P}_n\Big(\log\frac{1-D}{1-D_\theta}-\log\frac{D}{D_\theta}\Big)\bigg|>u\Big)\\
\le \frac{1}{u}\mathbb{E}^*\sup_{D\in\mathcal{D}^{\theta}_{C_n\delta_n}}\bigg|\mathbb{P}_n\Big(\log\frac{1-D}{1-D_\theta}-\log\frac{D}{D_\theta}\Big)\bigg|
\end{aligned}$$

by Markov's inequality. The proof of Theorem 4.1 of Kaji and Ročková (2022) shows that the expectation is $O(C_n\delta_n)$. Using the triangle inequality and the Bonferroni inequality, since $h(p_0,p_\theta)\le 2$, we can then write

$$\begin{aligned}
&P\Bigg(\Big|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K(p_0,p_\theta)\Big|>2u, d_\theta(\hat{D}^{\theta}_{n,m},D^{\theta})\le C_n\delta_n\Bigg)\\
&\le P\Bigg(\Big|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K_n\Big|+|K_n-K(p_0,p_\theta)|>2u, d_\theta(\hat{D}^{\theta}_{n,m},D_\theta)\le C_n\delta_n\Bigg)\\
&\le P\Bigg(\Big|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K_n\Big|>u, d_\theta(\hat{D}^{\theta}_{n,m},D_\theta)\le C_n\delta_n\Bigg)+P_0^{(n)}[|K_n-K(p_0,p_\theta)|>u]\\
&\le O\Big(\frac{C_n\delta_u}{u}\Big)+\frac{32(2+\Lambda)}{nu^2}.
\end{aligned}$$

2.7.3.2 Proof of Theorem 7

Throughout, we continue to assume that $\tilde{\boldsymbol{X}}^{\theta}=g_\theta(\tilde{\boldsymbol{X}})$ and we denote with $P=P_0^{(n)}\otimes\tilde{P}^{(m)}$ the joint measure for $(\boldsymbol{X},\tilde{\boldsymbol{X}})$. Below, we will be using the notation $\Pi(\cdot)$ to denote the generic probability, i.e. for $(\theta,\tilde{\boldsymbol{X}})$ or for the conditional probability θ given $\tilde{\boldsymbol{X}}$. Later, we will define a high-probability event $\Omega_n(C,\varepsilon_n)$ such that $P_0^{(n)}[\Omega_n(C,\varepsilon_n)^c]=o(1)$ for some $C\in(0,1)$. Given $\delta_n>0$ from our assumptions, we can write for every $\lambda_n>0$ and $\epsilon_n>0$ and for every

arbitrarily slowly increasing sequence $C_n > 0$

$$P_0^{(n)}\Pi\left(K(p_0,p_\theta) > \lambda_n \middle| \hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \le \epsilon_n\right) \le \Pi_1 + o(1) + \\ P_0^{(n)}\Pi\left(K(p_0,p_\theta) > \lambda_n \middle| \hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \le \epsilon_n, d(\hat{D}^\theta_{n,m}, D_\theta) \le C_n\delta_n\right) \mathbb{I}[\Omega_n(C,\varepsilon_n)], \quad (2.29)$$

where, using Lemma 2 and the fact that the rate δ_n is uniform (see Remark 4),

$$\Pi_1 \equiv P_0^{(n)}\Pi(d(\hat{D}^\theta_{n,m}, D_\theta) > C_n\delta_n) \le \sup_{\theta\in\Theta} P(d(\hat{D}^\theta_{n,m}, D_\theta) > C_n\delta_n) = o(1).$$

Consider the joint event, for some $\delta' > 0$,

$$A_{\epsilon_n}(\delta') = \{(\tilde{\boldsymbol{X}},\theta) : \hat{K}[\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})] \le \epsilon_n\} \cap \{K(p_0,p_\theta) > \delta'\}.$$

For every $(\tilde{\boldsymbol{X}},\theta) \in A_{\epsilon_n}(\delta')$ we have

$$K(p_0,p_\theta) \le \hat{K}(\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})) + \left|\hat{K}(\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})) - K(p_0,p_\theta)\right| \le \epsilon_n + \left|\hat{K}(\boldsymbol{X}, g_\theta(\tilde{\boldsymbol{X}})) - K(p_0,p_\theta)\right|.$$

Hence $(\tilde{\boldsymbol{X}},\theta) \in A_{\epsilon_n}(\delta')$ implies that

$$\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > \delta' - \epsilon_n,$$

and choosing $\delta' \ge \epsilon_n + t_\epsilon$ leads to

$$\Pi[A_{\epsilon_n}(\delta')] \le \int_\Theta \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > t_\epsilon\right] d\Pi(\theta).$$

Using (2.29), we now focus on the conditional probability, given $d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n$,

$$\Pi\left(K(p_0,p_\theta) > \epsilon_n + t_\epsilon \middle| \hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n, d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right)$$
$$\leq \frac{\int_\Theta \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > t_\epsilon \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)}{\int_\Theta \tilde{P}^{(m)}\left[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)}. \quad (2.30)$$

We now find a lower bound for the denominator. Recall the KL neighborhood $B_2(p_0,\epsilon_n)$ defined in (2.13). Since

$$\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \leq K(p_0,p_\theta) + \left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| \leq \epsilon_n/2 + K(p_0,p_\theta),$$

provided that $\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| \leq \epsilon_n/2$. The denominator can be then bounded by

$$\int_\Theta \tilde{P}^{(m)}[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n] d\Pi(\theta)$$
$$\geq \int_{B_2(p_0,\epsilon_n/2)} \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| \leq \epsilon_n/2 \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)$$
$$\geq \Pi[B_2(p_0,\epsilon_n/2)] - \int_{B_2(p_0,\epsilon_n/2)} \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > \epsilon_n/2 \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta).$$

Denoting

$$Z(\boldsymbol{X}) \equiv \int_{B_2(p_0,\epsilon_n/2)} \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > \epsilon_n/2 \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)$$

we can write, for every $C > 0$, using Fubini's theorem and Markov's inequality

$$P_0^{(n)}\left(Z(\boldsymbol{X}) > C\right) \leq \frac{1}{C}\int_{B_2(p_0,\epsilon_n/2)} P\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > \epsilon_n/2 \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)$$
$$= \frac{\Pi(B_2(p_0,\epsilon_n/2))}{C} \frac{\sup_{\theta\in\Theta} P\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta)\right| > \epsilon_n/2, d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right]}{\sup_{\theta\in\Theta} P\left[d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right]}$$
$$= \frac{\Pi(B_2(p_0,\epsilon_n/2))}{C(1+o(1))} \sup_{\theta\in\Theta} \rho_{n,\theta}(\varepsilon_n/2; C_n; \delta_n),$$

where we have used Theorem 5 and the fact that $P(d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n) = 1 + o(1)$ for every $\theta \in \Theta$ from Lemma 2. We now define an event, for some $0 < C < 1$ and $\epsilon_n > 0$,

$$\Omega_n(C, \epsilon_n) = \{\boldsymbol{X} : Z(\boldsymbol{X}) \leq C \times \Pi(B_2(p_0, \epsilon_n/2))\}.$$

Using Theorem 5, we have

$$\rho_{n,\theta}(\epsilon_n/2; C_n; \delta_n) = O\Big(\frac{C_n\delta_n}{\epsilon_n} + \frac{1}{n\epsilon_n^2}\Big) \quad \text{for every} \quad \theta \in \Theta.$$

Choosing $\epsilon_n > 0$ such that $\epsilon_n = o(1)$ and $n\epsilon_n^2 \to \infty$ and $C_n\delta_n = o(\epsilon_n)$ we have $P_0^{(n)}[\Omega_n(C, \epsilon_n)^c] = o(1)$ for every $C \in (0, 1)$. On the event $\Omega_n(C, \epsilon_n)$, for some $0 < C < 1$, we can lower-bound the denominator with

$$\int_\Theta \tilde{P}^{(m)}[\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n] d\Pi(\theta) > (1 - C) \times \Pi(B_2(p_0, \epsilon_n/2).$$

Using this bound and applying Fubini's theorem, we can further write

$$\begin{aligned} P_0^{(n)}\Pi\left(K(p_0, p_\theta) > \epsilon_n + t_\epsilon \mid \hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) \leq \epsilon_n, d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right) \mathbb{I}[\Omega_n(C, \varepsilon_n)] \\ \leq \frac{\int_\Theta P\left[\left|\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^\theta) - K(p_0, p_\theta)\right| > t_\epsilon \mid d(\hat{D}^{\theta}_{n,m}, D_\theta) \leq C_n\delta_n\right] d\Pi(\theta)}{(1 - C) \times \Pi(B_2(p_0, \epsilon_n/2)} \end{aligned} \tag{2.31}$$

Using Theorem 5 again, we obtain an upper bound for the display above with

$$\frac{\rho_n(t_\epsilon; C_n; \delta_n)}{(1 - C) \times \Pi(B_2(p_0, \epsilon_n/2)(1 + o(1))}.$$

Using the prior Assumption 3, we can choose t_ϵ such that

$$\Big(\frac{C_n\delta_n}{t_\epsilon} + \frac{1}{nt_\epsilon^2}\Big)/\epsilon_n^\kappa = 1/M_n$$

for some arbitrarily slowly increasing sequence $M_n > 0$. We choose $t_\epsilon = M_n C_n \delta_n / \epsilon_n^\kappa + \sqrt{M_n} n^{-1/2} / \epsilon_n^{\kappa/2}$. Since $\delta_n \gtrsim n^{-1/2}$ and $\epsilon_n^{-\kappa} \geq \epsilon_n^{-\kappa/2}$, the overall rate is then driven by $\epsilon_n + t_\epsilon = \epsilon_n + \widetilde{M}_n \delta_n \epsilon_n^{-\kappa}$, where $\widetilde{M}_n = M_n C_n$.

2.7.3.3 Proof of Theorem 9

Recall from Theorem 7 that with the accept-reject strategy, the true KL divergence $K(p_0, p_\theta)$ is contracting at the rate $\lambda_n = \epsilon_n + \widetilde{M}_n \delta_n \epsilon_n^{-\kappa}$, where $\widetilde{M}_n$ is a slowly increasing sequence that diverges faster than C_n. Consider the case when $\epsilon_n \gg \widetilde{M}_n \delta_n \epsilon_n^{-\kappa}$ or, equivalently, $\epsilon_n \gg \delta_n^{1/(\kappa+1)}$. Denote

$$x(\theta) = \epsilon_n^{-1} K(p_0, p_\theta) \quad \text{and} \quad f_n(\theta - \theta_0) = f\big(\epsilon_n^{-1/2}(\theta - \theta_0)\big).$$

We express the ABC posterior expectation of $f_n(\theta - \theta_0)$ for a non-negative and bounded function $f_n(\cdot)$ by

$$\begin{aligned}
P_0^{(n)} E_{\hat{\Pi}_{\epsilon_n}^{AR}}\Big[f_n(\theta - \theta_0)\Big] &= P_0^{(n)} \int f_n(\theta - \theta_0) \mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta \mid \boldsymbol{X}) \\
&= \underbrace{P_0^{(n)} \int f_n(\theta - \theta_0) \mathbb{I}[K(p_0, p_\theta) \leq \lambda_n, d(\hat{D}_{n,m}^\theta, D_\theta) \leq C_n \delta_n)] \mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta \mid \boldsymbol{X})}_{\text{(I)}} \\
&\quad + \underbrace{P_0^{(n)} \int f_n(\theta - \theta_0) \mathbb{I}[K(p_0, p_\theta) \leq \lambda_n, d(\hat{D}_{n,m}^\theta, D_\theta) > C_n \delta_n)] \mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta \mid \boldsymbol{X})}_{\text{(II)}} \\
&\quad + \underbrace{P_0^{(n)} \int f_n(\theta - \theta_0) \mathbb{I}[K(p_0, p_\theta) > \lambda_n] \mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta \mid \boldsymbol{X})}_{\text{(III)}}
\end{aligned}$$

where the term (III) can be controlled using Fubini's theorem and the concentration result

in Theorem 7 as follows

$$\begin{aligned}(\text{III}) &= \int f_n(\theta-\theta_0)P_0^{(n)}\mathbb{I}[K(p_0,p_\theta)>\lambda_n]\mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta\mid \boldsymbol{X}) \\ &\le \|f\|_\infty P_0^{(n)}\Pi\Big(K(p_0,p_\theta)>\lambda_n \mid \hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n\Big) = o(1).\end{aligned}$$

The second term (II) can be bounded similarly as

$$\begin{aligned}(\text{II}) &= P_0^{(n)}\int_{K(p_0,p_\theta)\le\lambda_n} f_n(\theta-\theta_0)\mathbb{I}[d(\hat{D}_{n,m}^\theta,D_\theta)>C_n\delta_n)]\mathrm{d}\hat{\Pi}_{\epsilon_n}^{AR}(\theta\mid\boldsymbol{X}) \\ &\le \|f\|_\infty \int_{K(p_0,p_\theta)\le\lambda_n} P(d(\hat{D}_{n,m}^\theta,D_\theta)>C_n\delta_n)\mathrm{d}\Pi(\theta) \\ &\le \|f\|_\infty \sup_{\theta\in\Theta} P(d(\hat{D}_{n,m}^\theta,D_\theta)>C_n\delta_n) = o(1)\end{aligned}$$

where we use the fact that $\sup_{\theta\in\Theta} P(d(\hat{D}_{n,m}^\theta,D_\theta)>C_n\delta_n)=o(1)$ from Lemma 2.

Thus, the asymptotic behavior is mainly determined by the term (I). Combined with the continuity of $\pi(\theta)$ at θ_0 we can re-write (I) as

$$\begin{aligned}(\text{I}) &= P_0^{(n)}\frac{\int_{K(p_0,p_\theta)\le\lambda_n}\pi(\theta)f_n(\theta-\theta_0)\tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}_{n,m}^\theta,D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta}{\int_{K(p_0,p_\theta)\le\lambda_n}\pi(\theta)\tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}_{n,m}^\theta,D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta} \\ &= P_0^{(n)}\frac{\int_{K(p_0,p_\theta)\le\lambda_n} f_n(\theta-\theta_0)\tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}_{n,m}^\theta,D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta}{\int_{K(p_0,p_\theta)\le\lambda_n}\tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}_{n,m}^\theta,D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta}(1+o(1))\end{aligned}$$

We have $x(\theta)\ge 0$ for all $\theta\in\Theta$ and since

$$\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) = \hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta) - K(p_0,p_\theta) + \epsilon_n x(\theta),$$

we can write

$$\begin{aligned}
&\tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})\leq\epsilon_n, d(\hat{D}^{\theta}_{n,m},D_{\theta})\leq C_n\delta_n\big]\\
&\qquad\geq \tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K(p_0,p_{\theta})\right|\leq\epsilon_n(1-x(\theta)), d(\hat{D}^{\theta}_{n,m},D_{\theta})\leq C_n\delta_n\right]\\
&\qquad= 1-\tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K(p_0,p_{\theta})\right|>\epsilon_n(1-x(\theta)), d(\hat{D}^{\theta}_{n,m},D_{\theta})\leq C_n\delta_n\right]. \quad (2.32)
\end{aligned}$$

Denoting with

$$\tilde{Z}(\boldsymbol{X})=\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\tilde{P}^{(m)}\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K(p_0,p_{\theta})\right|>\epsilon_n\big(1-x(\theta)\big), d(\hat{D}^{\theta}_{n,m},D_{\theta})\leq C_n\delta_n\right]\mathrm{d}\theta.$$

Using Markov's inequality and Fubini's theorem we have, for every $\tilde{C}>0$,

$$\begin{aligned}
&P_0^{(n)}(\tilde{Z}(\boldsymbol{X})>\tilde{C})\\
&\leq\frac{1}{\tilde{C}}\int\limits_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}P\left[\left|\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^{\theta})-K(p_0,p_{\theta})\right|>\epsilon_n\big(1-x(\theta)\big), d(\hat{D}^{\theta}_{n,m},D_{\theta})\leq C_n\delta_n\right]\mathrm{d}\theta\\
&\leq\frac{1}{\tilde{C}}\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\rho_n\big(\epsilon_n(1-x(\theta));C_n;\delta_n\big)\mathrm{d}\theta\\
&\leq\frac{1}{\tilde{C}}\times\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta\times\sup_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\rho_n\big(\epsilon_n(1-x(\theta));C_n;\delta_n\big)\\
&\leq\frac{\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta}{\tilde{C}}\times\rho_n(M_nC_n\delta_n;C_n;\delta_n)\\
&=\frac{\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta}{\tilde{C}}\times O\left(\frac{1}{M_n}+\frac{1}{nM_n^2C_n^2\delta_n^2}\right).
\end{aligned}$$

where $\rho_n(\cdot;C_n;\delta_n)$ is defined in Theorem 5. Thus, we can define a set $\tilde{\Omega}_n(\tilde{C}_n)$, for some for some arbitrarily slowly increasing sequence $\tilde{C}_n>0$, and $\tilde{C}_n=O(1/M_n)$, as

$$\tilde{\Omega}_n(\tilde{C}_n)=\left\{\boldsymbol{X}:\tilde{Z}(\boldsymbol{X})\leq\tilde{C}_n\times\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta\right\}.$$

Then we have that $P_0^{(n)}(\tilde{\Omega}_n(\tilde{C}_n)^c)=o(1)$. Recall the inequality in (2.32), on $\tilde{\Omega}_n(\tilde{C}_n)$, we

have that

$$\int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta$$
$$\ge \int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \mathrm{d}\theta - \tilde{Z}(\boldsymbol{X}) = \int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \mathrm{d}\theta\times(1-o(1))$$

And this quantity is upper-bounded by $\int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta$. Therefore, we can conclude that

$$\int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta = \int_{x(\theta)\le 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \mathrm{d}\theta\big(1+o(1)\big).$$

Note that $x(\theta)\le 1-M_nC_n\delta_n/\epsilon_n$ implies that $K(p_0,p_\theta)\le\epsilon_n-M_nC_n\delta_n$. On this set $\tilde{\Omega}_n(\tilde{C}_n)$, we can further lower bound the denominator as

$$\begin{aligned}
&\int_{K(p_0,p_\theta)\le\lambda_n} \tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta\\
&\quad=\underbrace{\int_{K(p_0,p_\theta)\le\epsilon_n-M_nC_n\delta_n}\mathrm{d}\theta(1+o(1))}_{D_1}\\
&\quad\quad+\underbrace{\int_{\epsilon_n-M_nC_n\delta_n<K(p_0,p_\theta)\le\lambda_n} \tilde{P}^{(m)}\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\le\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\le C_n\delta_n\big]\mathrm{d}\theta}_{D_2}.
\end{aligned}$$

Next, we show that the second term D_2 is $o(D_1)$. Let $u=\epsilon_n^{-1/2}(\theta-\theta_0)$. Under Assumption 5, we have $K(p_0,p_\theta)=\frac{1}{2}(\theta-\theta_0)'I(\theta_0)(\theta-\theta_0)\{1+o(1)\}$, which is $x(\theta_0+\sqrt{\epsilon}_nu)=\frac{1}{2}u'I(\theta_0)u$. Since $I(\theta_0)$ is positive definite we can write

$$\frac{D_2}{D_1}\le\frac{\int_{1-M_nC_n\delta_n/\epsilon_n<x(\theta)\le 1+\widetilde{M}_n\delta_n/\epsilon_n^{\kappa+1}}\mathrm{d}\theta}{\int_{x(\theta)\le 1-M_nC_n\delta_n/\epsilon_n}\mathrm{d}\theta} \tag{2.33}$$

$$\le\frac{\int_{2(1-M_nC_n\delta_n/\epsilon_n)\le u'I(\theta_0)u\le 2(1+\widetilde{M}_n\delta_n/\epsilon_n^{\kappa+1})}\mathrm{d}\theta}{\int_{u'I(\theta_0)u\le 2(1-M_nC_n\delta_n/\epsilon_n)}\mathrm{d}\theta}\lesssim\frac{M_nC_n\delta_n}{\epsilon_n}+\frac{\widetilde{M}_n\delta_n}{\epsilon_n^{\kappa+1}}=o(1). \tag{2.34}$$

where the approximation follows from the fact that $\epsilon_n \gg \delta_n^{1/(\kappa+1)}$ and because the denominator is approximating the integral $\int_{u'I(\theta_0)u\leq 2} \mathrm{d}\theta$ and the numerator is the length of shrinking intervals. Combining the above results, we find that, on the set $\tilde{\Omega}_n(\tilde{C}_n)$, the denominator can be lower-bounded by $\int_{x(\theta)\leq 1-\frac{M_n C_n \delta_n}{\epsilon_n}} \mathrm{d}\theta\{1+o(1)\}$.

This implies

$$\begin{aligned}(\mathrm{I}) &= \frac{\int_{K(p_0,p_\theta)\leq\lambda_n} f_n(\theta-\theta_0) P\big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\leq\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\leq C_n\delta_n\big]\mathrm{d}\theta}{(1+o(1))\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta}(1+o(1))+o(1)\\ &= (N_1+N_2)\{1+o(1)\}+o(1). \end{aligned} \tag{2.35}$$

with

$$N_1 \equiv \frac{\int\limits_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}} f\Big(\epsilon_n^{-1/2}(\theta-\theta_0)\Big)\mathrm{d}\theta}{\int\limits_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta} \tag{2.36}$$

and

$$N_2 \equiv \frac{\int\limits_{K(p_0,p_\theta)\leq\lambda_n} \mathbb{I}\Big[x(\theta)>1-\frac{M_nC_n\delta_n}{\epsilon_n}\Big] f\Big(\epsilon_n^{-1/2}(\theta-\theta_0)\Big) P\Big[\hat{K}(\boldsymbol{X},\tilde{\boldsymbol{X}}^\theta)\leq\epsilon_n, d(\hat{D}^\theta_{n,m},D_\theta)\leq C_n\delta_n\Big]\mathrm{d}\theta}{\int\limits_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}}\mathrm{d}\theta}, \tag{2.37}$$

where the second equality follows from the fact that $x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}$ leads to $K(p_0,p_\theta)\leq \epsilon_n - M_nC_n\delta_n$ and then $K(p_0,p_\theta)\leq\lambda_n$ is trivially satisfied and where the last $o(1)$ comes from the set $\tilde{\Omega}_n(\tilde{C}_n)^c$.

Since we have $\epsilon_n \gg \delta_n$, with $u=\epsilon_n^{-1/2}(\theta-\theta_0)$, the first term is approximately equal to

$$N_1 = \frac{\int_{K(p_0,p_{\theta_0+\sqrt{\epsilon_n}u})\leq\epsilon_n} f(u)\mathrm{d}u}{\int_{K(p_0,p_{\theta_0+\sqrt{\epsilon_n}u})\leq\epsilon_n}\mathrm{d}u}\left(1+o(1)\right).$$

Using Assumption 5 again, we have

$$\int_{K(p_0,p_{\theta_0+\sqrt{\epsilon_n}u})\leq\epsilon_n} \mathrm{d}u = \int_{\frac{1}{2}\sqrt{\epsilon_n}u'I(\theta)\sqrt{\epsilon_n}u\leq\epsilon_n} \mathrm{d}u + o(1) = \int_{u'I(\theta_0)u\leq 2} \mathrm{d}u + o(1).$$

This leads to

$$N_1 = \frac{\int_{u'I(\theta_0)u\leq 2} f(u)\mathrm{d}u}{\int_{u'I(\theta_0)u\leq 2} \mathrm{d}u}\left(1+o(1)\right).$$

Next, we show that the fraction N_2 converges to 0. The numerator can be simplified as an integral over $1-\frac{M_nC_n\delta_n}{\epsilon_n} < x(\theta) \leq 1+\frac{\widetilde{M}_n\delta_n}{\epsilon_n^{\kappa+1}}$, which can be bounded by (2.33) as

$$\begin{aligned} N_2 &\leq \frac{\|f\|_\infty \int_{1-\frac{M_nC_n\delta_n}{\epsilon_n}<x(\theta)\leq 1+\frac{\widetilde{M}_n\delta_n}{\epsilon_n^{\kappa+1}}} \mathrm{d}\theta}{\int_{x(\theta)\leq 1-\frac{M_nC_n\delta_n}{\epsilon_n}} \mathrm{d}\theta} \\ &\leq \|f\|_\infty \frac{D_2}{D_1} = o(1). \end{aligned}$$

Since we have $\epsilon_n^{\kappa+1} \gg \delta_n$, putting all the terms together, we obtain that the $P_0^{(n)}$-averaged ABC posterior distribution of $\epsilon_n^{-1/2}(\theta-\theta_0)$ is asymptotically uniform over the ellipsoid $\{u : u'I(\theta_0)u \leq 2\}$.

CHAPTER 3

ADVERSARIAL BAYESIAN SIMULATION

In the absence of explicit or tractable likelihoods, Bayesians often resort to approximate Bayesian computation (ABC) for inference. Our work bridges ABC with deep neural implicit samplers based on generative adversarial networks (GANs) and adversarial variational Bayes. Both ABC and GANs compare aspects of observed and fake data to simulate from posteriors and likelihoods, respectively. We develop a Bayesian GAN (B-GAN) sampler that directly targets the posterior by solving an adversarial optimization problem. B-GAN is driven by a deterministic mapping learned on the ABC reference by *conditional* GANs. Once the mapping has been trained, iid posterior samples are obtained by filtering noise at a negligible additional cost. We propose two post-processing local refinements using (1) data-driven proposals with importance reweighing, and (2) variational Bayes. We support our findings with frequentist-Bayesian results, showing that the typical total variation distance between the true and approximate posteriors converges to zero for certain neural network generators and discriminators. Our findings on simulated data show highly competitive performance relative to some of the most recent likelihood-free posterior simulators.

3.1 ABC and Beyond

For a practitioner, much of the value of the Bayesian inferential approach hinges on the ability to compute the entire posterior distribution. Very often, it is easier to infer data-generating probability distributions through simulator models rather than likelihood functions. However, Bayesian computation with simulator models can be particularly grueling.

. Adopted from Yuexi Wang and Veronika Ročková. Adversarial bayesian simulation. *arXiv preprint arXiv:2208.12113*, 2022.

We assume that $\theta \in \Theta \subset \mathbb{R}^d$ is a parameter controlling a simulator-based model that gives rise to a data vector $X_\theta^{(n)} = (X_1, \ldots, X_n)' \sim P_\theta^{(n)}$ which is *not* necessarily iid. The model may be provided by a probabilistic program that can be easily simulated but its implicit likelihood $p_\theta^{(n)} = \pi(X^{(n)} \,|\, \theta)$ cannot be evaluated. For an unknown inferential target $\theta_0 \in \Theta$, our goal is to approximate the post-data inferential density (i.e. the posterior)

$$\pi(\theta \,|\, X_0^{(n)}) \propto p_\theta^{(n)}(X_0^{(n)})\pi(\theta), \tag{3.1}$$

where $X_0^{(n)}$ denotes the observed data. We allow for the possibility that *both* the likelihood $p_\theta^{(n)}(\cdot)$ and/or the prior $\pi(\theta)$ are analytically intractable but easy to draw from.

Without the obligation to build a model, Approximate Bayesian Computation (ABC) (Beaumont et al., 2002; Sisson et al., 2018) provides an approximation to the posterior (3.1) by matching aspects of observed and fake data. This is accomplished via forward simulation of the so-called ABC reference table $\{(\theta_j, X_j^{(n)})\}_{j=1}^T$ where θ_j's have been sampled from the prior $\pi(\theta)$ and fake data $X_j^{(n)}$'s have been sampled from the likelihood $p_{\theta_j}^{(n)}(\cdot)$. In order to keep only plausible parameter draws, this table is then filtered through an Accept/Reject mechanism to weed out parameter values θ_j for which the summary statistics of the fake and observed data were too far. Our work, albeit not being an ABC method per-se, builds off of recent ABC and simulation-based Bayesian inference innovations described below.

ABC Regression adjustment (Beaumont et al., 2002; Beaumont, 2003; Blum and François, 2010) is a post-processing step that re-weights and re-adjusts the location of θ_j's gathered by rejection ABC by fitting a (weighted) regression model of θ_j's onto summary statistics $s_j = s(X_j^{(n)})$. Such a model can be regarded as provisional density estimator of $\pi(\theta \,|\, X^{(n)})$ derived from $s(X^{(n)})$ under certain regression distributional assumptions. More flexible conditional density estimators, such as neural mixture density networks (Papamakarios and Murray, 2016; Lueckmann et al., 2017), have been successfully integrated into ABC without the burden of choosing summary statistics. Our approach is related to these developments.

However, we do not attempt to learn a flexible parametric approximation to the posterior (or the likelihood (Lueckmann et al., 2019; Papamakarios et al., 2019)).Instead, we find an *implicit* neural sampler from an approximation to $\pi(\theta \mid X^{(n)})$ by training Generative Adversarial Networks (Goodfellow et al., 2016) on the ABC reference table. GANs have been originally conceived to simulate from complex likelihoods by contrasting observed and fake data. ABC, on the other hand, contrasts observed and fake data to simulate from complex posteriors. Bringing together these two approaches, we propose the B-GAN posterior sampler, an incarnation of conditional GANs (Gauthier, 2014; Mirza and Osindero, 2014; Athey et al., 2021; Zhou et al., 2022) for likelihood-free Bayesian simulation. By contrasting the ABC reference table with a fake dataset under the same marginal distribution $\pi(X^{(n)})$, B-GAN learns to simulate from an approximation to the conditional distribution $\pi(\theta \mid X^{(n)})$. Similarly as (Papamakarios and Murray, 2016) and (Lueckmann et al., 2017), our method is also global in the sense that it learns $\pi(\theta \mid X^{(n)})$ for *any* $X^{(n)}$, not necessarily $X_0^{(n)}$. More perfected posterior reconstructions can be obtained with post-processing steps that zoom in onto the posterior distribution evaluated at $X_0^{(n)}$. We consider two such refinements based on: (1) reinforcement learning with importance sampling, and (2) adversarial variational Bayes. We describe each approach below.

Simple rejection ABC may require exceedingly many trials to obtain only a few accepted samples when the posterior $\pi(\theta \mid X_0^{(n)})$ is much narrower than the prior $\pi(\theta)$ (see e.g. Marjoram et al. (2003); Sisson et al. (2007); Beaumont et al. (2009)). This has motivated query-efficient ABC techniques which intelligently decide where to propose next (see Jarvenpaa et al. (2020); Hennig and Schuler (2012) for decision-theoretic reasoning or Järvenpää et al. (2019) and Gutmann and Corander (2016) for implementations based on Bayesian optimization and surrogate models). Alternatively, Lueckmann et al. (2017) learn a Bayesian mixture density network approximating the posterior over multiple rounds of adaptively chosen simulations and use more flexible proposal distributions (not necessarily the prior) with a

built-in importance-reweighting scheme. A similar strategy was used in Papamakarios et al. (2019) who used a pilot run of mixture density networks to learn the proposal distribution for the next round. Although $X_0^{(n)}$ *is not* used in B-GAN training, it can be used in the proposal inside the ABC reference table. Similarly as in Papamakarios et al. (2019), we use $X_0^{(n)}$ to construct a flexible proposal (i.e. an empirical Bayes prior) and convert the draws to posterior samples under the original prior by importance reweighing. This 'reinforcement learning' refinement substantially improves the reconstruction accuracy and can be justified by theory.

Our vanilla B-GAN sampler uses contrastive learning (Gutmann et al., 2018; Durkan et al., 2020) to estimate the conditional distribution $\pi(\theta \mid X^{(n)})$ for *any* $X^{(n)}$. Since $X_0^{(n)}$ is used only at the evaluation stage (not the training stage), we can custom-make the sampler to $X_0^{(n)}$ by using the B-GAN output (or the output after reinforcement learning) as an initialization for implicit variational Bayes optimization (Tran et al., 2017; Huszár, 2017). Implicit variational Bayes attempts to approximate the posterior using densities which are defined implicitly by a push-forward mapping. B-GAN also trains such a mapping but the generator will have never seen observed data. At later stages of B-GAN training, we can thereby modify the objective function for the generator so that it minimizes a lower bound to the marginal likelihood. Since the likelihood cannot be evaluated, we use contrastive learning inside the variational objective to compute the lower bound (Huszár, 2017; Tran et al., 2017). We consider the joint-contrastive form (Huszár, 2017; Durkan et al., 2020), where the classifier is still trained to learn the joint likelihood ratio using the ABC reference table (similarly as in B-GAN). However, the generator is now trained on $X_0^{(n)}$ by maximizing the evidence lower bound. This algorithm is related to the B-GAN simulator, but uses $X_0^{(n)}$ during the training stage.

Contrastive learning has been used inside Bayesian likelihood-free sampling algorithms before (see e.g. Wang et al. (2022a); Gutmann et al. (2018); Kaji and Ročková (2022)).

Both Wang et al. (2022a) and Kaji and Ročková (2022) assume iid data with a large enough sample size n to be able to apply classification algorithms for each iteration of Metropolis Hastings and ABC, respectively. Our approach does not require iid data and works even with $n = 1$. We also do not require to run classification at each posterior simulation step.

We show highly competitive performance of our methods (relative to state-of-the art likelihood-free Bayesian methods) on several simulated examples. While conceptually related methodology has occurred before (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Ramesh et al., 2022), theory supporting these likelihood-free Bayesian approaches has been lacking. We provide new frequentist-Bayesian theoretical results for the typical variational distance between the true and approximated posteriors. We analyze Wasserstein versions of both the B-GAN algorithm as well as adversarial variational Bayes. With properly tuned neural networks, we show that this distance goes to zero as $n \to \infty$ with large enough ABC reference tables.

The outline of our paper is as follows. Section 3.2 reviews conditional GANs and introduces the Bayesian GAN sampler together with the reinforcement adjustments. In Section 3.3, we describe another local enhancement strategy inspired by implicit variational Bayes. In Section 3.4, we investigate the theoretical guarantees of the B-GAN posteriors. The performance of our methods is illustrated on simulated datasets in Section 3.5. In Section 5.6, we conclude with a discussion.

3.2 Adversarial Bayes

Generative Adversarial Networks (GANs) (Goodfellow et al., 2016) are a game-theoretic construct in artificial intelligence designed to simulate from likelihoods over complex objects. GANs involve two machines playing a game against one another. A *Generator* aims to deceive a *Discriminator* by simulating fake samples that resemble observed data while, at the same time, the Discriminator learns to tell the fake and real data apart. This process iterates

until the generated data are indistinguishable by the Discriminator and can be regarded as genuine likelihood samples. Below, we review several recent GAN innovations and propose an incarnation for simulation from a posterior as opposed to a likelihood.

3.2.1 Vanilla GANs

In its simplest form, GANs learn how to implicitly simulate from the likelihood $p^{(n)}_{\theta_0}(\cdot)$ using only its realizations $X^{(n)}_0 \in \mathcal{X}$ where $X^{(n)}_0 \sim p^{(n)}_{\theta_0}$ when θ_0 is unknown. Recall that draws from implicit distributions can be obtained by passing a random noise vector $Z \in \mathcal{Z} \subseteq \mathbb{R}^d$ through a non-stochastic push-forward mapping $g_{\boldsymbol{\beta}}(\cdot) : \mathcal{Z} \to \mathcal{X}$. The original GANs formulation (Goodfellow et al., 2016) involves a Generator, specified by the mapping $g_{\boldsymbol{\beta}}(\cdot)$, that attempts to generate samples similar to $X^{(n)}_0$ by filtering Z, i.e.

$$X^{(n)} = (X_1, \ldots, X_n)' \text{ where } X_i = g_{\boldsymbol{\beta}}(Z_i) \quad \text{with} \quad Z_i \overset{iid}{\sim} \pi_Z(Z) \quad \text{so that} \quad X^{(n)} \sim p^{(n)}_{\theta}.$$

The generative coefficients $\boldsymbol{\beta}$ are iteratively updated depending on the feedback received from the Discriminator. The Discriminator, specified by a mapping $d(\cdot) : \mathcal{X} \to (0, 1)$, gauges similarity between $X^{(n)}$ and $X^{(n)}_0$ with a discrepancy between their (empirical) distributions. Hereafter we use X to denote a generic dataset as $X \in \mathcal{X}$ for simplicity of notation. At a population level, a standard way of comparing two distributions, say $P^{(n)}_{\theta_0}$ and $P^{(n)}_{\theta}$, is with the symmetrical Jensen-Shannon divergence[1] which can be equivalently written as a solution to a particular optimization problem

$$\mathrm{JS}(P^{(n)}_{\theta}, P^{(n)}_{\theta_0}) = \ln 2 + 0.5 \times \sup_{d:\mathcal{X}\to(0,1)} \left\{ E_{X\sim P^{(n)}_{\theta}} \ln \big[d(X)\big] + E_{X\sim P^{(n)}_{\theta_0}} \ln \big[1 - d(X)\big] \right\}. \tag{3.2}$$

1. defined as $\mathrm{JS}(P^{(n)}_{\theta}, P^{(n)}_{\theta_0}) = \mathrm{KL}(P^{(n)}_{\theta} \mid (P^{(n)}_{\theta} + P^{(n)}_{\theta_0})/2) + \mathrm{KL}(P^{(n)}_{\theta_0} \mid (P^{(n)}_{\theta} + P^{(n)}_{\theta_0})/2)$

The optimal Discriminator $d^*(\cdot)$, solving the optimization (3.2), appears to be $d^*(X) = p_\theta^{(n)}(X)/\left[p_\theta^{(n)}(X)+p_{\theta_0}^{(n)}(X)\right]$ (Goodfellow et al., 2014, Proposition 1). The optimal Generator is then defined through the optimal value $\boldsymbol{\beta}^*$ which leaves the Discriminator maximally confused, i.e. $d^*(X) = 1/2$ and therefore $p_{\theta_0}^{(n)}(X) = p_\theta^{(n)}(X)$ uniformly over $\mathcal{X}$.

Despite the nice connection to likelihood ratios, original GANs (Goodfellow et al., 2014) may suffer from training difficulties when the discriminator becomes too proficient early on (Gulrajani et al., 2017; Arjovsky and Bottou, 2017). Alternative divergences have been implemented inside GANs that are less prone to these issues. For example, the Wasserstein distance (Arjovsky and Bottou, 2017) also admits a dual representation

$$d_W\left(P_\theta^{(n)}, P_{\theta_0}^{(n)}\right) = \sup_{f\in\mathcal{F}_W} \left| E_{X\sim P_\theta^{(n)}} f(X) - E_{X\sim P_{\theta_0}^{(n)}} f(X) \right| \tag{3.3}$$

where $\mathcal{F}_W = \{f : \|f\|_L \leq 1\}$ are functions with a Lipschitz semi-norm $\|f\|_L$ at most one. The function $f(\cdot)$ is often referred to as the *Critic*. In our implementations, we will concentrate on the Wasserstein version of GANs (Arjovsky et al., 2017).

3.2.2 Conditional GANs

While originally intended for simulating from likelihoods underlying observed data, GANs can be extended to simulating from distributions *conditional on* observed data. Certain aspects of conditional GANs (cGANs) have been investigated earlier (Gauthier, 2014; Mirza and Osindero, 2014) in various contexts including causal inference (Athey et al., 2021) or non-parametric regression (Zhou et al., 2022). Our work situates conditional GANs firmly within the context of ABC and likelihood-free posterior simulation. Before we describe our development in Section 3.2.3, we first introduce the terminology of cGANs within a Bayesian context. We will intentionally denote with X the conditioning variables and focus on the conditional distribution $\pi(\theta \mid X)$ for the inferential parameter $\theta \in \Theta$ with a prior

$\pi(\theta)$. Similarly as with vanilla GANs in Section 3.2.1, cGANs again involve two adversaries represented by two mappings.

Definition 14. *(Generator) We define a deterministic* generative model *as a mapping* $g : (\mathcal{Z} \times \mathcal{X}) \to \Theta$ *that filters noise random variables* $Z \in \mathcal{Z}$ *to obtain samples from an implicit conditional density* $\pi_g(\theta \mid X)$*. This conditional model then defines an implicit joint model* $\pi_g(X, \theta) = \pi_g(\theta \mid X)\pi(X)$*, where* $\pi(X) = \int_{\mathcal{X}} p_\theta^{(n)}(X)\pi(\theta)\mathrm{d}\theta$ *is the marginal likelihood.*

Definition 15. *(Discriminator) We define a deterministic* discriminative model *as a mapping* $d : (\mathcal{X} \times \Theta) \to (0, 1)$ *which predicts whether the data pair* (X, θ) *came from* $\pi(X, \theta)$ *(label* 1*) or from* $\pi_g(X, \theta)$ *(label* 0*).*

The main distinguishing feature, compared to vanilla GANs, is that the conditioning random vector X enters *both* mappings. The task is to flexibly parametrize $g_{\boldsymbol{\beta}}(\cdot)$, e.g. using neural networks as will be seen later, in order to approximate the joint density model $\pi(X, \theta)$ as closely as possible. Ideally, we would like to recover an (oracle) function $g^* : \mathcal{Z} \times \mathcal{X} \to \Theta$ such that the conditional distribution of $g^*(Z, X)$ given X is the same as $\pi(\theta \mid X)$. The existence of such an oracle g^* is encouraged by the noise-outsourcing lemma from probability theory (Kallenberg, 2002; Zhou et al., 2022) which we reiterate below using Gaussian Z.

Lemma 16. *(Zhou et al., 2022, Lemma 2.1) Let* (X, θ) *be a random pair taking values in* $\mathcal{X} \times \Theta$ *with a joint distribution* $\pi(X, \theta)$*. Then, for any given* $d \geq 1$*, there exists a random vector* $Z \sim \pi_Z = N(0, I_d)$ *and a Borel-measurable function* $g^* : \mathbb{R}^d \times \mathcal{X} \to \Theta$ *such that* Z *is independent of* X *and* $(X, \theta) = (X, g^*(Z, X))$ *almost surely.*

The premise of conditional GANs rests in the fact that matching two joint distributions, while fixing a marginal distribution, is equivalent to matching conditional distributions. This implies that $g^*(Z, X)$, given X, is indeed distributed according to $\pi(\theta \mid X)$. The question remains how to find the oracle mapping g^* in practice. Using the Jensen-Shannon divergence

(3.2) for comparing the joint distributions $\pi(X,\theta)$ and $\pi_g(X,\theta)$, the oracle mapping g^* emerges at the equilibrium of a minimax game.

Lemma 17. *Consider a minimax game* $(g^*, d^*) = \arg\min_{g\in\mathcal{G}} \max_{d\in\mathcal{D}} D(g,d)$ *prescribed by*

$$D(g,d) = E_{(X,\theta)\sim\pi(X,\theta)} \log d(X,\theta) + E_{X\sim\pi(X),Z\sim\pi_Z} \log[1 - d(X, g(Z,X)]. \tag{3.4}$$

Assume that $\mathcal{G}$ *and* $\mathcal{D}$ *are universal approximators capable of representing any function* $g : (\mathcal{Z}\times\mathcal{X}) \to \Theta$ *and* $d : (\mathcal{X}\times\Theta) \to (0,1)$*, respectively. Then, uniformly on* $\mathcal{X}$ *and* Θ*, the solution* (g^*, d^*) *satisfies*

$$\pi_{g^*}(\theta \mid X) = \frac{\pi(X,\theta)}{\pi(X)} = \pi(\theta \mid X) \quad \textit{and} \quad d_g^*(X,\theta) = \frac{\pi(X,\theta)}{\pi(X,\theta) + \pi_g(X,\theta)} \quad \textit{for any } g \in \mathcal{G}.$$

Proof. The expression for $d_g^*(X,\theta)$ is an immediate consequence of Proposition 1 in Goodfellow et al. (2016). Plugging-in this expression into (3.4), we find that

$$g^* = \arg\min_{g\in\mathcal{G}} \Big(E_{(X,\theta)\sim\pi(X,\theta)} \log d_g^*(X,\theta) + E_{X\sim\pi(X),z\sim\pi_Z} \log[1 - d_g^*(X, g(Z,X))] \Big),$$

According to Theorem 1 of Goodfellow et al. (2016), the minimum is achieved if and only if $\pi_{g^*}(X,\theta) = \pi(X,\theta) = \pi(\theta \mid X)\pi(X)$. The fact that $\pi(X,\theta)$ and $\pi_{g^*}(X,\theta)$ have the same marginal $\pi(X)$ implies the expression for $\pi_{g^*}(\theta \mid X)$. □

While the Jensen-Shannon (JS) variant of the cGAN game in Lemma 17 is conceptually compelling and can be supported by theory (Zhou et al., 2022), implementation difficulties may arise (such as convergence failure or mode collapse (Arjovsky and Bottou, 2017)). We thereby focus on the Wasserstein variant of the game from Lemma 17

$$(g^*, f^*) = \arg\min_{g\in\mathcal{G}} \max_{f\in\mathcal{F}} \Big| E_{X\sim\pi(X),Z\sim\pi_Z} f(X, g(Z,X)) - E_{(\theta,X)\sim\pi(X,\theta)} f(X,\theta) \Big|, \tag{3.5}$$

where, using $\mathcal{F} = \mathcal{F}_W$, g^* minimizes the Wasserstein distance between $\pi_g(X, \theta)$ and $\pi(X, \theta)$.

3.2.3 Bayesian GANs

We now submerge the conditional GAN framework within the context of Bayesian simulation. To implement the adversarial game (3.5) in practice, one needs to (a) parametrize $\mathcal{F}$ and $\mathcal{G}$ (for instance using neural networks) and (b) to replace the expectations in (3.5) with empirical counterparts. Both of these steps will introduce approximation error. We provide theoretical insights later in Section 3.4.

We assume that the generator class $\mathcal{G} = \{g_{\boldsymbol{\beta}} : (\mathcal{Z} \times \mathcal{X}) \to \Theta \text{ where } \boldsymbol{\beta} \in \mathbb{R}^G\}$ is parametrized with $\boldsymbol{\beta}$ and the critic class $\mathcal{F} = \{f_{\boldsymbol{\omega}} : (\mathcal{X} \times \Theta) \to \mathbb{R} \text{ where } \boldsymbol{\omega} \in \mathbb{R}^C\}$ is parametrized with $\boldsymbol{\omega}$. We consider neural networks (with ReLU activations) in our implementations and support this choice with theory (Corollary 23).

For the empirical version, one can use the ABC reference table which consists of simulated data pairs $\{(\theta_j, X_j^{(n)})\}_{j=1}^T$ generated from the joint model $\pi(X^{(n)}, \theta) = p_\theta^{(n)}(X^{(n)})\pi(\theta)$ under the prior $\pi(\theta)$. We can think of each draw $X_j^{(n)} = (X_1^j, \ldots, X_n^j)' \sim p_\theta^{(n)}$ as either a stacked collection of n iid vectors X_i^j (of dimension q) sampled from a product likelihood $\prod_{i=1}^n p_\theta(X_i^j)$ or a single observation from a general likelihood $p_\theta^{(n)}(\cdot)$ which may not necessarily assume independence (e.g. one time series observation vector). From now on, we will simply denote $X_j^{(n)}$ with X_j, similarly for $X_0^{(n)}$ and $X^{(n)}$.

We can break the relationship between θ and X in the ABC reference table by contrasting these data pairs with another dataset consisting of $\{(g(Z_j, X_j), X_j)\}_{j=1}^T$ where Z_j's have been sampled from $\pi_Z(\cdot)$. Keeping the same X_j's essentially means that we are keeping the same marginal. The dataset $\{(g(Z_j, X_j), X_j)\}_{j=1}^T$ encapsulates iid draws from $\pi_g(X, \theta)$. These two datasets can be then used to approximate the expectations in (3.4) and (3.5). A high-level description of an algorithm for solving the Jensen-Shannon (JS) version of the game from Lemma 17 is outlined in Algorithm 2 of Wang and Ročková (2022, Appendix

Algorithm 3: *B-GAN for Bayesian Simulation (Wasserstein Version).*

Input

Prior $\pi(\theta)$, observed data X_0 and noise distribution $\pi_Z(\cdot)$

Training

Initialize network parameters $\boldsymbol{\omega}^{(0)} = 0$ and $\boldsymbol{\beta}^{(0)} = 0$

Reference Table

For $j = 1, \ldots, T$: Generate (X_j, θ_j) where $\theta_j \sim \pi(\theta)$ and $X_j \sim P_{\theta_j}^{(n)}$.

Wasserstein GAN

For $t = 1, \ldots, N$:

Critic Update (N_{critic} steps): For $k = 1, \ldots, N_{\text{critic}}$

Generate $Z_j \sim \pi_Z(z)$ for $j = 1, \ldots, T$.

Generate $\epsilon_j \overset{iid}{\sim} U[0,1]$ and set $\bar{\theta}_j = \epsilon_j \theta_j + (1-\epsilon_j) g_{\boldsymbol{\beta}^{(t-1)}}(Z_j, X_j)$ for $j = 1, \ldots, T$.

Update $\boldsymbol{\omega}^{(t)}$ by applying stochastic gradient descent on (3.6) with the penalty (3.7).

Generator Update (single step)

Generate noise $Z_j \sim \pi_Z(z)$ for $j = 1, \ldots, N$.

Update $\boldsymbol{\beta}^{(t)}$ by applying stochastic gradient descent on (3.6).

Posterior Simulation:

For $i = 1, \ldots, M$: Simulate $Z_i \sim \pi_Z(z)$ and set $\widetilde{\theta}_i = g_{\boldsymbol{\beta}^{(N)}}(Z_i, X_0)$.

C) [2]. As mentioned earlier, the JS version may suffer from training issues (Arjovsky and Bottou, 2017). We provide an illustration of such issues using convergence diagnostics on a toy example in Wang and Ročková (2022, Appendix C). In our implementations, we thereby consider the empirical version of (3.5) which again involves simulated datasets $\{(\theta_j, X_j)\}_{j=1}^T$ and $\{Z_j\}_{j=1}^T \overset{iid}{\sim} \pi_Z(\cdot)$ to obtain

$$\widehat{\boldsymbol{\beta}}_T = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[\max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}_W} \left| \sum_{j=1}^T f_{\boldsymbol{\omega}}\big(X_j, g_{\boldsymbol{\beta}}(Z_j, X_j)\big) - \sum_{j=1}^T f_{\boldsymbol{\omega}}(X_j, \theta_j) \right| \right]. \tag{3.6}$$

One particular way of solving this problem is summarized in Algorithm 3. In terms of the constraint on $\boldsymbol{\omega}$ to ensure the Lipschitz condition $\|f_{\boldsymbol{\omega}}\|_L \leq 1$, the original Wasserstein GANs implementation (Arjovsky and Bottou, 2017) used gradient clipping, which may lead to computational issues. Alternatively, Gulrajani et al. (2017) imposed a soft version of the

2. During the course of this work, we found out that this algorithm was proposed in a simultaneous and independent work (Ramesh et al., 2022).

constraint with a penalty[3] on the gradient of $f_{\boldsymbol{\omega}}$ with respect to a convex combination[4] of the two contrasting datasets. Similarly as Athey et al. (2021), we adopt the one-sided penalty only with respect to θ_j

$$\lambda\Big\{\frac{1}{T}\sum_{j=1}^{T}\Big[\max\Big(0,\big\|\nabla_{\bar{\theta}}f_{\boldsymbol{\omega}}(X_j,\bar{\theta}_j)\big\|_2-1\Big)\Big]^2\Big\} \tag{3.7}$$

where $\bar{\theta}_j=\epsilon_j\theta_j+(1-\epsilon_j)g(Z_j,X_j)$ with the ϵ_j re-drawn from a uniform distribution at each step. The choice of λ is discussed in Wang and Ročková (2022, Appendix D). To stabilize gradients, the critic is updated multiple times (ideally until convergence) before each update of the generator, which is different from the JS version where such a stabilization may not be feasible (Arjovsky et al., 2017)

Our Bayesian GAN framework for posterior simulation (i.e. Algorithm 3 further referred to as B-GAN) consists of a neural sampler $g_{\widehat{\boldsymbol{\beta}}_T}(Z,X)$ which generates samples from an approximate posterior, i.e. conditioning on the observed data X_0, by filtering iid noise as follows

$$\widetilde{\theta}_j=g_{\widehat{\boldsymbol{\beta}}_T}(Z_j,X_0)\quad\text{where}\quad Z_j\overset{iid}{\sim}\pi_Z(\cdot)\quad\text{for}\quad j=1,\ldots,M. \tag{3.8}$$

When $g_{\widehat{\boldsymbol{\beta}}_T}$ is close to g^*, the samples $\widetilde{\theta}_j$ will arrive approximately from $\pi(\theta\,|\,X_0)$. One of the practical appeals of this sampling procedure is that, once the generator has been trained, the simulation cost is negligible. Note that our observed data $X_0\sim p^{(n)}_{\theta_0}(X)$ are *not involved* in the training stage, *only* in the simulation stage (3.8). We illustrate Algorithm 3 on a toy example. The configurations of our B-GAN networks and optimization hyperparameters are described in Wang and Ročková (2022, Appendix D.1).

Example 1 (Toy Example)**.** *This toy example (analyzed earlier in (Papamakarios et al.,*

3. They adopt the two-sided penalty encouraging the norm of the gradient to go towards 1 instead of just staying below 1 (one-sided penalty).

4. This is inspired by the fact that the optimal critic function contains straight lines with gradient norm 1 connecting coupled points from the contrasted distributions (Gulrajani et al., 2017, Proposition1).

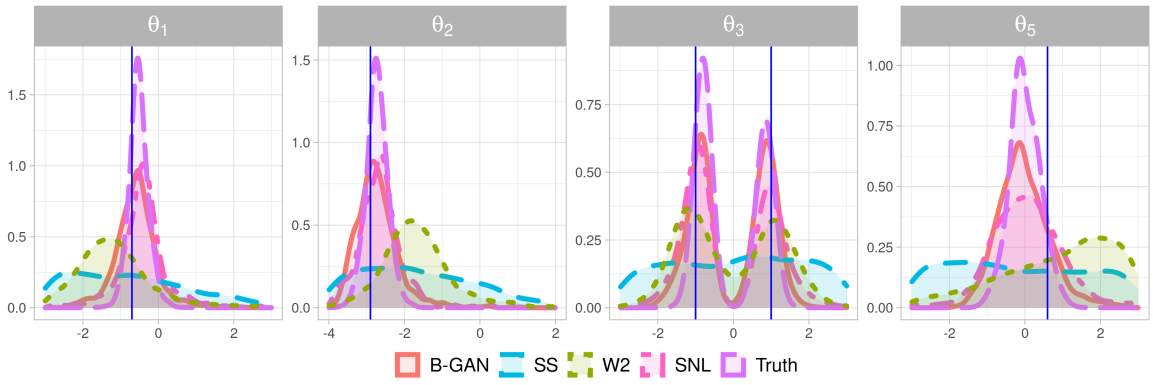


Figure 3.1: Approximated posteriors given by B-GAN, SNL, SS, and W2 for the toy example. The results for θ_4 are similar to θ_3 and thus not shown here.

2019)) exposes the fragility of ABC methods in a relatively simple setting. The experiment entails $n = 4$ *two-dimensional Gaussian observations* $X = (x_1, x_2, x_3, x_4)'$ *with* $x_j \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$ *parametrized by* $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)'$*, where*

$$\mu_{\boldsymbol{\theta}} = (\theta_1, \theta_2)' \quad \textit{and} \quad \Sigma_{\boldsymbol{\theta}} = \begin{pmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{pmatrix}$$

with $s_1 = \theta_3^2, s_2 = \theta_4^2$ *and* $\rho = tanh(\theta_5)$*. The parameters* $\boldsymbol{\theta}$ *are endowed with a uniform prior*[5]*. Approximating the posterior can be tricky because the signs of parameters* θ_3 *and* θ_4 *are not identifiable, yielding multimodality. We generate* X_0 *with parameters* $\boldsymbol{\theta}_0 = (-0.7, -2.9, -1.0, -0.9, 0.6)'$*. Since we have access to the true posterior, we can directly compare our posterior reconstructions with the truth. We compare B-GAN with (1) ABC using naive summary statistics (SS) (mean and variance), (2) 2-Wasserstein distance ABC (Bernton et al., 2019), and (3) Sequential Neural Likelihood (SNL) (Papamakarios et al., 2019) with the default setting suggested by the authors. We provide all implementation details in Wang and Ročková (2022, Appendix D.1). For each method, we obtained* $M = 1\,000$ *samples and plotted them in Figure 3.1. Our B-GAN approach as well as SNL nicely capture the multimodality. There appears to be an overestimation of variance (relative*

5. on $[-3, 3] \times [-4, 4] \times [-3, 3] \times [-3, 3] \times [-3, 3]$

Algorithm 4: *2-Step Refinement (B-GAN 2step)*

INPUT
Prior $\pi(\theta)$, observed data X_0 and noise distribution $\pi_Z(z)$
Training
Initialize network parameters $\boldsymbol{\omega}^{(0)} = 0$ and $\boldsymbol{\beta}^{(0)} = 0$
Pilot Run
Apply Algorithm 3 with $\pi(\theta)$ to learn $\widehat{g}_{pilot}(\cdot)$
Reference Table
Generate pairs $\{(X_j, \theta_j)\}_{j=1}^{T}$ where $\theta_j = \widehat{g}_{pilot}(Z_j, X_0)$ for $Z_j \sim \pi_Z$ and $X_j \sim P_{\theta_j}^{(n)}$.
Refinement
Apply Wasserstein GAN step in Algorithm 3 on $\{(X_j, \theta_j)\}_{j=1}^{T}$ and return $g\widehat{\boldsymbol{\beta}}_T(\cdot)$
Posterior Simulation
Simulate $\{Z_i\}_{i=1}^{M} \overset{iid}{\sim} \pi_Z(z)$ and set $\widetilde{\theta}_i = g\widehat{\boldsymbol{\beta}}_T(Z_i, X_0)$. Estimate $\hat{w}_i$ using either (3.9) or (3.10).
OUTPUT
Pairs of posterior samples and weights $(\widetilde{\theta}_1, \hat{w}_1), \ldots, (\widetilde{\theta}_M, \hat{w}_M)$

to the truth) which, as we explain later in Section 3.4, is expected because B-GAN is trained to perform well on average for any X, not *necessarily for X_0. This motivates our two refinement strategies: one based on active learning (Section 3.2.4) and one based on variational Bayes (Section 3.3).*

Similarly as with default ABC techniques, our B-GAN approach is not query-efficient, i.e. many prior guesses θ_j in the training dataset may be too far from the interesting areas with a likelihood support leaving only a few observations to learn about the conditional $\pi(\theta \,|\, X_0)$. The next section presents a two-step approach which uses X_0 for proposal construction in the ABC reference table to obtain more valuable data-points in the reference table.

3.2.4 Two-step Refinement

Our chief goal is to find a high-quality approximation to the conditional distribution $\pi(\theta \,|\, X)$ evaluated at the observed data $X = X_0$, not necessarily uniformly over the entire domain $\mathcal{X}$. However, the ABC reference table $\{(\theta_j, X_j)\}_{j=1}^{T}$ may not contain enough data points X_j in the vicinity of X_0 to train the simulator when the prior $\pi(\theta)$ is too vague. This can be remedied by generating a reference table using an auxiliary proposal distribution $\widetilde{\pi}(\theta)$

which is more likely to produce pseudo-observations X_j that are closer to X_0. For example, a pilot simulator $g\widehat{\boldsymbol{\beta}}_T(Z, X_0)$ in (3.8) obtained from Algorithm 3 under the original prior $\pi(\theta)$ can be used to guide simulations in the next round to sharpen the reconstruction accuracy around X_0 (Papamakarios and Murray, 2016). Training the generator[6] $\widetilde{g}_{\widehat{\boldsymbol{\beta}}_T}$ under the 'wrong' prior can be corrected for by importance re-weighting with weights $r(\theta) = \pi(\theta)/\widetilde{\pi}(\theta)$. Since the posterior $\widetilde{\pi}(\theta \mid X_0)r(\theta)$ is proportional to $\pi(\theta \mid X_0)$, reweighing the resulting samples $\widetilde{\theta}_j = \widetilde{g}_{\widehat{\boldsymbol{\beta}}_T}(Z, X_0^{(n)})$ with weights $w_j = r(\theta_j)$ will produce samples from an approximation to the original posterior (after normalization). Algorithm 4 summarizes this two-step strategy, referred to as B-GAN-2S.

Since the proposal density $\pi_{g\widehat{\boldsymbol{\beta}}_T}(\theta \mid X_0)$ obtained in the pilot run *may not* have an analytical form, computing the importance weights $w_j = \pi(\theta_j)/\pi_{g\widehat{\boldsymbol{\beta}}_T}(\theta_j \mid X_0)$ directly may not feasible. We consider leaky ReLU generative networks for which, in fact, the analytical form does exist (Liang, 2021). More generally, the density ratio $r(\theta)$ can always be approximated. For example, with a tractable prior $\pi(\theta)$ the importance weights w_j can be estimated by

$$\hat{w}_j = \frac{\pi(\theta_j)}{\hat{\pi}_{g\widehat{\boldsymbol{\beta}}_T}(\theta_j \mid X_0)} \tag{3.9}$$

where $\hat{\pi}_{g_{\widehat{\boldsymbol{\beta}}T}}(\theta_j \mid X_0)$ is a plugged-in kernel density estimator (KDE) estimator (Terrell and Scott, 1992). This is particularly useful and efficient when the parameter dimension is low. When the prior is also not tractable but simulatable, the weights w_j can be directly estimated from classification by contrasting datasets $\widetilde{\theta}_j \sim \widetilde{\pi}(\theta)$ (label '0') and $\theta_j \sim \pi(\theta)$ (label '1'). In particular, training a classifier $\tilde{D}$ (see e.g. (Cranmer et al., 2015; Durkan et al., 2020; Gutmann and Hyvärinen, 2012) for the explanation of the 'likelihood-ratio-trick' for

6. which simulates from an approximation to $\widetilde{\pi}(\theta \mid X_0) \propto p_{\theta_0}^{(n)}(X_0)\widetilde{\pi}(\theta)$

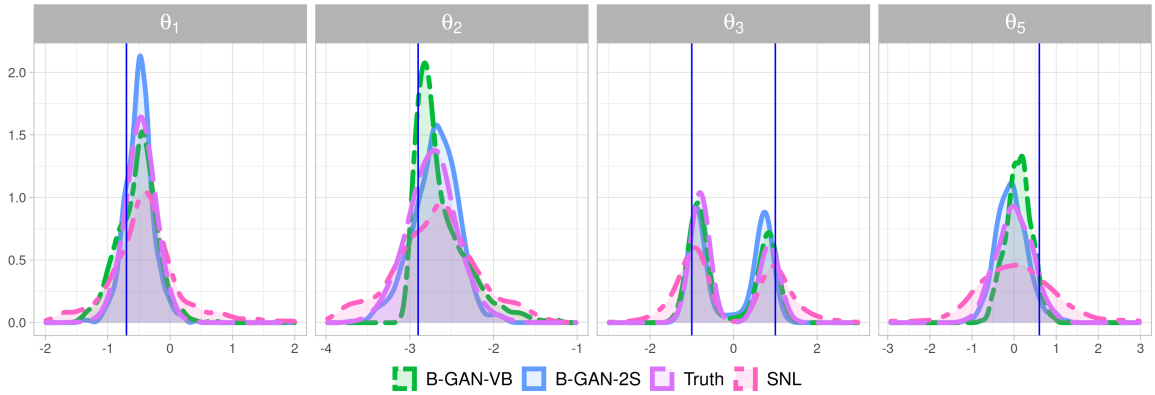


Figure 3.2: Posterior densities under the Gaussian model. The true parameter is $\boldsymbol{\theta}_0 = (-0.7, -2.9, -1.0, -0.9, 0.6)'$, while the signs of θ_3 and θ_4 are not identifiable.

classification based estimators), we can obtain

$$\hat{w}_j = \frac{\tilde{D}(\widetilde{\theta}_j)}{1 - \tilde{D}(\widetilde{\theta}_j)}. \tag{3.10}$$

Papamakarios and Murray (2016) used mixture density networks estimators of the conditional distribution $\pi(\theta \,|\, x)$ after a pilot run to learn the proposal distribution $\widetilde{\pi}(\theta)$. In order to obtain an analytically tractable Gaussian mixture representation, their proposal $\widetilde{\pi}(\theta)$ has to be Gaussian and it cannot be narrower than any of the mixture components. We do not require such assumptions. Lueckmann et al. (2019) instead propose to directly incorporate the weights $r(\theta)$ inside training and relax the Gaussianity assumption to avoid the variance instability. Similarly as in Lueckmann et al. (2019), we could also incorporate weights $\hat{w}_j$ inside the objective function, e.g. multiplying each summand in (3.6) by $\hat{w}_j$.

Example 2 (Toy Example Continued)**.** *We continue our exploration of the Toy Example 1. We now use the output from B-GAN (Algorithm 3) under the original uniform prior as a proposal distribution $\widetilde{\pi}(\theta)$ and generate training data $\{(X_j, \theta_j)\}_{j=1}^{T}$ with a marginal $\widetilde{\pi}(X) = \int_{\mathcal{X}} p_\theta^{(n)}(X)\widetilde{\pi}(\theta)d\theta$. To convert the generated posterior samples to the original uniform prior, we perform reweighing by $r(\theta) = \pi(\theta)/\widetilde{\pi}(\theta)$ using the kernel density estimator of $\widetilde{\pi}(\theta)$ obtained from the pilot B-GAN run in Algorithm 3. The number of training points used*

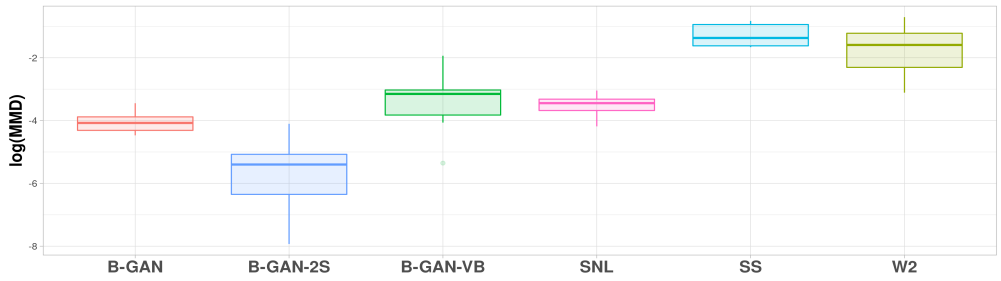


Figure 3.3: Maximum Mean Discrepancies (MMD, log scale) between the true posteriors and the approximated posteriors. The box-plots are computed from 10 repetitions.

in the second step is $T = 50\,000$. *We note that much smaller* T *could be used if one were to perform more sequential refinements, not just one. The re-weighted (and normalized) posterior is plotted against the truth, SNL and a variational Bayesian variant (introduced later in Section 3.3) in Figure 3.2. Both B-GAN and B-GAN-2S provide tighter approximations to the true posterior We repeated the experiment* 10 *times and report the Maximum Mean Discrepancies (MMD) (Gretton et al., 2012) between the true posterior and its approximations obtained from* $M = 1\,000$ *posterior draws for each method in Figure 3.3. Satisfyingly, B-GAN-2S yields the smallest MMD. We support this encouraging finding with our theoretical results in Section 3.4.*

Remark 18 (Computation When $n > 1$)**.** *There are various ways of handling larger* n*. For example, our approach can be deployed sequentially on batches of smaller iid samplers: using simulations from a posterior on the smaller batch as a prior for the next. Alternatively, we could stack the* n *replicates into one data vector and learn a higher-dimensional mapping. Deep learning has been used for large sets of covariates and we regard this option doable when* n *and the dimensionality of the vectors is not overwhelmingly large.*

In the two-step refinement, the observed data X_0 only contribute to the proposal distribution $\widetilde{\pi}(\theta)$, not the training of the simulator $g_{\widehat{\boldsymbol{\beta}}}(\cdot)$. In the next section, we consider a variational Bayes variant which does involve X_0 in training.

3.3 Adversarial Variational Bayes

Variational Bayes (VB) is an optimization-centric Bayesian inferential framework based on minimizing a divergence between approximate and real posteriors. VB typically reduces the infinite-dimensional optimization problem to a finite-dimensional one by molding approximations into structured parametric forms. Implicit distributions (defined as probabilistic programs) have the potential to yield finer and tighter VB posterior approximations (Huszár, 2017; Kingma and Welling, 2013; Tran et al., 2017; Titsias and Ruiz, 2019). This section highlights the connection between the implicit variational Bayes inference and our B-GAN framework (Algorithm 3 and 4), both of which target the posterior.

The VB setup consists of an (intractable) likelihood $p_\theta^{(n)}(\cdot)$, prior $\pi(\theta)$ and a class of posterior approximations $q_{\boldsymbol{\beta}}(\theta \,|\, X_0)$ indexed by $\boldsymbol{\beta}$. We are recycling the notation of $\boldsymbol{\beta}$ here to highlight the connection between the GAN generator and the the implicit variational generator. The goal of the VB approach is to find a set of parameters $\boldsymbol{\beta}^*$ that maximize a lower bound to the marginal likelihood

$$\log \pi(X_0) \geq \mathcal{L}(\boldsymbol{\beta}) \equiv \int \log \left(\frac{\pi(X_0, \theta)}{q_{\boldsymbol{\beta}}(\theta \,|\, X_0)} \right) q_{\boldsymbol{\beta}}(\theta \,|\, X_0) \mathrm{d}\theta. \tag{3.11}$$

The tightness of the inequality increases with expressiveness of the inference model $q_{\boldsymbol{\beta}}(\cdot)$, where the equality occurs when $q_{\boldsymbol{\beta}}(\theta \,|\, X_0) = \pi(\theta \,|\, X_0)$. Writing the evidence lower bound $\mathcal{L}(\boldsymbol{\beta}) = -\mathrm{KL}\left(q_{\boldsymbol{\beta}}(\theta \,|\, X_0) \| \pi(\theta \mid X_0)\right) + C$ in terms of Kullback-Leibler discrepancy, we have

$$\boldsymbol{\beta}^* = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\beta}} \mathrm{KL}\big(q_{\boldsymbol{\beta}}(\theta \,|\, X_0) \| \pi(\theta \mid X_0)\big). \tag{3.12}$$

Besides implicit likelihood, we also assume implicit posterior approximation $q_{\boldsymbol{\beta}}(\theta \,|\, X)$. Similarly as before in Section 3.2.2, this approximation can be defined by stochastic generative networks which take a simple distribution and transform it nonlinearly by a deep neural network. In general, we assume that the density $q_{\boldsymbol{\beta}}(\theta \,|\, X_0)$ *does not* have any particular form

but, instead, that its samples are obtained by passing noise $Z \sim \pi_Z$ through a deterministic generator mapping $g_{\boldsymbol{\beta}}(Z, X)$ parametrized by $\boldsymbol{\beta} \in \mathbb{R}^G$.

With implicit likelihoods, where only forward simulation is possible, one may need to use contrastive learning (Bickel et al., 2007) to optimize the lower bound (3.11). Ideally, we would estimate the conditional density ratio by two contrasting datasets $\theta \sim \pi(\theta \mid X_0)$ (label '1') and $\tilde{\theta} \sim q_{\boldsymbol{\beta}}(\theta \mid X_0)$ (label '0'). However, this is not feasible since the posterior distribution $\pi(\theta \mid X_0)$ is unknown. Fortunately, we can simulate from (and contrast) two *joint* distributions with a different conditional, given X, but the same marginal $\pi(X)$. We define

$$\frac{d^*_{g_{\boldsymbol{\beta}}}(X, \theta)}{1 - d^*_{g_{\boldsymbol{\beta}}}(X, \theta)} = \frac{\pi(X, \theta)}{q_{\boldsymbol{\beta}}(\theta \mid X)\pi(X)}, \tag{3.13}$$

where $d^*_{g_{\boldsymbol{\beta}}} : (\mathcal{X} \times \Theta) \to (0, 1)$ can be viewed as the 'oracle classifier' when distinguishing data pairs (θ, X) as arising from either $\pi_{g_{\boldsymbol{\beta}}}(\theta, X) \equiv q_{\boldsymbol{\beta}}(\theta \mid X)\pi(X)$ or $\pi(\theta, X)$. We have seen this oracle classifier earlier in Lemma 17, with the variational conditional distribution $q_{\boldsymbol{\beta}}(\theta \mid X)$ taking the place of $\pi_g(\theta \mid X)$.

The variational lower bound (3.11) can be re-written as

$$\mathcal{L}(\boldsymbol{\beta}) \equiv E_{\theta \sim q_{\boldsymbol{\beta}}(\theta \mid X_0)} \Big[\operatorname{logit}\big(d^*_{g_{\boldsymbol{\beta}}}(X_0, \theta)\big) \Big] + C. \tag{3.14}$$

For implicit approximating distributions $q_{\boldsymbol{\beta}}(\theta \mid X)$ it may not be possible to directly optimize (3.14) with respect to $\boldsymbol{\beta}$ (even using the re-parametrization trick and stochastic gradient descent (Kingma and Welling, 2019)). The lower bound (3.14) depends on the oracle classifier which is unknown. Going back to Lemma 17, we note that $d^*_{g_{\boldsymbol{\beta}}}(\theta, X)$ is a solution to an infinite-dimensional classification problem under the entropy loss (Goodfellow et al., 2016)

$$d^*_{g_{\boldsymbol{\beta}}}(\theta, X) = \arg\max_{d \in \mathcal{D}} D(g_{\boldsymbol{\beta}}, d). \tag{3.15}$$

where $D(g_{\boldsymbol{\beta}}, d)$ was defined in (3.4) in Lemma 17. Although the oracle classifier is unknown,

it can be estimated by solving a classification problem (3.15) by focusing on a particular class of classifiers $\mathcal{D} = \{d_{\boldsymbol{\phi}} : (\mathcal{X} \times \Omega) \to (0,1); \boldsymbol{\phi} \in \mathbb{R}^C\}$, for instance neural networks indexed by parameters $\boldsymbol{\phi}$. We can thereby reframe maximizing the evidence lower bound (ELBO) in (3.14) as an adversarial game between two agents optimizing different objectives: (1) the *Generator* $g_{\boldsymbol{\beta}}(Z, X)$ tries to maximize ELBO, (2) the *Discriminator* $d_{\boldsymbol{\phi}}(X, \theta)$ tries to distinguish between the two joint distributions.

The idea of replacing aspects of the evidence lower bound with adversarial objectives occurred earlier (Mescheder et al., 2017; Huszár, 2017; Tran et al., 2017). These papers focus on hierarchical models with latent variables when either the prior or the likelihood (or both) are implicit. Instead of focusing on maximum likelihood estimation (Mescheder et al., 2017), we focus purely on VB inference with approximate posteriors $q_{\boldsymbol{\beta}}(\theta \mid X_0)$ when θ is assigned a prior.

All adversarial variational Bayesian papers have considered the KL formulation. Similarly as with the JS version of B-GAN , we have seen training issues. We thereby resort to Wasserstein formulation (Ranganath et al., 2016; Ambrogioni et al., 2018) which minimizes the Wasserstein distance (instead of the KL divergence in (3.12))

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \sup_{f_{\boldsymbol{\omega}} \in \mathcal{F}_W} \left| E_{\theta \sim q_{\boldsymbol{\beta}}(\theta \mid X_0)} \left(\frac{\pi(\theta \mid X_0)}{q_{\boldsymbol{\beta}}(\theta \mid X_0)} - 1 \right) f_{\boldsymbol{\omega}}(\theta) \right|, \tag{3.16}$$

where we have rewritten the Wasserstein distance (3.3) intentionally using density ratios and where the critic $f_{\boldsymbol{\omega}}$ operates only on Θ. The parameters $\boldsymbol{\beta}^*$ can be approximated by replacing the density ratio with a classification estimator for the joint distributions (as discussed in Wang and Ročková (2022, Appendix B)). We also implement a different algorithm which trains a critic on the joint space $\Theta \times \mathcal{X}$ without having to directly estimate the joint likelihood ratio.

Choosing $\mathcal{F}$ that is symmetrical[7], we start from $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\omega}^{(0)}$ and using the reference table $\{(\theta_j, X_j)\}_{j=1}^T \overset{iid}{\sim} \pi(\theta, X)$ with $\{Z_j\}_{j=1}^T \overset{iid}{\sim} \pi_Z(\cdot)$, we

- update $\boldsymbol{\omega}^{(t+1)}$ to approximate the Wasserstein distance, given $\boldsymbol{\beta}^{(t)}$,

$$\boldsymbol{\omega}^{(t+1)} = \arg \max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left[\sum_{j=1}^T f_{\boldsymbol{\omega}}(X_j, g_{\boldsymbol{\beta}^{(t)}}(Z_j, X_j)) - \sum_{j=1}^T f_{\boldsymbol{\omega}}(X_j, \theta_j) \right] \tag{3.17}$$

- update $\boldsymbol{\beta}^{(t+1)}$ to minimize the distance *evaluated at* X_0, given $\boldsymbol{\omega}^{(t+1)}$,

$$\boldsymbol{\beta}^{(t+1)} = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[\sum_{j=1}^T f_{\boldsymbol{\omega}^{(t+1)}}(X_0, g_{\boldsymbol{\beta}}(Z_j, X_0)) + C \right], \tag{3.18}$$

where C does not depend on $\boldsymbol{\beta}$, given the most recent update $\boldsymbol{\omega}^{(t+1)}$. This iterative procedure can be regarded as targeting the estimator

$$\widehat{\boldsymbol{\beta}}_T = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left| \frac{1}{T} \sum_{j=1}^T f_{\boldsymbol{\omega}(\boldsymbol{\beta})} \left(X_0, g_{\boldsymbol{\beta}}(Z_j, X_0) \right) - E_{\theta \sim \pi(\theta \,|\, X_0)} f_{\boldsymbol{\omega}(\boldsymbol{\beta})} \left(X_0, \theta \right) \right| \tag{3.19}$$

where

$$\boldsymbol{\omega}(\boldsymbol{\beta}) = \arg \max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}_W} \left| \frac{1}{T} \sum_{j=1}^T f_{\boldsymbol{\omega}} \left(X_j, g_{\boldsymbol{\beta}}(Z_j, X_j) \right) - \frac{1}{T} \sum_{j=1}^T f_{\boldsymbol{\omega}}(X_j, \theta_j) \right|.$$

Proceeding iteratively, given the update $\boldsymbol{\omega}^{(t+1)}$, the second term in (3.19) does not depend on $\boldsymbol{\beta}$ and is consumed by the constant C in (3.18). This iterative procedure resembles the GAN simulator (3.8). There are, however, fundamental differences compared to Algorithm 3. Unlike coefficients $\widehat{\boldsymbol{\beta}}_T$ in (3.6) of the B-GAN generator, coefficients $\widehat{\boldsymbol{\beta}}_T$ in (3.19) are *trained on* X_0 under the variational loss (3.14). This means that the generator $g_{\boldsymbol{\beta}}(Z, X_0)$ directly targets $\pi(\theta \,|\, X)$ evaluated at $X = X_0$. We would thereby expect this version to work better than Algorithm 3 which does not use X_0 at all.

7. i.e. $f_{\boldsymbol{\omega}} \in \mathcal{F} \to -f_{\boldsymbol{\omega}} \in \mathcal{F}$

Indeed, on the toy simulated example in Example 2 we can see that the VB variant produces tighter reconstructions relative to the B-GAN approach. The performance, however, is not uniformly better than Algorithm 4. We provide a snapshot from another repetition in Wang and Ročková (2022, Appendix B.1), where the spikiness of B-GAN-VB (especially obvious when estimating θ_2) may explain why the MMDs between B-GAN-VB and the true posteriors are larger than for B-GAN-2S in Figure 3.3. The algorithmic description is provided in Algorithm 5 and implementation details of B-GAN-VB are provided in Wang and Ročková (2022, Appendix B). We initialize the generator at a network returned from Algorithm 4.

Algorithm 5: *Adversarial Variational Bayes (Wasserstein Version)*

INPUT

Prior $\pi(\theta)$, observed data X_0 and noise distribution $\pi_Z(z)$

Training

Pilot Run

Apply Algorithm 4 with $\pi(\theta)$ to learn $\widehat{g}_{pilot}(\cdot)$

Initialize critic network $\boldsymbol{\omega}^{(0)} = 0$ and the generator network at $g_{\boldsymbol{\beta}^{(0)}} = g_{pilot}(\cdot)$

Reference Table

Generate pairs $\{(X_j, \theta_j)\}_{j=1}^{T}$ where $\theta_j = \widehat{g}_{pilot}(Z_j, X_0)$ for $Z_j \sim \pi_Z$ and $X_j \sim P_{\theta_j}^{(n)}$.

WGAN Training

For $t = 1, \ldots, N$:

Critic Update (N_{critic} steps): Same as in Critic Update of Algorithm 3

Generator Update (single step)

Generate noise $Z_j \sim \pi_Z(z)$ for $j = 1, \ldots, N$.

Update $\boldsymbol{\beta}^{(t)}$ by applying stochastic gradient descent on (3.18).

Posterior Simulation

Simulate $\{Z_i\}_{i=1}^{M} \overset{iid}{\sim} \pi_Z(z)$ and set $\widetilde{\theta}_i = g_{\boldsymbol{\beta}^{(N)}}(Z_i, X_0)$.

Estimate $\hat{w}_i$ using either (3.9) or (3.10).

OUTPUT

Pairs of posterior samples and weights $(\widetilde{\theta}_1, \hat{w}_1), \ldots, (\widetilde{\theta}_M, \hat{w}_M)$

3.4 Theory

The purpose of this section is to provide theoretical solidification for the implicit posterior simulators in Algorithms 3 to 5. We will quantify the typical total variation (TV) distance between the actual posterior and its approximation and illustrate that with carefully chosen

neural generators and discriminators, the expected total variation distance vanishes as $n \to \infty$. We will continue denoting $X^{(n)}$ simply by X.

We define with $\nu = P_\theta^{(n)} \otimes \Pi$ the joint measure on $\mathcal{X} \times \Theta$ with a density $\pi(X, \theta) = p_\theta^{(n)}(X)\pi(\theta)$. The goal is to approximate this measure with μ_g defined semi-implicitly through a density function $\pi_g(X, \theta) = \pi(X)\pi_g(\theta \mid X)$ where $\pi(X) = \int \pi(X, \theta)\mathrm{d}\theta$ is the marginal likelihood and where the samples from the density $\pi_g(\theta \mid X)$ are obtained by the *Generator* in Definition 14. Thus, by keeping the marginal distribution the same, the distribution $\pi_g(\theta \mid X)$ is ultimately approximating the conditional distribution $\pi(\theta \mid X)$. The quality of the approximation will be gauged under the integral probability metric[8] (IPM)

$$d_{\mathcal{F}}(\mu_g, \nu) = \sup_{f \in \mathcal{F}} \left| E_{(X,\theta)\sim\mu_g} f(X, \theta) - E_{(X,\theta)\sim\nu} f(X, \theta) \right|, \tag{3.20}$$

where $\mathcal{F}$ is a class of evaluation metrics[9]. The IPM metric (3.20), due to shared marginals of the two distributions, satisfies

$$d_{\mathcal{F}}(\mu_g, \nu) \leq E_X d_{\mathcal{F}}\big(\mu_g(X), \nu(X)\big), \tag{3.21}$$

where $\mu_g(X)$ and $\nu(X)$ denote the *conditional* measures with densities $\pi_g(\theta \,|X)$ and $\pi(\theta \,|X)$. At the population level, the B-GAN (Algorithm 3) minimax game finds an equilibrium

$$g^* = \min_{g \in \mathcal{G}} d_{\mathcal{F}}(\mu_g, \nu),$$

where $\mathcal{G}$ is a class of generating functions (that underlie the implicit distribution μ_g).

Typically, both $\mathcal{F}$ and $\mathcal{G}$ would be parameterized by neural networks with the hope that the discriminator networks can closely approximate the metric $d_{\mathcal{F}}$ and that the generator

8. The absolute value can be removed due to the Monge-Rubinstein dual (Villani, 2008).

9. For example, Lipschitz-1 functions yield the Wasserstein-1 metric and functions bounded by 1 yield the TV metric.

networks can flexibly represent distributions. In practice, one would obtain a data-driven estimator based on the *empirical* distribution $\bar{\nu}_T$ of (θ_j, X_j) for $1 \leq j \leq T$ and the *empirical* distribution $\bar{\mu}_g$ of $(g(Z_j, X_j), X_j)$ where $Z_j \sim \pi_Z$ for $1 \leq j \leq T$. Assuming that $\mathcal{G} = \{g_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \mathbb{R}^G\}$, the B-GAN estimator can be written as

$$\widehat{\boldsymbol{\beta}}_T \in \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left| E_{\bar{\mu}_g} f_{\boldsymbol{\omega}}\big(X, g_{\boldsymbol{\beta}}(Z, X)\big) - E_{\bar{\nu}_T} f_{\boldsymbol{\omega}}(X, \theta) \right|. \tag{3.22}$$

For brevity, we will often denote the generator density $\pi_{g_{\boldsymbol{\beta}}}(\cdot)$ (see Definition 14) simply by $\pi_{\boldsymbol{\beta}}(\cdot)$ and similarly for $\mu_{g_{\boldsymbol{\beta}}}$. The next Theorem provides an upper bound on the typical total variation (TV) distance between true and the approximated posterior measures $\nu(X_0)$ and $\mu_{\widehat{\boldsymbol{\beta}}_T}(X_0)$ with densities $\pi(\theta \,|\, X_0)$ and $\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \,|\, X_0)$, respectively. The total variation distance can be upper bounded by three terms: (1) the ability of the critic to tell the true model apart from the approximating model

$$\mathcal{A}_1(\mathcal{F}, \mathcal{G}) \equiv \mathbb{E} \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \log \frac{\pi(\theta \,|\, X)}{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \,|\, X)} - f_{\omega}(X, \theta) \right\|_{\infty} \tag{3.23}$$

(2) the ability of the generator to approximate the average true posterior

$$\mathcal{A}_2(\mathcal{G}) \equiv \inf_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[E_X \left\| \log \frac{\pi_{\boldsymbol{\beta}}(\theta \,|\, X)}{\pi(\theta \,|\, X)} \right\|_{\infty} \right]^{1/2}, \tag{3.24}$$

and (3) the complexity of the (generating and) critic function classes measured the pseudo-dimension $Pdim(\cdot)$ and defined in Definition 2 in (Bartlett et al., 2017). We denote with $\mathcal{H} = \big\{h_{\boldsymbol{\omega},\boldsymbol{\beta}} : h_{\boldsymbol{\omega},\boldsymbol{\beta}}(Z, X) = f_{\boldsymbol{\omega}}\big(g_{\boldsymbol{\beta}}(Z, X), X\big)\big\}$ a structured composition of networks $f_{\boldsymbol{\omega}} \in \mathcal{F}$ and $g_{\boldsymbol{\beta}} \in \mathcal{G}$.

Theorem 19. *Let $\widehat{\boldsymbol{\beta}}_T$ be as in (3.22) where $\mathcal{F} = \{f : \|f\|_{\infty} \leq B\}$ for some $B > 0$. Denote with $\mathbb{E}$ the expectation with respect to $\{(X_j, \theta_j)\}_{j=1}^T \overset{iid}{\sim} \pi(X, \theta)$ and $\{Z_j\}_{j=1}^T \overset{iid}{\sim} \pi_Z$ in the*

reference table. Assume that the prior satisfies

$$\Pi[B_n(\theta_0;\epsilon)] \geq \mathrm{e}^{-C_2 n\epsilon^2} \quad \textit{for some } C_2 > 0 \textit{ and } \epsilon > 0. \tag{3.25}$$

Then for $T \geq Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$ *we have for any* $C > 0$

$$P_{\theta_0}^{(n)} \mathbb{E}\, d_{TV}^2 \left(\nu(X_0), \mu\widehat{\boldsymbol{\beta}}_T(X_0)\right) \leq \mathcal{C}_n^T(\epsilon, C),$$

where, for some $\widetilde{C} > 0$ *and* $Pmax \equiv Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$,

$$C_n^T(\epsilon, C) = \frac{1}{C^2 n\epsilon^2} + \frac{\mathrm{e}^{(1+C_2+C)n\epsilon^2}}{4}\left[2\mathcal{A}_1(\mathcal{F},\mathcal{G}) + \frac{B\,\mathcal{A}_2(\mathcal{G})}{\sqrt{2}} + 4\widetilde{C}\,B\sqrt{\frac{\log T \times Pmax}{T}}\right].$$

Proof. Section 3.7.1.1.

From (3.21), it can be seen that Algorithm 3 targets a lower bound to the *average* Wasserstein distance between the posterior $\pi(\theta \,|\, X)$ and $\pi_g(\theta \,|\, X)$ after margining over $\pi(X)$. In other words, Algorithm 3 *is not* necessarily targeting $\pi(\theta \,|\, X_0)$. The 2-step enhancement in Algorithm 4 provides more data draws in the ABC table that more closely resemble X_0. Theorem 19 applies to Algorithm 4 as well with slight modifications.

Corollary 20. *(2step B-GAN) Assume that* $\widehat{\boldsymbol{\beta}}_T$ *in* (3.22) *is learned under the proposal distribution* $\widetilde{\pi}(\theta)$ *and denote with* $\widetilde{\mathbb{E}}$ *the expectation of the reference table under* $\widetilde{\pi}(\theta)$*. Assume that the original prior* $\pi(\theta)$ *satisfies* (3.25)*. Then the importance re-weighted posterior reconstruction from Algorithm 4 satisfies the statement in Theorem 19 with* $\mathbb{E}$ *replaced by* $\widetilde{\mathbb{E}}$ *and with* $\widetilde{\mathcal{A}}_2(\mathcal{G}) \equiv \inf_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \int_{\mathcal{X}} \widetilde{\pi}(X) \left\| \log \frac{\pi_{\boldsymbol{\beta}}(\theta \,|\, X)}{\pi(\theta \,|\, X)} \right\|_\infty \mathrm{d}X$ *and* $\widetilde{\mathcal{A}}_1(\mathcal{F},\mathcal{G}) \equiv \widetilde{\mathbb{E}} \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \frac{\pi(\theta)}{\widetilde{\pi}(\theta)} \log \frac{\pi(\theta \,|\, X)}{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \,|\, X)} - f_\omega(X,\theta) \right\|_\infty$.

Proof. Section 3.7.1.2.

Remark 21. *Zhou et al. (2022) provided theoretical results for the total variation distance between* joint *distributions using the Jensen-Shannon version of conditional GANs. Contrastingly, we provide frequentist-Bayesian results, quantifying the typical total variation distance*

between the true and approximate posteriors. In addition, we use the Wasserstein GANs, building on oracle inequalities established in Liang (2021).

The (nonasymptotic) bounds for the *typical* TV distance in Theorem 19 and Corollary 20 are not refined enough to fully appreciate the benefits of the 2-step enhancement. In Section 3.7.1.3, we provide an intuitive explanation for why the 2-step refinement version works so well in practice on each particular realization of X_0 (not only on average over many realizations X_0). We also provide a version of Theorem 19 for adversarial variational Bayes (Algorithm 5) in Theorem 26 (Section 3.7.1.5).

One of the appeals of our method is that it can accommodate situations where $n = 1$ (such as spatiotemporal dependent datasets). For n independent copies of observed data vectors, it is desirable to inquire when the average total variation distance in Theorem 19 converges to zero as $n \to \infty$ for a suitable choice of ϵ and C (potentially depending on n). We provide one example using discriminator feed-forward networks $\mathcal{F}$ with a ReLU activation function $\sigma_{ReLU}(x) = \max\{x, 0\}$ which have good approximation properties (Schmidt-Hieber, 2020). For the generator networks, we need to make sure that the resulting density $\pi_g(\theta \mid X)$ is absolutely continuous. To this end, we consider $\mathcal{G}$ that are leaky feed-forward ReLU neural networks with an activation function $\sigma^a_{ReLU}(x) = \max\{ax, x\}$ for some $0 < a \leq 1$. Liang (2021, Section 3.3) shows that these networks indeed produce densities that are absolutely continuous and provided a closed-form expression for the log density.

Definition 22. *We denote with $\mathcal{F}^B_L(S, W)$ a class of feed-forward ReLU neural networks f with depth L (i.e. the number of hidden layers plus one), width W and size S (total number of parameters in the network) such that $\|f\|_\infty \leq B$. The width is defined as $W = \max\{w_0, \ldots, w_L\}$ where w_l is the width of the l^{th} layer with w_0 the input data dimension and w_L the output dimension. With $\mathcal{G}^B_L(S, W)$, we denote the leaky ReLU neural networks with the same meaning of parameters.*

The following remark (clarified in Section 3.7.1.4 in the Appendix) warrants optimism

when using neural networks for the generator and the discriminator. We formulate the remark for Algorithm 3 and note that a similar conclusion holds for Algorithm 4 as well.

Remark 23. *Assume that the joint distribution $\pi(\theta, X)$ is realizable in the sense that there exists $g_{\boldsymbol{\beta}_0} \in \mathcal{G}_{L_0}^{B_0}(S_0, W_0)$ such that $\pi(\theta, X) = \pi_{g_{\boldsymbol{\beta}_0}}(\theta, X)$. Assume that $\mathcal{G}_{L^*}^{B^*}(S^*, W^*) \subseteq G_{L_0}^{B_0}(S_0, W_0)$ is a class of leaky ReLU generative networks indexed by $\boldsymbol{\beta}$ where $\|\boldsymbol{\beta}_0\|_\infty \vee \|\boldsymbol{\beta}\|_\infty \leq b$ for some $b > 0$. Assume that $\mathcal{F} = \mathcal{F}_L^B(S, W)$ are ReLU discriminator networks and π_Z is uniform on $[0,1]^d$. Assume the prior concentration (3.25) is satisfied with $\epsilon_n > 0$ such that $\epsilon_n = O(1/\sqrt{n})$. For each arbitrarily slowly increasing sequence C_n, there exists $L, S, W > 0$ and T_n such that we have $P_{\theta_0}^{(n)} \mathbb{E} d_{TV}^2 \left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}_{T_n}}(X_0)\right) = o(1)$ as $n \to \infty$.*

3.5 Performance Evaluation

This section demonstrates very promising performance of our B-GAN approaches in Algorithm 3 (B-GAN), Algorithm 4 (B-GAN-2S) and Algorithm 5 (B-GAN-VB) and on simulated examples relative to other Bayesian likelihood-free methods (plain ABC using summary statistics (SS); 2-Wasserstein distance ABC by (Bernton et al., 2019); Sequential Neural Likelihood (SNL) (Papamakarios et al., 2019) with default settings). The implementation details of our methods and the counterparts are described in Appendix D.2 for the Lotka-Volterra model and Appendix D.3 for the Boom-and-Bust model in Wang and Ročková (2022)

3.5.1 Lotka-Volterra Model

The Lotka-Volterra (LV) predator-prey model (Wilkinson, 2018) is one of the classical likelihood-free examples and describes population evolutions in ecosystems where predators interact with prey. The state of the population is prescribed deterministically via a system of ordinary differential equations (ODEs). Inference for such models is challenging because the transition density is intractable. However, simulation from the model is possible, which

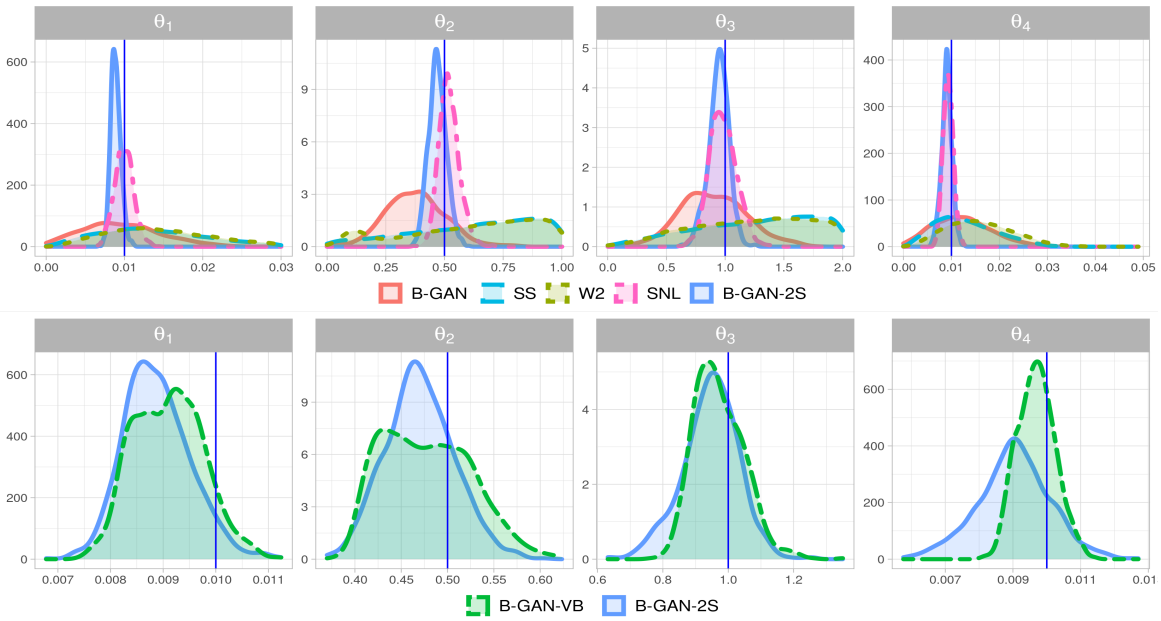


Figure 3.4: Approximate posterior densities under the Lotka-Volterra Model. The true parameter vector (marked by vertical lines) is $\boldsymbol{\theta}_0 = (0.01, 0.5, 1, 0.01)'$.

makes it a natural candidate for simulator-based inference methods.

The model monitors population sizes of predators x_t and prey y_t over time t. The changes in states are determined by four parameters $\theta = (\theta_1, \ldots, \theta_4)'$ controlling: (1) the rate $r_1^t = \theta_1 x_t y_t$ of a predator being born; (2) the rate $r_2^t = \theta_2 x_t$ of a predator dying; (3) the rate $r_3^t = \theta_3 y_t$ of a prey being born; (4) the rate $r_4^t = \theta_4 x_t y_t$ of a prey dying. Given the initial population sizes (x_0, y_0) at time $t = 0$, the dynamics can be simulated using the Gillespie algorithm (Gillespie, 1977). The algorithm samples times to an event from an exponential distribution (with a rate $\sum_{j=1}^4 r_j^t$) and picks one of the four reactions with probabilities proportional to their individual rates r_j^t. We use the same setup as Kaji and Ročková (2022) where each simulation is started at $x_0 = 50$ and $y_0 = 100$ and state observations are recorded every 0.1 time units for a period of 20 time units, resulting in a series of 201 observations each.

The real data X_0 are generated with true values $\theta_0 = (0.01, 0.5, 1, 0.01)'$. The data vector X_0 is stretched into one $(201 \times 2 \times n)$ vector. The advantage of our approach is that it can

be used even for $n = 1$ when other methods (such as (Kaji and Ročková, 2022)) cannot. We focus on the $n = 1$ case here. We use an informative prior $\theta \in U(\Xi)$ with a restricted domain $\Xi = [0, 0.1] \times [0, 1] \times [0, 2] \times [0, 0.1]$ to make it easier for classical ABC methods (see (Kaji and Ročková, 2022)) and to make the GAN training more efficient. Previous analyses Papamakarios and Murray (2016) suggested summary statistics as the mean, log-variance, autocorrelation (at lag 1 and 2) of each series as well as their correlation. Papamakarios et al. (2019) also built their sequential neural network on top of this set of summary statistics. We build our model on the time series itself. This example is quite challenging due to the spikiness of the likelihood in very narrow areas of the parameter space (as explained in (Kaji and Ročková, 2022)). Our adversarial approaches are implemented using the Wasserstein versions.

To recover the posterior distributions, we draw $M = 1\,000$ samples for each method, except the top 1% for the ABC methods. A typical snapshot (for one particular data realization) of the approximated posteriors is given in Figure 3.4 and the summary statistics averaged over 10 repetitions are reported in Table 3.1. Since we do not have access to the true posterior, we look at the width of the 95% credible interval, its coverage (proportion of the 10 replications such that the true value is inside the credible interval), and bias of the posterior mean. Again, we observe that B-GAN-2S and B-GAN-VB outperforms B-GAN and SNL with a smaller bias and tighter variance. In Figure 3.4, B-GAN-VB appears to have smaller bias than B-GAN-2S when estimating all parameters but θ_1. The computation cost requirements are compared in Wang and Ročková (2022, Appendix D.5).

3.5.2 Simple Recruitment, Boom and Bust

Our second demonstration is on the simple recruitment, boom and bust model (Fasiolo et al., 2018). The model is prescribed by a discrete stochastic process, characterizing the fluctuation of the population size of a certain group over time. Given the population size N_t

	$\theta_1 = 0.01$		$\theta_2 = 0.5$		$\theta_3 = 1.0$		$\theta_4 = 0.01$	
(scale)	bias ($\times 10^{-3}$)	CI width ($\times 10^{-2}$)	bias ($\times 10^{-1}$)	CI width	bias	CI width	bias ($\times 10^{-2}$)	CI width ($\times 10^{-2}$)
B-GAN	4.15	1.89	1.09	0.45	0.24	1.00	0.49	2.18
B-GAN-2S	**0.70**	**0.21 (0.9)**	0.42	**0.10 (0.7)**	**0.11**	0.33 (0.9)	0.13	0.34 (0.8)
B-GAN-VB	1.02	0.25 (0.7)	**0.38**	0.11 (0.9)	**0.11**	**0.29 (0.8)**	**0.12**	**0.29 (0.7)**
SNL	1.05	0.44	0.45	0.17	0.13	0.48	0.15	0.52
SS	9.58	3.80	2.49	0.91	0.49	1.76	0.68	2.72
W2	10.99	4.02 (0.9)	2.42	0.84	0.47	1.73	0.79	2.82

Table 3.1: Summary statistics of the approximated posteriors under the Lotka-Volterra model (averaged over 10 repetitions). Bold fonts mark the best model of each column. The coverage of the 95% credible intervals are 1 unless otherwise noted in the parentheses.

and parameter $\boldsymbol{\theta} = (r, \kappa, \alpha, \beta)'$, the population size at the next timestep N_{t+1} follows the following distribution

$$N_{t+1} \sim \begin{cases} \text{Poisson}\big(N_t(1+r)\big) + \epsilon_t, & \text{if } N_t \leq \kappa \\ \text{Binom}\big(N_t, \alpha\big) + \epsilon_t, & \text{if } N_t > \kappa \end{cases},$$

where $\epsilon_t \sim \text{Pois}(\beta)$ is a stochastic arrival process, with rate $\beta > 0$. The population grows stochastically at rate $r > 0$, but it crashes if the carrying capacity κ is exceeded. The survival probability $\alpha \in (0, 1)$ determines the severity of the crash. Over time the population fluctuates between high and low population sizes for several cycles.

This model has been shown to be extra challenging for both synthetic likelihood (SL) methods and ABC methods in Fasiolo et al. (2018). The distribution of the statistics is far from normal which breaks the normality assumption of SL. In addition, the authors show that ABC methods require exceedingly low tolerances and low acceptance rates to achieve satisfying accuracy.

We first run the simulation study using the setup in An et al. (2020). The real data X_0 is generated using parameters $r = 0.4, \kappa = 50, \alpha = 0.09$ and $\beta = 0.05$, and the prior distribution is uniform on $[0, 1] \times [10, 80] \times [0, 1] \times [0, 1]$. The observed data has 250 time-steps, with 50 burn-in steps to remove the transient phase of the process.

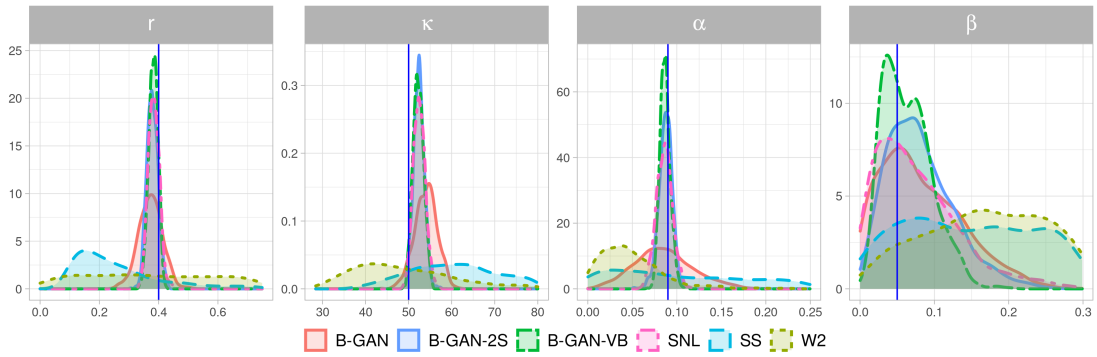


Figure 3.5: Approximate posterior densities under the Boom-and-Bust Model. The true parameter is $\boldsymbol{\theta}_0 = (0.01, 0.5, 1, 0.01)'$.

Previous analyses of the model suggested various summary statistics, including the mean, variance, skewness, kurtosis of the data, lag 1 differences, and lag 1 ratios (An et al., 2020). We use them in SS and SNL methods. We have explored three types of input: the time series itself, the time series in conjunction with their summary statistics, and the summary statistics only. We find that the network built on the summary statistics appears to perform the best, thus we only include the results from this network here.

We include SS, W2, SNL as competitors for comparisons. One snapshot of the approximate posterior densities is provided in Figure 3.5. We report the performance summary averaged over 10 repetitions in Table 3.2. ABC methods struggle to identify the parameters and provide very flat posteriors. The vanilla B-GAN is able to identify the correct location of parameters but with rather wide credible intervals. We observe great improvements after applying the 2-step refinements, the most obvious one for the parameter α. SNL, B-GAN-2S and B-GAN-VB are all able to provide very accurate estimates with tight credible intervals. B-GAN-VB provides the smallest bias and the tightest posteriors, especially for β, although at the cost of lower coverage. While B-GAN-VB is performing better, the performance of B-GAN-2S is slightly inferior relative to SNL in this example. Potentially, the performance of B-GAN-2S can be further improved if we add more sequential refinement steps in B-GAN-2S training, since the prior range here is too wide.

	$r = 0.4$		$\kappa = 50$		$\alpha = 0.09$		$\beta = 0.05$	
(scale)	bias ($\times 10^{-1}$)	CI width ($\times 10^{-1}$)	bias	CI width	bias ($\times 10^{-2}$)	CI width ($\times 10^{-1}$)	bias ($\times 10^{-1}$)	CI width
B-GAN	0.44	1.63	2.92	10.78	3.03	1.38	1.22	0.36 (0.8)
B-GAN-2S	0.27	0.79 (0.8)	1.60	5.29 (0.9)	1.06	0.34	1.05	0.26 (0.7)
B-GAN-VB	**0.23**	**0.65 (0.8)**	**1.29**	**4.88 (0.9)**	**0.89**	**0.25 (0.7)**	**1.00**	**0.19 (0.5)**
SNL	0.24	0.93	1.52	5.37	1.01	0.38	1.28	0.39 (0.9)
SS	2.16	8.26	10.60	37.17	15.08	9.18	4.41	0.95
W2	2.59	9.49	10.16	43.20	5.46	2.77	3.92	0.86 (0.6)

Table 3.2: Summary statistics of the approximated posteriors under the Boom-and-Bust model (averaged over 10 repetitions). Bold fonts mark the best model of each column. The coverage of the 95% credible intervals are 1 unless otherwise noted in the parentheses.

3.6 Discussion

This paper propels strategies for Bayesian simulation using generative networks. We have formalized several schemes for implicit posterior simulation using GAN conditional density regression estimators as well as implicit variational Bayes. The common denominator behind our techniques is (joint) contrastive adversarial learning (Tran et al., 2017; Huszár, 2017). We have provided firm theoretical support in the form of bounds for the typical total variation distance between the posterior and its approximation. We have highlighted the potential of our adversarial samplers on several simulated examples with very encouraging findings. We hope that our paper will embolden practitioners to implement neural network posterior samplers in difficult situations when likelihood (and prior) are implicit.

3.7 Appendix

3.7.1 Proofs from Section 3.4

3.7.1.1 Proof of Theorem 19

Proof. We continue denoting $X^{(n)}$ simply by X. Recall the definition of the KL neighborhood

$$B_n(\theta_0;\epsilon) = \{\theta \in \Theta : \mathrm{KL}(P_{\theta_0}^{(n)} \| P_\theta^{(n)}) \leq n\epsilon^2, V_{2,0}(P_{\theta_0}^{(n)}, P_\theta^{(n)}) \leq n\epsilon^2\}, \tag{3.26}$$

where $\mathrm{KL}(P_{\theta_0}^{(n)} \| P_\theta^{(n)}) = P_{\theta_0}^{(n)} \log[p_{\theta_0}^{(n)}/p_\theta^{(n)}]$ and

$$V_{2,0}(P_{\theta_0}^{(n)}, P_\theta^{(n)}) = P_{\theta_0}^{(n)} \left| \log[p_{\theta_0}^{(n)}/p_\theta^{(n)}] - \mathrm{KL}(P_{\theta_0}^{(n)} \| P_\theta^{(n)}) \right|^2. \tag{3.27}$$

We define an event, for some fixed $C > 0$ and $\epsilon > 0$,

$$\mathcal{A}_n(\epsilon) = \left\{ X : \int_{B_n(\theta_0;\epsilon)} \frac{p_\theta^{(n)}(X)}{p_{\theta_0}^{(n)}(X)} \pi(\theta)\mathrm{d}\theta > \mathrm{e}^{-(1+C)n\varepsilon^2} \Pi[B_n(\theta_0;\epsilon)] \right\}.$$

We denote with $\mathbb{E}$ the expectation with respect to $\{(\theta_j, X_j)\}_{j=1}^T$ from the ABC reference table sampled from the joint $\pi(\theta, X)$. For simplicity of notation, we use $\mathbb{E}_{\widehat{\boldsymbol{\beta}}_T}$ interchangeably with $\mathbb{E}$, since it is equivalently accounting for the randomness in $\widehat{\boldsymbol{\beta}}_T$. Using the fact that the total variation distance is bounded by 2, we have

$$P_{\theta_0}^{(n)} \mathbb{E}_{\hat{\beta}_T} d_{TV}^2 \left(\nu(X_0), \mu\widehat{\boldsymbol{\beta}}_T(X_0)\right) = \int_{\mathcal{A}_n(\epsilon)} p_{\theta_0}^{(n)}(X_0) \mathbb{E}_{\hat{\beta}_T} d_{TV}^2 \left(\nu(X_0), \mu\widehat{\boldsymbol{\beta}}_T(X_0)\right) \mathrm{d}X_0 \tag{3.28}$$

$$+ 4\, \mathbb{P}_{\theta_0}^{(n)}[\mathcal{A}_n^c(\epsilon)]. \tag{3.29}$$

According to (Ghosal and Van Der Vaart, 2007, Lemma 10), we have $\mathbb{P}_{\theta_0}^{(n)}[\mathcal{A}_n^c(\epsilon)] \leq \frac{1}{C^2 n\epsilon^2}$. Denoting with $\pi(X) = \int_\theta p_\theta^{(n)}(X)\pi(\theta)\mathrm{d}\theta$ the marginal likelihood, we can rewrite the term in

(3.28) as

$$\int_{\mathcal{X}} \mathbb{I}_{\mathcal{A}_n(\epsilon)}(X)\pi(X)r(X)\mathbb{E}_{\hat{\beta}_T} d_{TV}^2\left(\nu(X), \mu_{\widehat{\boldsymbol{\beta}}_T}(X)\right) \mathrm{d}X$$

where

$$\frac{1}{r(X)} \equiv \int_{\Theta} \frac{p_\theta^{(n)}(X)}{p_{\theta_0}^{(n)}(X)}\pi(\theta)\mathrm{d}\theta \geq \int_{B_n(\theta_0;\epsilon)} \frac{p_\theta^{(n)}(X)}{p_{\theta_0}^{(n)}(X)}\pi(\theta)\mathrm{d}\theta.$$

On the event $\mathcal{A}_n(\epsilon)$, we can thus write

$$r(X) < \frac{\mathrm{e}^{(1+C)n\epsilon^2}}{\Pi[B_n(\theta_0;\epsilon)]}.$$

Under the assumption (3.25), the term in (3.28) can be upper bounded by

$$\mathrm{e}^{(1+C+C_2)n\epsilon^2} \int_{\mathcal{X}} \pi(X)\mathbb{E}_{\hat{\beta}_T} d_{TV}^2\left(\nu(X), \mu_{\widehat{\boldsymbol{\beta}}_T}(X)\right) \mathrm{d}X \leq \frac{\mathrm{e}^{(1+C+C_2)n\epsilon^2}}{4} \mathbb{E}_{\hat{\beta}_T}[\mathrm{KL}(\nu\|\mu_{\widehat{\boldsymbol{\beta}}_T}) + \mathrm{KL}(\mu_{\widehat{\boldsymbol{\beta}}_T}\|\nu)]. \tag{3.30}$$

The inequality above stems from the Pinsker's inequality (Van Handel, 2014, Theorem 4.8) and the fact that the joint measures ν and $\mu_{\widehat{\boldsymbol{\beta}}_T}$ have the same marginal distribution $\pi(X)$. In particular, using Fubini's theorem, we can write

$$\begin{aligned}
&\int_{\mathcal{X}} \pi(X)\mathbb{E}_{\hat{\beta}_T} d_{TV}^2\left(\nu(X), \mu_{\widehat{\boldsymbol{\beta}}_T}(X)\right) \mathrm{d}X \\
&\quad \leq \frac{1}{4}\int_{\mathcal{X}} \pi(X)\mathbb{E}_{\hat{\beta}_T}[\mathrm{KL}(\nu(X)\|\mu_{\widehat{\boldsymbol{\beta}}_T}(X)) + \mathrm{KL}(\mu_{\widehat{\boldsymbol{\beta}}_T}(X)\|\nu(X))]\mathrm{d}X \\
&\quad = \frac{1}{4}\mathbb{E}_{\hat{\beta}_T}\int_{\mathcal{X}} \pi(X)\int_{\Theta} \log\frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \mid X)}\left[\pi(\theta \mid X) - \pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \mid X)\right]\mathrm{d}\theta\mathrm{d}X \\
&\quad = \frac{1}{4}\mathbb{E}_{\hat{\beta}_T}\left[\mathrm{KL}(\nu\|\mu_{\widehat{\boldsymbol{\beta}}_T}) + \mathrm{KL}(\mu_{\widehat{\boldsymbol{\beta}}_T}\|\nu)\right] \equiv \frac{1}{4}\mathbb{E}_{\hat{\beta}_T} d_{\mathrm{KL}}^S(\nu, \mu_{\widehat{\boldsymbol{\beta}}_T}).
\end{aligned}$$

The above inequality is essential for understanding how the average squared total variation distance between the posterior and its approximation (with the average taken with respect to the observed data generating process) can be related to the 'symmetrized' KL divergence $d_{\mathrm{KL}}^S(\nu, \mu_{\widehat{\boldsymbol{\beta}}_T})$ between the *joint* distribution and its approximation. We now continue to

bound the symmetrized KL divergence. For simplicity, we denote with $\widehat{\boldsymbol{\beta}}$ the estimator $\widehat{\boldsymbol{\beta}}_T$ in (3.22). We have the following decomposition, for any $\boldsymbol{\omega}$ such that $f_{\boldsymbol{\omega}} \in \mathcal{F}$,

$$\begin{aligned} d^S_{\mathrm{KL}}(\nu, \mu_{\widehat{\boldsymbol{\beta}}}) &= \int_{\mathcal{X}} \pi(X) \int_{\Theta} f_\omega(\theta, X)[\pi(\theta \,|\, X) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \,|\, X)]\mathrm{d}\theta\mathrm{d}X \\ &\quad + \int_{\mathcal{X}} \pi(X) \int_{\Theta} \left[\log \frac{\pi(\theta \,|\, X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \,|\, X)} - f_\omega(\theta, X)\right] [\pi(\theta \,|\, X) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \,|\, X)]\mathrm{d}\theta\mathrm{d}X \\ &\leq d_{\mathcal{F}}\left(\nu, \mu_{\widehat{\boldsymbol{\beta}}}\right) + 2 \left\| \log \frac{\pi(\theta \,|\, X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \,|\, X)} - f_\omega(\theta, X) \right\|_\infty , \end{aligned}$$

where we have used the inequality $\int fg \leq \|f\|_\infty \|g\|_1$ and the fact that $\pi(\theta \,|\, X)$ and $\pi_{\widehat{\boldsymbol{\beta}}}(\theta \,|\, X)$ are both non-negative and integrate to one. Then, choosing $f_{\boldsymbol{\omega}} \in \mathcal{F}$ that minimizes the second term we obtain

$$d^S_{\mathrm{KL}}(\nu, \mu_{\widehat{\boldsymbol{\beta}}}) \leq 2\, \mathcal{A}_1(\mathcal{F}, \mathcal{G}, \widehat{\boldsymbol{\beta}}) + d_{\mathcal{F}}\left(\nu, \mu_{\widehat{\boldsymbol{\beta}}}\right),$$

where $\mathcal{A}_1(\mathcal{F}, \mathcal{G}, \widehat{\boldsymbol{\beta}})$ is defined as

$$\mathcal{A}_1(\mathcal{F}, \mathcal{G}, \widehat{\boldsymbol{\beta}}) \equiv \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \log \frac{\pi(\theta \,|\, X)}{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \,|\, X)} - f_\omega(X, \theta) \right\|_\infty$$

and $\mathcal{A}_1(\mathcal{F}, \mathcal{G}) = \mathbb{E}_{\widehat{\boldsymbol{\beta}}} \mathcal{A}_1(\mathcal{F}, \mathcal{G}, \widehat{\boldsymbol{\beta}})$ was defined in (3.23).

We now apply a mild modification of the oracle inequality in (Liang, 2021, Lemma 12). As long as $\mathcal{F}$ and $\mathcal{H}$ are symmetric[10], then for any $\boldsymbol{\beta}$ such that $g_{\boldsymbol{\beta}} \in \mathcal{G}$ we have

$$d_{\mathcal{F}}(\nu, \mu_{\widehat{\boldsymbol{\beta}}}) \leq d_{\mathcal{F}}(\mu_{\boldsymbol{\beta}}, \nu) + 2 d_{\mathcal{F}}(\bar{\nu}_T, \nu) + d_{\mathcal{F}}(\bar{\mu}^{\boldsymbol{\beta}}_T, \mu_{\boldsymbol{\beta}}) + d_{\mathcal{H}}(\bar{\pi}_T, \pi), \tag{3.31}$$

where $\bar{\nu}_T$ and $\bar{\mu}^{\boldsymbol{\beta}}_T$ are the empirical counterparts of $\nu, \mu_{\boldsymbol{\beta}}$ based on T iid samples (ABC reference table $\{(\theta_j, X_j)\}_{j=1}^T$ for ν and $\{(g_{\boldsymbol{\beta}}(Z_j, X_j), X_j)\}_{j=1}^T$ with $Z_j \overset{iid}{\sim} \pi_Z$ for $\mu_{\boldsymbol{\beta}}$). In addition $\bar{\pi}_T$ is the empirical version for the distribution π_Z for Z and $\mathcal{H} = \{h_{\boldsymbol{\omega},\boldsymbol{\beta}} : h_{\boldsymbol{\omega},\boldsymbol{\beta}}(Z, X) =$

10. i.e. if $f \in \mathcal{F}$ then also $-f \in \mathcal{F}$

$f_{\boldsymbol{\omega}}(X, g_{\boldsymbol{\beta}}(Z, X))\}$. This oracle inequality can be proved analogously as (Liang, 2021, Lemma 12) the only difference being that due to the conditional structure of our GANs the function class $\mathcal{H}$ is not entirely a composition of networks $f_{\boldsymbol{\omega}}$ and $g_{\boldsymbol{\beta}}$ but has a certain nested structure. Similarly as in (Liang, 2021) (proof of Theorem 13), we can write for any $\boldsymbol{\beta}$ such that $g_{\boldsymbol{\beta}} \in \mathcal{G}$

$$
\begin{aligned}
d_{\mathcal{F}}(\mu_{\boldsymbol{\beta}}, \nu) &\leq B \times d_{TV}(\mu_{\boldsymbol{\beta}}, \nu) \leq B \sqrt{\frac{1}{4} d^{S}_{\mathrm{KL}}(\mu_{\boldsymbol{\beta}}, \nu)} \\
&\leq \frac{B}{\sqrt{2}} \left[\int_{\mathcal{X}} \left\| \log \frac{\pi_{\boldsymbol{\beta}}(\theta \mid X)}{\pi(\theta \mid X)} \right\|_{\infty} \pi(X) \mathrm{d}X \right]^{1/2}.
\end{aligned}
$$

Choosing $\boldsymbol{\beta}$ that minimizes the expectation on the right side, we obtain $d_{\mathcal{F}}(\mu_{\boldsymbol{\beta}}, \nu) \leq \frac{B}{\sqrt{2}} \mathcal{A}_2(\mathcal{G})$, where the term $A_2(\mathcal{G})$ was defined in (3.24). Denote with

$$
R_T(\mathcal{F}) = E_{\boldsymbol{\varepsilon}} \sup_{f \in \mathcal{F}} \frac{1}{T} \sum_{j=1}^{T} \varepsilon_j f(X_j, \theta_j)
$$

the Rademacher complexity with $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_T)'$ iid Rademacher[11] variables. For the second term in (3.31), the symmetrization property (see e.g. Lemma 26 in Liang (2021) or Theorem 3.17 in Sen (2018)) yields for $T \geq Pdim(\mathcal{F})$

$$
\mathbb{E}\, d_{\mathcal{F}}(\bar{\nu}_T, \nu) \leq 2\, \mathbb{E}\, R_T(\mathcal{F}) \leq \widetilde{C} \times B \sqrt{\frac{Pdim(\mathcal{F}) \log T}{T}}
$$

for some $\widetilde{C} > 0$, where $Pdim(\mathcal{F})$ is the pseudo-dimension of the function class $\mathcal{F}$ (Definition 2 in (Bartlett et al., 2017)). The second inequality follows, for example, from Lemma 29 in (Liang, 2021). The bounds on $\mathbb{E}\, d_{\mathcal{H}}(\bar{\pi}_T, \pi)$, and $\mathbb{E}\, d_{\mathcal{F}}(\bar{\mu}^{\beta}_T, \mu_{\beta})$ in (3.31) are analogous. Putting the pieces together from the oracle inequality in (3.31) we can upper-bound

11. taking values $\{-1, +1\}$ with probability $1/2$.

$P_{\theta_0}^{(n)} \mathbb{E} d_{TV}^2 \left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}_T}(X_0)\right)$ with

$$\frac{1}{C^2 n\varepsilon^2} + \frac{\mathrm{e}^{(1+C+C_2)n\varepsilon^2}}{4} \left[2\mathcal{A}_1(\mathcal{F}, \mathcal{G}) + \frac{B}{\sqrt{2}} \mathcal{A}_2(\mathcal{G}) + 4\widetilde{C}\, B \sqrt{\frac{\log T}{T}} \left(Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})\right)^{1/2} \right]$$

which yields the desired statement. □

3.7.1.2 Proof of Corollary 20

Proof. We continue to use the shorthand notation X for $X^{(n)}$ and $\widehat{\boldsymbol{\beta}}$ for $\widehat{\boldsymbol{\beta}}_T$. Denote with $\widetilde{\pi}(\theta)$ the proposal distribution. Then, the posterior $\widetilde{\pi}(\theta \mid X)$ under $\widetilde{\pi}(\theta)$ satisfies

$$\pi(\theta \mid X) = \widetilde{\pi}(\theta \mid X) \times R(X) \times r(\theta), \quad \text{where} \quad R(X) = \frac{\widetilde{\pi}(X)}{\pi(X)} \text{ and } r(\theta) = \frac{\pi(\theta)}{\widetilde{\pi}(\theta)}. \tag{3.32}$$

Our reconstruction in Algorithm 4 works by first approximating the joint distribution $\widetilde{\pi}(\theta, X)$ and then reweighing by the prior ratio, namely

$$\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X) = \widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X) \times R(X) \times r(\theta), \tag{3.33}$$

where $\widehat{\boldsymbol{\beta}}$ has been learned by B-GAN (Algorithm 3) by matching the joint $\widetilde{\pi}(\theta, X) = \widetilde{\pi}(\theta \mid X)\widetilde{\pi}(X)$ under the prior $\widetilde{\pi}(\theta)$. We denote the joint measure with this density by $\widetilde{\nu}$. Denote with $\mu_{\widehat{\boldsymbol{\beta}}_T}(X)$ the approximating conditional measure with a density (3.33). We can apply the same steps as in the proof of Theorem 19 until the step in (3.30). Similarly, we denote with $\widetilde{\mathbb{E}}$ the expectation with respect to $\{(\theta_j, X_j)\}_{j=1}^T$ from the ABC reference table sampled from the joint $\tilde{\pi}(\theta, X)$, and we use $\widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}_T}$ interchangeably with $\widetilde{\mathbb{E}}$. The next steps will have minor modifications. Notice that

$$\log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)} = \log \frac{\widetilde{\pi}(\theta \mid X)}{\widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)}$$

and thereby

$$
\begin{aligned}
&\int_{\mathcal{X}} \pi(X) \widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}} d_{TV}^2 \left(\nu(X), \mu_{\widehat{\boldsymbol{\beta}}_T}(X)\right) \mathrm{d}X \\
&\quad \leq \frac{1}{4} \int_{\mathcal{X}} \pi(X) \widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}} \Big[\mathrm{KL}\big(\nu(X) \| \mu_{\widehat{\boldsymbol{\beta}}}(X)\big) + \mathrm{KL}\big(\mu_{\widehat{\boldsymbol{\beta}}}(X) \| \nu(X)\big) \Big] \mathrm{d}X \\
&\quad = \frac{1}{4} \widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}} \int_{\mathcal{X}} \pi(X) \int_{\Theta} \log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)} \left[\pi(\theta \mid X) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)\right] \mathrm{d}\theta \mathrm{d}X \\
&\quad = \frac{1}{4} \widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}} \int_{\mathcal{X}} \widetilde{\pi}(X) \int_{\Theta} r(\theta) \log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)} \left[\widetilde{\pi}(\theta \mid X) - \widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)\right] \mathrm{d}\theta \mathrm{d}X \\
&\quad \equiv \frac{1}{4} \widetilde{\mathbb{E}}_{\widehat{\boldsymbol{\beta}}} d_{\mathrm{KL}}^S(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\boldsymbol{\beta}}}).
\end{aligned}
$$

Using similar arguments as in the proof of Theorem 19, we have the following decomposition, for any $\boldsymbol{\omega}$ such that $f_{\boldsymbol{\omega}} \in \mathcal{F}$,

$$
\begin{aligned}
d_{\mathrm{KL}}^S(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\boldsymbol{\beta}}}) &= \int_{\mathcal{X}} \widetilde{\pi}(X) \int_{\Theta} f_{\omega}(\theta, X) [\widetilde{\pi}(\theta \mid X) - \widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)] \mathrm{d}\theta \mathrm{d}X \\
&\quad + \int_{\mathcal{X}} \widetilde{\pi}(X) \int_{\Theta} \left[r(\theta) \log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)} - f_{\omega}(\theta, X) \right] [\widetilde{\pi}(\theta \mid X) - \widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)] \mathrm{d}\theta \mathrm{d}X \\
&\leq d_{\mathcal{F}} \left(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\boldsymbol{\beta}}}\right) + 2 \left\| r(\theta) \log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)} - f_{\omega}(\theta, X) \right\|_{\infty}.
\end{aligned}
$$

The rest of the proof is analogous. The only difference is that $\widehat{\boldsymbol{\beta}}$ now minimizes the empirical version of $d_{\mathcal{F}} \left(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\boldsymbol{\beta}}}\right)$ under the proposal distribution $\widetilde{\pi}(\theta)$.

3.7.1.3 Motivation for the Sequential Refinement

Remark 24 (2step Motivation)**.** *For the proposal distribution $\widetilde{\pi}(\theta)$, using similar arguments as in the proof of Theorem 19, the TV distance of the posterior at X_0 (not averaged over $P_{\theta_0}^{(n)}$) can be upper-bounded by*

$$
4\, d_{TV}^2 \left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}}(X_0)\right) \leq 2\, \mathcal{A}_1(\mathcal{F}, \mathcal{G}, X_0) + \frac{B}{\sqrt{2}} \mathcal{A}_2(\mathcal{G}) + 4\widetilde{C}\, B \sqrt{\frac{\log T \times Pmax}{T}} + A_3(\widetilde{\pi})
$$

where $\mathcal{A}_1(\mathcal{F}, \mathcal{G}, X_0) \equiv \sup_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \log \frac{\pi(\theta \mid X_0)}{\pi_{\boldsymbol{\beta}}(\theta \mid X_0)} - \frac{f_\omega(X_0, \theta)}{r(\theta)} \right]$ *is the discriminability* evaluated at X_0 *(as opposed to* (3.23)*) and where*

$$A_3(\widetilde{\pi}) = 2 \int_{\mathcal{X}} \widetilde{\pi}(X) \left[\| f_{\boldsymbol{\omega}}(X_0, \theta) - f_{\boldsymbol{\omega}}(X, \theta) \|_\infty + B \| g_{\widehat{\boldsymbol{\beta}}}(\theta)(X) - g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0) \|_1 \right] \mathrm{d}X$$

and $g_{\widehat{\boldsymbol{\beta}}}(\theta)(X) \equiv \pi(\theta \mid X) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)$*. This decomposition reveals how the TV distance can be related to discriminability around* X_0 *and an average discrepancy between the true and approximated posterior densities relative to their value at* X_0 *where the average is taken over the marginal* $\widetilde{\pi}(X)$*. These averages will be smaller since the marginal* $\widetilde{\pi}(X)$ *produces datasets more similar to* X_0*. For example, an approximation to the the posterior predictive distribution* $\widetilde{\pi}(X) = \int_{\mathcal{X}} p_\theta^{(n)}(X) \pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)$ *where* $\widehat{\boldsymbol{\beta}}$ *has been learned by B-GAN (Algorithm 3) is likely to yield datasets similar to* X_0*, thereby producing a tighter upper bound than a flat prior.*

We provide clarifications of the calculations and reasoning in Remark 24. We assume that a prior distribution $\widetilde{\pi}(\theta)$ has been used in the ABC reference table that yields the marginal $\widetilde{\pi}(X) = \int_X p_\theta^{(n)}(X) \widetilde{\pi}(\theta) \mathrm{d}\theta$. Recall the definition of the reweighted posterior reconstruction in (3.33) and (3.32). Denote with

$$g_{\widehat{\boldsymbol{\beta}}}(\theta)(X) \equiv \pi(\theta \mid X) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X) = R(X) \times r(\theta) \times [\widetilde{\pi}(\theta \mid X) - \widetilde{\pi}_{\widehat{\boldsymbol{\beta}}}(\theta \mid X)]$$

the difference between true and approximated posteriors at X, where $\widehat{\boldsymbol{\beta}}$ has been trained using the proposal $\widetilde{\pi}(\theta)$ and where $R(X) = \widetilde{\pi}(X)/\pi(X)$ and $r(\theta) = \pi(\theta)/\widetilde{\pi}(\theta)$. Using similar arguments as in the proof of Theorem 19, the squared TV distance of the posterior and its

approximation satisfies, for any element $f_{\boldsymbol{\omega}} \in \mathcal{F} = \{f : \|f\|_\infty \leq B\}$,

$$
\begin{aligned}
4\, d_{TV}^2\left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}}(X_0)\right) \leq & \int_\Theta \log \frac{\pi(\theta \mid X_0)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)} g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0) \mathrm{d}\theta \\
= & \int_\Theta \left[\log \frac{\pi(\theta \mid X_0)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)} - \frac{f_\omega(X_0, \theta)}{r(\theta)}\right] g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0) \mathrm{d}\theta \\
& + \int_{\mathcal{X}} \pi(X) \int_\Theta \frac{f_\omega(X, \theta)}{r(\theta)} g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0) \mathrm{d}\theta \mathrm{d}X \\
& + \int_{\mathcal{X}} \frac{\pi(X)}{r(\theta)} \int_\Theta [f_\omega(X_0, \theta) - f_\omega(X, \theta)] g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0) \mathrm{d}\theta \mathrm{d}X \\
\leq & 2 \inf_{\boldsymbol{\omega}} \left\| \log \frac{\pi(\theta \mid X_0)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)} - \frac{f_\omega(X_0, \theta)}{r(\theta)} \right\|_\infty + d_{\mathcal{F}}\left(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\beta}}\right) \\
& + 2 \int_{\mathcal{X}} \widetilde{\pi}(X) \|f_{\boldsymbol{\omega}}(X_0, \theta) - f_{\boldsymbol{\omega}}(X, \theta)\|_\infty \mathrm{d}X \\
& + 2B \times \int_{\mathcal{X}} \widetilde{\pi}(X) \left\| \frac{g_{\widehat{\boldsymbol{\beta}}}(\theta)(X)}{R(X) r(\theta)} - \frac{g_{\widehat{\boldsymbol{\beta}}}(\theta)(X_0)}{R(X_0) r(\theta)} \right\|_1 \mathrm{d}X.
\end{aligned}
$$

The term $d_{\mathcal{F}}\left(\widetilde{\nu}, \widetilde{\mu}_{\widehat{\boldsymbol{\beta}}}\right)$ can be bounded as in the proof of Corollary 20 by

$$
\frac{B}{\sqrt{2}} \widetilde{\mathcal{A}}_2(\mathcal{G}) + 4\widetilde{C}\, B \sqrt{\frac{\log T \times Pmax}{T}}
$$

Compared to Corollary 20, the upper bound on $d_{TV}^2\left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}}(X_0)\right)$ now involves the discriminability *evaluated at* X_0 (not averaged over the marginal $\widetilde{\pi}(X)$), i.e.

$$
\mathcal{A}_1(\mathcal{F}, \mathcal{G}, X_0) \equiv \sup_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \log \frac{\pi(\theta \mid X_0)}{\pi_{\boldsymbol{\beta}}(\theta \mid X_0)} - \frac{f_\omega(X_0, \theta)}{r(\theta)} \right\|.
$$

The additional two terms in the upper bound involve integration over $\widetilde{\pi}(X)$. □

3.7.1.4 Clarification of Remark 23

With $\varepsilon_n = O(1/\sqrt{n})$ we need to verify that for some suitable choice of $C_n \to \infty$ we have as $n \to \infty$

$$A_1(\mathcal{F}, \mathcal{G}) = o(\mathrm{e}^{-C_n}) \tag{3.34}$$

$$A_2(\mathcal{G}) = o(\mathrm{e}^{-C_n}) \tag{3.35}$$

$$\frac{\log T}{T} \times \big[Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})\big] = o(\mathrm{e}^{-2C_n}) \tag{3.36}$$

for T that is large enough, i.e. $T \geq Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$. The term $\mathcal{A}_2(\mathcal{G})$ equals zero from our assumption of representability of $\pi(\theta, X) = \pi_{g_{\boldsymbol{\beta}_0}}(\theta, X)$ for $g_{\boldsymbol{\beta}_0} \in \mathcal{G}$, which verifies (3.35). We assume that $X^{(n)}$ is a stacked vector of n observed vectors of length q, not necessarily iid, and denote $d^* = d + nq$. Using leaky ReLU networks and assuming representability, for any $\boldsymbol{\beta}$ such that $g_{\boldsymbol{\beta}} \in \mathcal{G}$ the log posterior ratio $r_{\boldsymbol{\beta}}(\theta, X^{(n)}) = \log \frac{\pi(\theta \,|\, X^{(n)})}{\pi_{\boldsymbol{\beta}}(\theta \,|\, X^{(n)})}$ is continuous and, due to boundedness of the network weights, satisfies

$$0 < \underline{C} \leq r_{\boldsymbol{\beta}}(\theta, X^{(n)}) \leq \bar{C} < \infty$$

for any fixed d^*. With large enough T and setting $E = [-\log T, \log T]^{d^*}$ and $R = \log T$, Lemma 50 yields that there exists a ReLU network $f_{\boldsymbol{\omega}} \in \mathcal{F}$ with a width

$$W = 3^{d^*+3} \max\{d^* \lfloor N^{1/d^*} \rfloor, N + 1\}$$

and depth $L = 12 \log T + 14 + 2d^*$ such that

$$\mathcal{A}_1(\mathcal{F}, \mathcal{G}) \leq \sup_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \| r_{\boldsymbol{\beta}}(\theta, X^{(n)}) - f_{\boldsymbol{\omega}}(X^{(n)}, \theta) \|_{L_\infty(E)} \leq 19\sqrt{d^*}\, \omega_f^E(2(\log T)^{1-2/d^*} N^{-2/d^*}),$$

where ω_f^E is the modulus of continuity of $f(t)$ satisfying $\omega_f^E(t) \to 0$ as $t \to 0^+$. Choosing N such that $2^{d^*/2}(\log T)^{d^*/2-1} = o(N)$ as $T \to \infty$, the right-hand side above goes to zero for any fixed $d^* = d + nq$. For each n, we can find T large enough (depending on the modulus of continuity) such that $\mathcal{A}_1(\mathcal{F}, \mathcal{G})\mathrm{e}^{C_n\sqrt{d^*}} \leq \eta_n$ for some $\eta_n = o(1)$, yielding (3.34). The smallest T that satisfies this will be denoted with T_n.

In order to verify (3.36), Theorem 14.1 in Anthony and Bartlett (1999) and Theorem 6 in Bartlett et al. (2017) show that for piecewise linear activation functions (including ReLU and leaky ReLU) there exist constants $c, C > 0$ such that

$$c \times SL\log(S/L) \leq Pdim(\mathcal{F}) \leq C \times SL\log S,$$

where $\mathcal{F}$ is a class of discriminator networks with L layers and S parameters. Since elements in $\mathcal{H}$ can be regarded as sparse larger neural networks with $L + L^*$ layers, $S + S^*$ parameters and piece-wise linear activations, we have

$$Pdim(\mathcal{F}) \vee Pdim(\mathcal{H}) \leq C \times (S + S^*)(L + L^*)\log(S + S^*).$$

Our assumption $T \geq Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$ will thus be satisfied, for instance, when

$$T > C \times (S + S^*)(L + L^*)\log(S + S^*). \tag{3.37}$$

Choosing $N = \lfloor 2^{d^*/2}(\log T)^{d^*/2} \rfloor$, which satisfies the requirement $2^{d^*/2}(\log T)^{d^*/2-1} = o(N)$ as $T \to \infty$, yields

$$W = 3^{d^*+3}\max\{d^*\lfloor N^{1/d^*} \rfloor, N + 1\} = 3^{d^*+3}\lfloor 2^{d^*/2}(\log T)^{d^*/2} + 1 \rfloor \tag{3.38}$$

for a sufficiently large n (and thereby d^*). Recall that in the feed-forward neural networks, the total number of parameters $S = \sum_{l=0}^{L-1}[w_l(w_l + 1)]$ satisfies $S \leq LW(W + 1)$. For any

fixed d^* (and thereby n), assuming $L = 12\log T + 14 + 2d^*$ as before and W as in (3.38), we define $T(d^*)$ as the smallest T that satisfies

$$C \times [LW(W+1) + S^*](L + L^*)\log[LW(W+1) + S^*] \leq \frac{T}{\log T} \times \mathrm{e}^{-2C_n} \times \eta_n$$

for some $\eta_n = o(1)$. Any $T > T(d^*)$ satisfies $T > Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$ and $\mathrm{e}^{2C_n}\frac{\log T}{T}[Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})] \leq \eta_n$. With $T \geq \max\{T_n, T(d^*)\}$, the condition (3.36) is verified.

Lemma 25. *(Zhou et al., 2022, Lemma B5) Let f be a uniformly continuous function defined on $E \subset [-R, R]^d$. For any $L, N \in \mathbb{N}^+$, there exists a ReLU network function $f_{\boldsymbol{\phi}}$ with width $3^{d+3}\max\{d\lfloor N^{1/d}\rfloor, N+1\}$ and depth $12L + 14 + 2d$ such that*

$$\|f - f_{\boldsymbol{\phi}}\|_{L_\infty(E)} \leq 19\sqrt{d}\,\omega_f^E(2RN^{-2/d}L^{-2/d}),$$

where $\omega_f^E(t)$ is the modulus of continuity of $f(t)$ satisfying $\omega_f^E(t) \to 0$ as $t \to 0^+$.

3.7.1.5 Theory for Adversarial Variational Bayes

Theorem 26. *Let $\widehat{\boldsymbol{\beta}}_T$ be as in (3.39) where $\mathcal{F} = \{f : \|f\|_\infty \leq B\}$ for some $B > 0$. Denote with $\mathbb{E}$ the expectation with respect to $\{Z_j\}_{j=1}^T$ in the reference table. Assume that the prior satisfies (3.25). Then for $T \geq Pdim(\mathcal{F} \circ \mathcal{G})$ we have for any $C_n > 0$*

$$P_{\theta_0}^{(n)}\mathbb{E}\, d_{TV}^2\left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}_T}(X_0)\right) \leq \mathcal{D}_n^T(\mathcal{F}, \mathcal{G}, \varepsilon_n, C_n),$$

where

$$D_n^T(\mathcal{F}, \mathcal{G}, \varepsilon_n, C_n) = \frac{\mathcal{A}_3(\mathcal{F}, \mathcal{G})}{2} + \frac{1}{C_n^2 n\varepsilon_n^2} + \frac{1}{2}\widetilde{C}\sqrt{\frac{\log T}{T}Pdim(\mathcal{F} \circ \mathcal{G})} + \frac{\mathrm{e}^{(1+C_2+C_n)n\varepsilon_n^2}}{4}\frac{B}{\sqrt{2}}\mathcal{A}_2(\mathcal{G})$$

for some $\widetilde{C} > 0$ *where* $\mathcal{A}_2(\mathcal{G})$ *was defined in* (3.24) *and where*

$$\mathcal{A}_3(\mathcal{F}, \mathcal{G}) = P_{\theta_0}^{(n)} \mathbb{E} \left\| \log \frac{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \mid X_0)}{\pi(\theta \mid X_0)} - f_{\boldsymbol{\omega}(\widehat{\boldsymbol{\beta}}_T)}(X_0, \theta) \right\|_\infty .$$

Here $\mathbb{E}$ *account for the nested randomness in the estimation process of* $\boldsymbol{\omega}(\widehat{\boldsymbol{\beta}}_T)$ *and* $\hat{\boldsymbol{\beta}}_T$.

Proof. We denote with $\bar{E}$ the expectation with respect to the empirical distribution. Because the class $\mathcal{F}$ is symmetrical (i.e. $f \in \mathcal{F}$ implies $-f \in \mathcal{F}$), the adversarial variational Bayes estimator is defined as

$$\widehat{\boldsymbol{\beta}}_T = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[\bar{E}_{Z \sim \pi_Z} f_{\boldsymbol{\omega}(\boldsymbol{\beta})}\big(X_0, g_{\boldsymbol{\beta}}(Z, X_0)\big) - E_{\theta \sim \pi(\theta \mid X_0)} f_{\boldsymbol{\omega}(\boldsymbol{\beta})}\left(X_0, \theta\right) \right] \tag{3.39}$$

where

$$\boldsymbol{\omega}(\boldsymbol{\beta}) = \arg \max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left[\bar{E}_{Z \sim \pi_Z, X \sim \pi(X)} f_{\boldsymbol{\omega}} \left(X, g_{\boldsymbol{\beta}}(Z, X)\right) - \bar{E}_{(\theta, X) \sim \pi(\theta, X)} f_{\boldsymbol{\omega}}(X, \theta) \right].$$

Note that the (stochastic) gradient descent update for $\boldsymbol{\beta}$, conditioning on the most recent value of $\boldsymbol{\omega}$, *does not* involve the second term $E_{\theta \sim \pi(\theta \mid X_0)} f_{\boldsymbol{\omega}(\boldsymbol{\beta})}\left(\theta, X_0\right)$ in (3.39) because it does not depend on $\boldsymbol{\beta}$. The minimization occurs only over the first term. We obtain theoretical results for $\widehat{\boldsymbol{\beta}}_T$ and note that our Algorithm 5 targets this estimator. In the sequel, we denote $\widehat{\boldsymbol{\beta}}_T$ simply by $\widehat{\boldsymbol{\beta}}$ and use the notation $\pi_{\boldsymbol{\beta}}(\theta, X) = \pi_{\boldsymbol{\beta}}(\theta \mid X)\pi(X)$ for the joint generator model. Using the Pinsker inequality we obtain

$$\begin{aligned}
P_{\theta_0}^{(n)} \mathbb{E} 4\, d_{TV}^2 \left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}}(X_0)\right) \leq & P_{\theta_0}^{(n)} \mathbb{E} \int_\Theta \log \frac{\pi(\theta \mid X_0)}{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)} \left[\pi(\theta \mid X_0) - \pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)\right] \mathrm{d}\theta \\
\leq & P_{\theta_0}^{(n)} \mathbb{E} 2 \left\| \log \frac{\pi_{\widehat{\boldsymbol{\beta}}}(\theta \mid X_0)}{\pi(\theta \mid X_0)} - f_{\boldsymbol{\omega}(\widehat{\boldsymbol{\beta}})}(X_0, \theta) \right\|_\infty \\
& + P_{\theta_0}^{(n)} \mathbb{E} d_{\widehat{\boldsymbol{\beta}}}(\nu_{\widehat{\boldsymbol{\beta}}}(X_0), \mu(X_0)),
\end{aligned}$$

where we define (for any $\boldsymbol{\beta}, \widetilde{\boldsymbol{\beta}}$ such that $g_{\boldsymbol{\beta}} \in \mathcal{G}$ and $g_{\widetilde{\boldsymbol{\beta}}} \in \mathcal{G}$)

$$d_{\widetilde{\boldsymbol{\beta}}}(\nu_{\boldsymbol{\beta}}(X), \mu(X)) \equiv E_{\theta \sim \pi_{\boldsymbol{\beta}}(\theta \,|\, X)} f_{\boldsymbol{\omega}(\widetilde{\boldsymbol{\beta}})}(X, \theta) - E_{\theta \sim \pi(\theta \,|\, X)} f_{\boldsymbol{\omega}(\widetilde{\boldsymbol{\beta}})}(X, \theta).$$

From the definition of (3.39) and since $\mathcal{F} \circ \mathcal{G}$ is symmetrical, we have for any realization X_0 and for any $\boldsymbol{\beta}$

$$\begin{aligned} d_{\widehat{\boldsymbol{\beta}}}(\nu_{\widehat{\boldsymbol{\beta}}}(X_0), \mu(X_0)) &= d_{\widehat{\boldsymbol{\beta}}}(\nu_{\widehat{\boldsymbol{\beta}}}(X_0), \bar{\nu}_{\widehat{\boldsymbol{\beta}}}(X_0)) + d_{\widehat{\boldsymbol{\beta}}}(\bar{\nu}_{\widehat{\boldsymbol{\beta}}}(X_0), \mu(X_0)) \\ &\leq d_{\widehat{\boldsymbol{\beta}}}(\nu_{\widehat{\boldsymbol{\beta}}}(X_0), \bar{\nu}_{\widehat{\boldsymbol{\beta}}}(X_0)) + d_{\boldsymbol{\beta}}(\bar{\nu}_{\boldsymbol{\beta}}(X_0), \mu(X_0)) \\ &\leq d_{\mathcal{F}}(\nu_{\widehat{\boldsymbol{\beta}}}(X_0), \bar{\nu}_{\widehat{\boldsymbol{\beta}}}(X_0)) + d_{\boldsymbol{\beta}}(\bar{\nu}_{\boldsymbol{\beta}}(X_0), \nu_{\boldsymbol{\beta}}(X_0)) + d_{\boldsymbol{\beta}}(\nu_{\boldsymbol{\beta}}(X_0), \mu(X_0)) \\ &\leq 2 d_{\mathcal{F} \circ \mathcal{G}}(\pi_Z, \bar{\pi}_Z) + d_{\boldsymbol{\beta}}(\nu_{\boldsymbol{\beta}}(X_0), \mu(X_0)). \end{aligned} \tag{3.40}$$

Next, using the same arguments as in the proof of Theorem 19, we obtain

$$P_{\theta_0}^{(n)} \mathbb{E} d_{\boldsymbol{\beta}}(\nu_{\boldsymbol{\beta}}(X_0), \mu(X_0)) \leq 4 \, \mathbb{P}_{\theta_0}^{(n)}[\mathcal{A}_n^c(\epsilon)] + \mathrm{e}^{(1 + C_n + C_2) n \epsilon^2} \mathbb{E} \int_{\mathcal{X}} d_{\boldsymbol{\beta}}(\nu_{\boldsymbol{\beta}}(X), \mu(X)) \pi(X) \mathrm{d}X$$

Since $\| f_{\boldsymbol{\omega}(\boldsymbol{\beta})} \|_\infty \leq B$, we have for any $\boldsymbol{\beta}$

$$\begin{aligned} E_X d_{\boldsymbol{\beta}}(\mu_{\boldsymbol{\beta}}(X), \nu(X)) &= \int_{\mathcal{X}} \pi(X) \int_\Theta f_{\boldsymbol{\omega}(\boldsymbol{\beta})}(X, \theta) \big[\pi(\theta \,|\, X) - \pi_{\boldsymbol{\beta}}(\theta \,|\, X) \big] \mathrm{d}\theta \mathrm{d}X \\ &\leq B \times E_X d_{TV}(\mu_{\boldsymbol{\beta}}(X), \nu(X)) \\ &\leq \frac{B}{\sqrt{2}} E_X \sqrt{\left\| \log \frac{\pi(\theta \,|\, X)}{\pi_{\boldsymbol{\beta}}(\theta \,|\, X)} \right\|_\infty}. \end{aligned}$$

In the sequel, we choose $\boldsymbol{\beta}$ which minimizes this term. Next, using the symmetrization techniques as before in the proof of Theorem 19 and denoting with $\mathbb{E}$ the expectation with

respect to $\{Z_j\}_{j=1}^T$, we have

$$d_{\mathcal{F}\circ\mathcal{G}}(\pi_Z, \bar{\pi}_Z) \leq \mathbb{E}\mathcal{R}_n(\mathcal{F}\circ\mathcal{G}) \leq \widetilde{C}\sqrt{\frac{\log T}{T} Pdim(\mathcal{F}\circ\mathcal{G})}.$$

Putting the pieces together, we obtain an upper bound for $P_{\theta_0}^{(n)} \mathbb{E}\, d_{TV}^2 \left(\nu(X_0), \mu_{\widehat{\boldsymbol{\beta}}_T}(X_0)\right)$.

CHAPTER 4

UNCERTAINTY QUANTIFICATION FOR SPARSE DEEP LEARNING

Deep learning methods continue to have a decided impact on machine learning, both in theory and in practice. Statistical theoretical developments have been mostly concerned with approximability or rates of estimation when recovering infinite dimensional objects (curves or densities). Despite the impressive array of available theoretical results, the literature has been largely silent about *uncertainty quantification* for deep learning. This paper takes a step forward in this important direction by taking a Bayesian point of view. We study Gaussian approximability of certain aspects of posterior distributions of sparse deep ReLU architectures in non-parametric regression. Building on tools from Bayesian non-parametrics, we provide semi-parametric Bernstein-von Mises theorems for linear and quadratic functionals, which guarantee that implied Bayesian credible regions have valid frequentist coverage. Our results provide new theoretical justifications for (Bayesian) deep learning with ReLU activation functions, highlighting their *inferential potential*.

4.1 Introduction

Neural networks have emerged as one of the most powerful prediction systems. Their empirical success has been amply documented in many applications including image classification (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012a) or game intelligence (Silver et al., 2016). Beyond algorithmic developments, there has been a rapid progress in theoretical understanding of deep learning (Anthony and Bartlett, 2009). The majority of

. Adopted from Yuexi Wang and Veronika Ročková. Uncertainty quantification for sparse deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 298–308. PMLR, 2020.

existing *statistical* theory has been concerned with *prediction* aspects, e.g. approximability (Telgarsky, 2016; Yarotsky, 2017; Vitushkin, 1964) or rates of convergence (either from a frequentist point of view (Mhaskar et al., 2017; Poggio et al., 2017; Schmidt-Hieber, 2020) or a Bayesian point of view (Polson and Ročková, 2018)). A distinguishing feature of statistics, that goes beyond mere construction of prediction maps, is providing uncertainty quantification (UQ) for inference (hypothesis testing and confidence assessments). The statistical approach to uncertainty quantification uses observations to construct a random subset (confidence set) which contains the truth with large probability. While computational methods such as Boostrapped DQN (Osband et al., 2016) and Deep Ensembles (Lakshminarayanan et al., 2017) have been proposed to quantify predictive uncertainty, theoretically justifiable developments on UQ for deep learning are more rare.

A structured approach to the problem of uncertainty assessment lies in Bayesian hierarchical modeling. The Bayesian paradigm for deep learning places a probabilistic blanket over architectures/parameters and allows for uncertainty quantification via posterior distributions (Neal, 1993). While exact Bayesian inference is computationally intractable, many approximate methods have been developed including MCMC (Neal, 2012), Variational Bayes (Ullrich et al., 2017), Bayes by Backprop (Blundell et al., 2015), Scalable Data Augmentation (Wang et al., 2022b), Monte Carlo Dropout (Gal and Ghahramani, 2016), Hamiltonian methods (Springenberg et al., 2016). The Bayesian inference is fundamentally justified by the Bernstein-von Mises (BvM) theorem. The BvM phenomenon occurs when, as the number of observations increases, the posterior distribution is approximately Gaussian, centered at an efficient estimator of the parameter of interest. Moreover, the posterior credible sets, i.e. regions with prescribed posterior probability, are then also confidence regions with the same asymptotic coverage. While the BvM limit is not unexpected in regular parametric models, infinite-dimensional notions of BvM are far from obvious (see e.g. Castillo and Nickl (2013)).

Our paper deals with uncertainty quantification. Our approach is inherently Bayesian and, as such, is conceptually epistemic where uncertainty about the unknown state of nature is expressed through priors and coherently updated with the data. The frequentist notion of uncertainty is primarily aleatoric as it reflects variability in possible realizations of an event that is largely stochastic in nature and is irreducible. The premise of the BvM phenomenon is that these two uncertainties, while qualitatively very different, are not mutually exclusive in the sense that their quantifications can agree. Priors that are not subjective and more automatic do not necessarily adhere to epistemic interpretation and can yield aleatoric measures of quantification. Our work sheds light on the fact that frequentist calibration is an attainable goal of Bayesian statistical procedures, where the BvM phenomenon facilitates communication of uncertainty using the more universally understood frequentist concept (Dawid, 1982).

In this note, we study the *semi-parametric* BvM phenomenon concerning the limiting behavior of the posterior distribution of certain low-dimensional summaries of a regression function. In particular, we assume a non-parametric regression model with fixed covariates and sparse deep ReLU network priors, which have been recently shown to attain the optimal speed of posterior contraction (Polson and Ročková, 2018). Building on Castillo and Rousseau (2015), who laid down the general framework for semi-parametric BvMs, and on Polson and Ročková (2018), we formulate asymptotic normality for linear and quadratic functionals. Related semi-parametric BvM results have been established for density estimation (Rivoirard and Rousseau, 2012), Gaussian process priors (Castillo, 2012a,b), covariance matrix (Gao and Zhou, 2016) and tree/forest priors (Ročková, 2020). Our results provide new frequentist theoretical justifications for Bayesian deep learning inference with certain aspects of a regression function.

Our analysis focuses on sparse deep ReLU networks. Deep networks have been shown to outperform shallow ones in terms of representation power (Telgarsky, 2016), model com-

plexity (Mhaskar et al., 2017) and generalization (Kawaguchi et al., 2017). The ReLU squashing function has been generally preferred due to its expressibility and inherent sparsity. For instance, Yarotsky (2017) provides error bounds for approximating polynomials and smooth functions with deep ReLU networks. Schmidt-Hieber (2020) showed that deep sparse ReLU networks can yield rate-optimal reconstructions of smooth functions and their compositions. Sparse architectures (in addition to ReLU) can reduce the test error. For example, sparsification can be achieved with dropout (Srivastava et al., 2014) which averages over sparse structures by randomly removing nodes and, thereby, alleviates overfitting. More recently, Polson and Ročková (2018) proposed Spike-and-Slab Deep Learning (SS-DL) as a fully Bayesian variant of dropout. Their framework provably does not overfit and achieves an *adaptive* near-minimax-rate optimal posterior concentration. Liu (2021) studies the BvM phenomena for the gradient function of Bayesian deep ReLU network and proposes a variable selection method based on the credible intervals. We continue the theoretical investigation of SS-DL in this paper.

Similar to Ročková (2020), we consider a non-parametric regression model where responses $\boldsymbol{Y}^{(n)} = (Y_1, \ldots, Y_n)'$ are linked to fixed covariates $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})' \in [0,1]^p$ for $i = 1, \ldots, n$ as follows

$$Y_i = f_0(\boldsymbol{x}_i) + \epsilon_i,\ \epsilon_i \overset{iid}{\sim} N(0,1), \tag{4.1}$$

where $f_0 \in \mathcal{H}_p^\alpha$ is an α-Hölder smooth function on a unit cube $[0,1]^p$ for some $\alpha > 0$. The true generative model implied by (4.1) will be denoted by $\mathbb{P}_0^n$. We want to reconstruct f_0 with $f \in \mathcal{F}$, where the model class $\mathcal{F}$ is assigned a prior distribution Π. Our goal is to study the asymptotic behavior of the posterior distribution

$$\Pi\left[\sqrt{n}(\Psi(f) - \hat{\Psi}) \,|\, \boldsymbol{Y}^{(n)}\right],$$

where $\Psi : \mathcal{F} \to \mathbb{R}$ is a measurable function of interest and where $\hat{\Psi}$ is a random centering

point (see Theorem 2.1 in Castillo and Rousseau (2015)).

Two functionals are considered in our work. The first one is the linear functional

$$\Psi(f) = \frac{1}{n}\sum_{i=1}^{n} a(\boldsymbol{x}_i) f(\boldsymbol{x}_i), \tag{4.2}$$

with a constant weighting functions $a(\cdot)$. We discuss potential generalizations to the non-constant case later in Section 5.6 . The second functional of interest is the squared-L^2 norm

$$\Psi(f) = \|f\|_L^2\,, \tag{4.3}$$

where $\|f\|_L^2 = \frac{1}{n}\sum_{i=1}^{n}[f(\boldsymbol{x}_i)]^2$. Note that $\|\cdot\|_L$ corresponds to the LAN (locally asymptotically normal) norm, which is equivalent to the empirical L^2-norm $\|\cdot\|_n$ in our model. There is extensive literature on minimax estimation of linear and quadratic functionals, initiated in Ibragimov and Khasminskii (1985) and followed by Cai and Low (2005); Efromovich and Low (1996); Collier et al. (2017), to name a few. While the linear functional is useful for inference about the average regression surface, the quadratic functional is useful in many testing problems, including construction of confidence balls (Cai and Low, 2006a) and goodness of fit tests (Dümbgen, 1998; Butucea, 2007). We study adaptive estimation of the two functionals from a Bayesian perspective.

First, we give the definition of asymptotic normality.

Definition 27. *Denote with β the bounded Lipschitz metric for weak convergence and with τ_n the mapping $\tau_n : f \to \sqrt{n}(\Psi(f) - \Psi_n)$. We say that the posterior distribution of the functional $\Psi(f)$ is asymptotically normal with centering Ψ_n and variance V if*

$$\beta(\Pi[\cdot \mid \boldsymbol{Y}^{(n)}] \circ \tau_n^{-1}, \mathcal{N}(0, V)) \to 0, \tag{4.4}$$

in $\mathbb{P}_0^n$-probability as $n \to \infty$. We will write this more compactly as $\Pi[\cdot \mid \boldsymbol{Y}^{(n)}] \circ \tau_n^{-1} \rightsquigarrow$

$\mathcal{N}(0, V)$.

Next, we say that the posterior distribution *satisfies the BvM theorem* if (4.4) holds with $\Psi_n = \hat{\Psi} + o_P(\frac{1}{\sqrt{n}})$ for $\hat{\Psi}$ a linear efficient estimator of $\Psi(f_0)$.

Castillo and Rousseau (2015) provide general conditions on the model and on the function $\Psi(\cdot)$ to guarantee that the BvM phenomenon holds. Our results are built on the first-order approximation technique developed in their work. Essentially, we want to show that the sparse deep learning posterior can approximate both f_0 and the linear expansion term well enough so that the remainder term vanishes when $n \to \infty$.

The rest of our paper is organized as follows. Section 4.2 defines sparse ReLU networks and reviews the posterior concentration results. Section 4.3 contains the main results of BvM properties of the two functionals and Section 4.4 discusses extensions to adaptive priors. Section 4.5 concludes with a discussion.

4.2 Deep ReLU Networks

We follow the notation used in Polson and Ročková (2018). We denote with $\mathcal{F}(L, \mathbf{p}, s)$ the class of sparse ReLU networks with $L \in \mathbb{N}$ layers, a vector of $\mathbf{p} = (p_0, \ldots, p_{L+1})' \in \mathbb{N}^{L+2}$ hidden units and sparsity level $s \in \mathbb{N}$, which is the upper bound on the number of nonzero parameters. In our model, we have $p_0 = p$ and $p_{L+1} = 1$. Each function $f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) \in \mathcal{F}(L, \mathbf{p}, s)$ takes the form

$$f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) = W_{L+1}\sigma_{b_L}\Big(W_L\sigma_{b_{L-1}}\cdots\sigma_{b_1}(W_1\boldsymbol{x})\Big) + b_{L+1} \tag{4.5}$$

where $b_l \in \mathbb{R}^{p_l}$ are shift vectors and W_l are $p_l \times p_{l-1}$ weight matrices that link neurons between the $(l-1)^{th}$ and l^{th} layers and $\sigma_b(\boldsymbol{x})$ is the squashing function. Throughout this work, we assume the *rectified linear (ReLU)* function $\sigma_b(\boldsymbol{x}) = \max(\boldsymbol{x} + b, 0)$ which applies to vectors elementwise. Note that the top layer shift parameter b_{L+1} is *outside* the ReLU

function since the top layer is only a linear function. We denote the sets of all model parameters with

$$\boldsymbol{B} = \{(W_1, b_1), \ldots, (W_L, b_L), (W_{L+1}, b_{L+1})\}. \tag{4.6}$$

Let $Z_l \in \mathbb{R}^{p_l}$ represent the hidden nodes of the l^{th} layer obtained as

$$\begin{aligned} Z_l(\boldsymbol{x}) &= \sigma_{b_l}(W_l Z_{l-1}(\boldsymbol{x})), \quad \text{for} \quad l = 1 \ldots, L, \\ Z_0(\boldsymbol{x}) &= \boldsymbol{x}. \end{aligned}$$

We use $Z = \{Z_l\}_{l=1}^{L}$ to represent the collection of all hidden neurons. Their values are completely determined by $\{W_l, b_l\}_{l=1}^{L}$, independently of the top layer parameters $\{W_{L+1}, b_{L+1}\}$.

4.2.1 Spike-and-Slab Priors

We place a probabilistic structure on $\boldsymbol{B}$ that is slightly different from Polson and Ročková (2018). In particular, we remove the spike-and-slab prior on the top layer L to obtain a fully-connected top layer for each function $f_{\boldsymbol{B}}^{DL}(x)$. Such a relaxation on the top layer facilitates the *change of measure* step in our results. Later we show that having a fully connected top layer *does not* affect the network approximability and the posterior concentration rate.

We convert $\boldsymbol{B}$ into a vector by stacking $\{W_l, b_l\}_{l=1}^{L+1}$ from the bottom to the top and denote $\boldsymbol{B} = (\beta_1, \ldots, \beta_T)'$, where $T = \sum_{l=0}^{L} p_{l+1}(p_l + 1)$ is the number of parameters in a fully connected network with L layers and a vector of $\mathbf{p}$ neurons. Note that $\{\beta_j\}_{j>T-(p_L+1)}$ corresponds to the top layer $\{W_{L+1}, b_{L+1}\}$. Then the priors on $\boldsymbol{B}$ are

$$\pi(\beta_j \mid \gamma_j) = \gamma_j \tilde{\pi}(\beta_j) + (1 - \gamma_j)\delta_0(\beta_j), \tag{4.7}$$

with

$$\gamma_j = 1 \quad \text{for} \quad j > T - (p_L + 1), \tag{4.8}$$

where $\tilde{\pi}(\beta)$ is specified as

$$\tilde{\pi}(\beta_j) = \begin{cases} N(0,1), & j > T - p_L + 1, \\ \text{Uniform}[-1,1], & j \leq T - p_L + 1, \end{cases} \tag{4.9}$$

i.e., the top layer weights follow standard normal distribution, while the deep weights follow uniform distribution on $[-1,1]$. $\delta_0(\beta)$ is a dirac spike at zero, and $\gamma_j \in \{0,1\}$ for whether or not β_j is nonzero. We let $\gamma_j = 1$ for all $j > T - (p_L + 1)$ so that the top layer is fully connected. The vector $\gamma = (\gamma_1, \ldots, \gamma_T)'$ encodes the connectivity pattern below the top layer. We assume that, given the network structure and the sparsity level $s = |\gamma| > p_L$, all architectures are equally likely a priori, i.e.

$$\pi(\gamma \mid \mathbf{p}, s) = \frac{\mathbb{I}(\gamma_j = 1 \text{ for } j > T - p_L - 1)}{\binom{T-p_L-1}{s-p_L-1}}. \tag{4.10}$$

We denote with $\mathcal{V}^{\mathbf{p},s}$ the set of all combinatorial possibilities of connectivity patterns below the top layer. For a given sparsity level s, we can write

$$\mathcal{F}(L, \mathbf{p}, s) = \bigcup_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \mathcal{F}(L, \mathbf{p}, \gamma), \tag{4.11}$$

where each shell $\mathcal{F}(L, \mathbf{p}, \gamma)$ consists of all uniformly bounded functions $f_{\boldsymbol{B}}^{DL}$ with the same connectivity pattern γ, i.e. $\mathcal{F}(L, \mathbf{p}, \gamma) = \{f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) \in \mathcal{F}(L, \mathbf{p}, s) : f_{\boldsymbol{B}}^{DL}(\boldsymbol{x})$ as in (5.2) with $\boldsymbol{B}$ arising from (4.7) for a given $\gamma \in \mathcal{V}^{\mathbf{p},s}$and where $\left\| f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) \right\|_\infty < F\}$ for some $F > 0$.

Remark 28. *The prior for the deep coefficients β_j in (4.9) can be replaced by*

$$\tilde{\pi}(\beta_j) = N(0,1), \forall j = 1, \ldots, T. \tag{4.12}$$

The posterior concentration rate can be also shown to be rate-optimal under this prior. The

sketch of the proof is given after Theorem 7.1 in Wang and Ročková (2020). Moreover, the BvM property for this prior can be immediately concluded from our proofs of Theorems 29, 31 and 32.

4.2.2 A Connection between Deep ReLUs and Trees

Before proceeding, it will be useful to revisit a connection between networks and trees. Recall that any deep ReLU network function can be written as a sum of local linear functions, i.e.

$$f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) = \sum_{k=1}^{K} \mathbb{I}(\boldsymbol{x} \in \Omega_k)(\tilde{\beta}_k' \boldsymbol{x} + \tilde{\alpha}_k), \tag{4.13}$$

where $\{\Omega_k\}_{k=1}^K$ is a partition of the predictor space made by recursive ReLU layers (see Polson and Sokolov (2017) for illustrations). Both the partition $\{\Omega_k\}_{k=1}^K$ and the coefficients of the local linear functions $\{\tilde{\beta}_k, \tilde{\alpha}_k\}_{k=1}^K$ are determined from $\{W_l, b_l\}_{l=1}^{L+1}$. We have omitted the dependence on $\boldsymbol{B}$ for simplicity of notation.

Balestriero and Baraniuk (2018) view ReLU as Max-Affine Spline Functions (MASO) and describe how the local linear functions and partitions are determined from weights $\boldsymbol{B}$. They point out that the partition by layer l contains up to 2^{p_l} convex conjoint regions. In practice, however, many of them could be empty intersections. Montufar et al. (2014) shows that the number of linear regions K of ReLU networks is upper-bounded by 2^T and lower-bounded by $(\prod_{l=1}^{L-1} \lfloor \frac{p_l}{p} \rfloor^p) \sum_{j=1}^{p} \binom{p_L}{j}$. Hanin and Rolnick (2019) further measure the volume of the boundaries between these regions.

Deep ReLU networks are similar to trees/forests methods in the sense that they also partition the predictor space. In fact, any regression tree can be represented by a neural network with a particular activation function, as we illustrate below using an example from Biau et al. (2016).

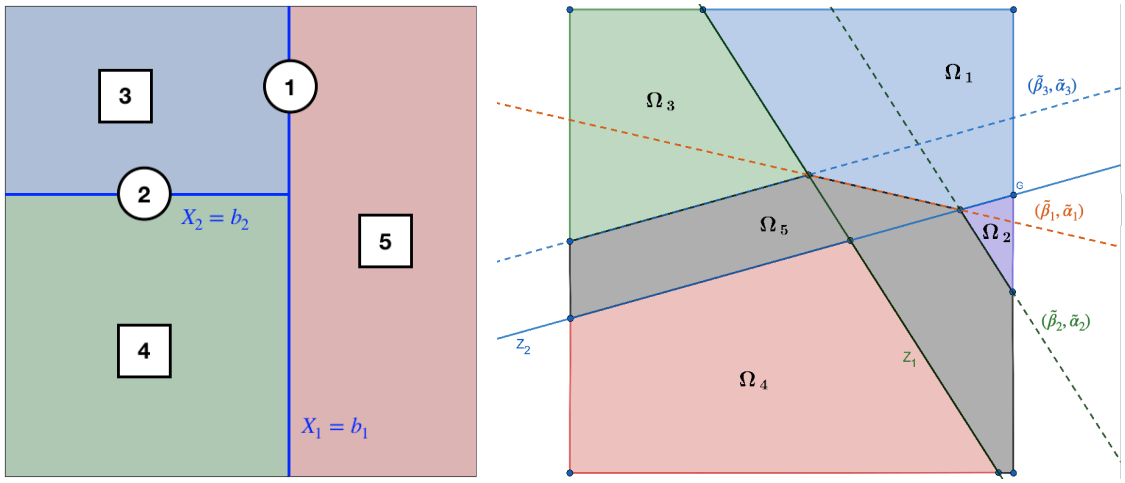
4.2.2.0.1 Example 1 Define an activation function $\tau_b : \mathbb{R} \to \{-1, 1\}$ such that

$$\tau_b(x) = 2\mathbb{I}_{x+b\geq 0} - 1.$$

We can reconstruct a two-dimensional ($p = 2$) example in Figure 4.1(a) with a neural network as

$$Z_1 = \tau_{-b_1}(X_1), \qquad Z_2 = \tau_{-b_2}(X_2), \qquad Z_3 = \tau_{-2}(-Z_1 + Z_2),$$
$$Z_4 = \tau_{-2}(Z_1 + Z_2), \qquad Z_5 = \tau_{-1}(Z_1), \qquad f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) = \sum_{i=3}^{5} W_i Z_i.$$

where b_1 and b_2 set the decision boundaries along (X_1, X_2) axes in the tree, and $\{W_i\}_{i=3}^5$ are the jump sizes in each leaf node. A more detailed explanation of the choice of weights can be found in Biau et al. (2016). By analogy, the hierarchical segmentation is determined by the deep layers while the values of the leaf nodes are assigned by the top layer.



(a) Example 1 (b) Example 2

Figure 4.1: Visualization of the two examples

Deep ReLU networks use a different activation function and thereby place fewer restrictions on the geometry of the partition boundaries (shards as opposed to boxes). There are

two aspects that make the analysis of deep ReLU networks more difficult. First, the partitioning lines do not align with coordinate axes when $W_l \neq 0$. Second, the partitioning cells $\{\Omega_k\}_{k=1}^K$ and the local linear coefficients $\{\tilde{\beta}_k, \tilde{\alpha}_k\}_{k=1}^K$ are related as they both depend on the unknown coefficients $\{W_l, b_l\}_{l=1}^L$. In tree models, on the other hand, they are independent parameters.

To illustrate the correspondence between the partitions and local linear functions as well as their relationship to $\boldsymbol{B}$, we consider the following toy example.

4.2.2.0.2 Example 2 Consider $L = 1, p = 2$ and $p_1 = 2$. Given the weights and shifts as

$$W_1 = \begin{pmatrix} W_1^1 \\ W_2^1 \end{pmatrix}, b_1 = (b_1^1, b_2^1), W_2 = \begin{pmatrix} W_1^2 \\ W_2^2 \end{pmatrix}, b_2 = b^2,$$

we can write the model as

$$Z_1 = \sigma_{b_1^1}(W_1^1 \boldsymbol{x}), \quad Z_2 = \sigma_{b_2^1}(W_2^1 \boldsymbol{x}), \quad f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) = \sigma_{b^2}(W_1^2 Z_1 + W_2^2 Z_2).$$

Then the corresponding $\{\tilde{\beta}_k, \tilde{\alpha}_k, \Omega_k\}_{k=1}^5$ for each local linear function can be organized as

i	$\tilde{\beta}_i$	$\tilde{\alpha}_i$	Ω_i
1	$W_1^2 W_1^1 + W_2^2 W_2^1$	$W_1^2 b_1^1 + W_2^2 b_2^1 + b^2$	$A_1 \cap A_2 \cap A_3$
2	$W_1^2 W_1^1$	$W_1^2 b_1^1 + b^2$	$A_1 \cap A_2^c \cap A_4$
3	$W_2^2 W_2^1$	$W_2^2 b_2^1 + b^2$	$A_1^c \cap A_2 \cap A_5$
4	0	$\max(b^2, 0)$	$A_1^c \cap A_2^c$
5	0	0	$(\Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \Omega_4)^c$

with

$$A_1 = \{\boldsymbol{x} : W_1^1 \boldsymbol{x} + b_1^1 > 0\}, \quad A_2 = \{\boldsymbol{x} : W_2^1 \boldsymbol{x} + b_2^1 > 0\}, \quad A_3 = \{\boldsymbol{x} : \tilde{\beta}_1 \boldsymbol{x} + \tilde{\alpha}_1 > 0\},$$
$$A_4 = \{\boldsymbol{x} : \tilde{\beta}_2 \boldsymbol{x} + \tilde{\alpha}_2 > 0\}, \quad A_5 = \{\boldsymbol{x} : \tilde{\beta}_3 \boldsymbol{x} + \tilde{\alpha}_3 > 0\}.$$

Here we use A_i^c to denote the complement of set A_i, i.e., $A_i^c = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \notin A_i\}$. The covariance matrix of $\{\tilde{\beta}_k\}_{k=1}^3$ is

$$\text{Var}\begin{pmatrix} \tilde{\beta}_1 \\ \tilde{\beta}_2 \\ \tilde{\beta}_3 \end{pmatrix} = \frac{2}{9}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

This example is plotted in Figure 4.1(b), where the boundaries of the partitions are nested according to $\{\tilde{\beta}_k, \tilde{\alpha}_k\}_{k=1}^5$ and determined by $\{W_l, b_l\}_{l=1}^2$.

4.2.3 Posterior Concentration

One essential prerequisite for our BvM analysis is optimal rate of posterior convergence. Polson and Ročková (2018) (PR18) showed that sparse deep ReLUs attain the near-minimax optimal rate and are *adaptive* to unknown smoothness under suitable priors on the architecture size.

Here, we use a modified prior with a fully connected top linear layer (as given by (4.8)). The posterior concentration result still holds. Indeed, for an arbitrary sparse network, there exists at least one network with a fully connected linear layer that achieves the same approximation error. The approximability of our class of networks is thus the same as the class considered in PR18. We illustrate how such a network can be constructed in Wang and Ročková (2020, Lemma 7.1).

Denoting (L^*, N^*, s^*) as in Theorem 5.1 of PR18 and choosing the parameters of the network as

$$\begin{cases} L = L^* + 1 \asymp \log(n), \\ s = s^* + 24pN^* \lesssim n^{p/(2\alpha+p)}, \end{cases} \tag{4.14}$$

we define

$$A_n^M = \{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}(L, \mathbf{p}, s) : \left\| f_{\boldsymbol{B}}^{DL} - f_0 \right\|_L \leq M\xi_n\} \tag{4.15}$$

with $\xi_n = n^{-\alpha/(2\alpha+p)} \log^{\delta}(n)$ for some $M > 0$ and $\delta > 0$. As we formalize in Lemma 35, one can show $\Pi[A_n^{M_n} | \boldsymbol{Y}^{(n)}] = 1 + o_P(1)$ for any $M_n \to \infty$ and uniformly bounded α-Hölder mappings f_0.

Our analyses in Section 4.3 will be performed locally on sets $A_n^{M_n}$ where the posterior concentrates.

4.3 Semi-parametric BvM's

Locally on the sets $A_n \equiv A_n^{M_n}$ we will perform expansions of the log-likelihood as well as the functional Ψ. The log-likelihood is denoted with

$$\ell_n(f) = -\frac{n}{2} \log 2\pi - \sum_{i=1}^{n} \frac{[Y_i - f(\boldsymbol{x}_i)]^2}{2}.$$

and the log-likelihood ratio $\Delta_\ell(f) = \ell(f) - \ell(f_0)$ can be expressed as a sum of a quadratic term and a stochastic term via the LAN expansion as follows

$$\Delta_\ell(f) = -\frac{n}{2} \|f - f_0\|_L^2 + \sqrt{n} W_n(f - f_0)$$

where

$$\begin{aligned} W_n(f - f_0) &= \langle f - f_0, \sqrt{n}\boldsymbol{\epsilon} \rangle_L \\ &= \frac{1}{n} \sum_{i=1}^{n} \sqrt{n} \epsilon_i [f_0(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)]. \end{aligned}$$

We focus on the first-order approximations of the functionals. For any $f \in A_n$, we write

$$\Psi(f) = \Psi(f_0) + \langle \Psi_0^{(1)}, f - f_0 \rangle_L + r(f, f_0).$$

The first-order term $\Psi_0^{(1)}$ is equal to a for linear functionals (4.2) and $2f_0$ for the quadratic functional (4.3). The inner product $\langle \cdot, \cdot \rangle_L$ is defined as $\langle g, h \rangle_L = \frac{1}{n} \sum_{i=1}^n g(\boldsymbol{x}_i) h(\boldsymbol{x}_i)$ for two functions g and h.

Before we dive into the main development, we recall the results in Castillo and Rousseau (2015) which will be leveraged in our analysis.

There are two sufficient conditions for obtaining weak asymptotic normality as defined in (4.4). The first one is the vanishing remainder

$$\sup_{f \in A_n} \left| t\sqrt{n} r(f, f_0) \right| = o_P(1). \tag{4.16}$$

The second one is verifying

$$\frac{\int_{A_n} e^{\ell_n(f_t) - \ell_n(f_0)} d\Pi(f)}{\int_{A_n} e^{\ell_n(f) - \ell_n(f_0)} d\Pi(f)} = 1 + o_P(1), \forall t \in \mathbb{R}, \tag{4.17}$$

where $f_t = f - \frac{t\Psi_0^{(1)}}{\sqrt{n}}$.

The second condition in (4.17) can be shown with a *change of measure* argument and it guarantees that the posterior has no extra bias term. With these two conditions satisfied, the posterior behavior of $\sqrt{n}(\Psi(f) - \hat{\Psi})$ is asymptotically mean-zero normal with variance $V_0 = \left\| \Psi_0^{(1)} \right\|_L^2$, where

$$\hat{\Psi} = \Psi(f_0) + \frac{W_n(\Psi_0^{(1)})}{\sqrt{n}}$$

is a random centering point.

A crucial step is performing the *change of measure* in (4.17), where we replace f with a shifted function f_t in the integration. This is complicated by the fact that the shifted function f_t does not necessarily have to correspond to a deep ReLU network from the class $\mathcal{F}(L, \mathbf{p}, s)$. In the analysis of trees, for instance, one can condition on the partition parameter and perform the shift of measure on functions supported on the *same* partition, where the

shift only affects step heights. For a deep ReLU network, however, partitions Ω_k and local linear coefficients $(\widetilde{b}_k, \widetilde{\alpha}_k)$ in (4.13) are not independent as they *both* depend on the deep weights $\{W_l, b_l\}_{l=1}^L$. It is thereby not obvious how the shift affects the partitions and the network coefficients. If we want to preserve the partitions of the predictor space, the only "free" parameters left to play with are the top layer weights $\{W_{L+1}, b_{L+1}\}$. Similarly as for trees, we consider conditioning on the *deep* coefficients $\{W_l, b_l\}_{l=1}^L$, which is equivalent to conditioning on γ and $Z = \{Z_l\}_{l=1}^L$, and perform the change of measure only on the top layer. We write the function class conditionally on (γ, Z) as

$$\mathcal{F}(L, \mathbf{p}, \gamma, Z) = \{f \in \mathcal{F}(L, \mathbf{p}, s) : f = W_{L+1} Z_L + b_{L+1} \text{ and } f \text{ has connectivity } \gamma\}. \quad (4.18)$$

Since the prior of $\{W_l, b_l\}_{l=1}^L$ is continuous, there are infinitely many (γ, Z)-dependent shells $\mathcal{F}(L, \mathbf{p}, \gamma, Z)$ inside $\mathcal{F}(L, \mathbf{p}, s)$. The general scheme of our proof is as follows. First, for each shell $\mathcal{F}(L, \mathbf{p}, \gamma, Z)$, we have a local centering point $\hat{\Psi}_Z^\gamma$ and a local variance V_Z^γ. Moreover, the shifted function f_t inside each shell lives on the *same partition* as f and the change of measure can therefore be performed more easily. Second, we show that $\hat{\Psi}_Z^\gamma$ and V_Z^γ converge *uniformly* to a global centering point $\hat{\Psi}$ and a global variance V_0 for all Z and γ inside A_n. This implies that we recover the global BvM on $\mathcal{F}(L, \mathbf{p}, s)$. The details of the local projections and the proof of all theorems are in Section 4.6.

4.3.1 Linear Functionals

To start, we consider the linear functional in (4.2) where $a(\cdot)$ is a constant function in which case $\Psi(f)$ can be viewed as a constant multiple of the average regression surface evaluated

at $\{\boldsymbol{x}_i\}_{i=1}^n$. Let

$$\Psi(f) = \Psi(f_0) + \langle a, f - f_0 \rangle_L, \quad \Psi_0^{(1)} = a,$$
$$\hat{\Psi} = \Psi(f_0) + \frac{W_n(a)}{\sqrt{n}}, \quad V_0 = \|a\|_L^2 .$$

Theorem 29. *Assume the model (4.1), where f is endowed with a prior on $F(L, \mathbf{p}, s)$ defined in (4.7), (4.8) and* (4.9)*. Assume that (4.14) is satisfied and that $f_0 \in \mathcal{H}_p^\alpha$, where $p = \mathcal{O}(1)$ as $n \to \infty$, $\alpha < p$ and $\|f_0\|_\infty \leq F$. When $a(\cdot)$ is constant, we have*

$$\Pi(\sqrt{n}(\Psi(f) - \hat{\Psi}) \mid \boldsymbol{Y}^{(n)}) \rightsquigarrow N(0, \|a\|_L^2)$$

in $\mathbb{P}_0^n$-probability as $n \to \infty$.

Proof. Reference to Section 4.6.4. When $a(\cdot)$ is constant, the shifted functions f_t can be easily constructed by shifting the top intercept $b_{L+1} \to b_{L+1} - \frac{ta}{\sqrt{n}}$. The projection of a is not needed as the remainder term is zero.

Remark 30. *When $a(\cdot)$ is not constant, we need the projection of $a(\cdot)$ (conditional on (γ, Z)), denoted by $a_{[Z]}^\gamma$, to be close to a for all Z and γ supported by A_n. In order for the BvM result to hold, we would then require the* no-bias *condition*

$$\langle a - a_{[Z]}^\gamma, f - f_0 \rangle_L = o_P\left(\frac{1}{\sqrt{n}}\right). \tag{4.19}$$

In order to verify this condition, one could view Z as a collection of random sparse ReLU features and study the approximability of this class. Although there are some studies on the universal approximation error of random ReLU features (Sun et al., 2019; Yehudai and Shamir, 2019), general conditions for the approximation ability of such projections are not yet obvious.

4.3.2 Squared L^2-norm Functional

We consider the quadratic functional in (4.3). The estimation of the L^2-norm is closely related to minimax optimal testing of hypothesis under empirical L^2 distance (Collier et al., 2017). This functional could serve as the risk function and has been used in many testing problems (Cai and Low, 2006a; Dümbgen, 1998). The next theorem relies on the following notation

$$\Psi(f) = \Psi(f_0) + 2\langle f_0, f - f_0\rangle_L + \|f - f_0\|_L^2, \Psi_0^{(1)} = 2f_0,$$
$$\hat{\Psi} = \Psi(f_0) + \frac{2W_n(f_0)}{\sqrt{n}}, V_0 = 4\|f_0\|_L^2.$$

Theorem 31. *Assume the model (4.1), where f is endowed with a prior on $F(L, \mathbf{p}, s)$ defined in (4.7), (4.8) and* (4.9)*. Assume that (4.14) is satisfied and that $f_0 \in \mathcal{H}_p^\alpha$, where $p = \mathcal{O}(1)$ as $n \to \infty$, $\alpha \in (\frac{p}{2}, p)$ and $\|f_0\|_\infty \leq F$. Then we have*

$$\Pi(\sqrt{n}(\Psi(f) - \hat{\Psi}) \mid \boldsymbol{Y}^{(n)}) \rightsquigarrow N(0, 4\|f_0\|_L^2)$$

in $\mathbb{P}_0^n$-probability as $n \to \infty$.

Proof. Reference to Section 4.6.5. For this quadratic functional, we use the (γ, Z)-dependent projection $f_{0[Z]}^\gamma$ to approximate $\Psi_0^{(1)} = 2f_0$ so that the *change of measure* can be conducted through $\{W_{L+1}, b_{L+1}\}$. The additional constraint $\alpha > p/2$ is added to obtain $\xi_n^2 = o(\frac{1}{\sqrt{n}})$, which ensures that the remainder term (4.16) vanishes.

4.4 Adaptive Priors

The results in previous section are predicated on the assumption that the smoothness α is *known*. This is hardly ever satisfied in practice and the next natural step is to inquire whether similar conclusions can be obtained when α is unknown. Similarly as PR18, instead

of the α-dependent choices of the width N and sparsity level s in (4.14), we deploy the following priors that adapt to smoothness

$$\pi(N) = \frac{\lambda^N}{(e^\lambda - 1)N!}, \text{ for } \lambda \in \mathbb{R}, \tag{4.20}$$

$$\pi(s) \propto e^{-\lambda_s s}, \text{ for } \lambda_s > 0. \tag{4.21}$$

The parameter space now consists of shells of sparse ReLU networks with different widths and sparsity levels, i.e.

$$\mathcal{F}(L) = \bigcup_{N=1}^{\infty} \bigcup_{s=0}^{T} \mathcal{F}(L, \mathbf{p}_N^L, s), \tag{4.22}$$

where $\mathcal{F}(L, \mathbf{p}_N^L, s)$ was defined in (4.11). An approximating sieve can be constructed that consists of sparse and not so wide networks, i.e.

$$\mathcal{F}_n = \bigcup_{N=1}^{N_n} \bigcup_{s=0}^{s_n} \mathcal{F}(L, \mathbf{p}_N^L, s) \tag{4.23}$$

with $N_n \asymp n\xi_n^2/\log n$ and $s_n \asymp n\xi_n^2$.

Following the same strategy as in the proof Theorem 6.2 of PR18, we extend the posterior concentration result to the case of adaptive priors (4.7), (4.8), (4.20) and (4.21) (see Theorem 37). The next step is extending the BvM results from the previous section. The following Theorem shows that one can obtain asymptotic normality of the quadratic and linear functionals without the exact knowledge of α.

Theorem 32. *Assume the model (4.1), where f is endowed with a prior on $F(L)$ defined through (4.7), (4.8),* (4.9), (4.20) *and* (4.21) *with $L \asymp \log(n)$. Assume that $f_0 \in \mathcal{H}_p^\alpha$, where $p = \mathcal{O}(1)$ as $n \to \infty, \alpha < p$ and $\|f_0\|_\infty \leq F$.*

(i) For the linear functional $\Psi(f)$ in (4.2) where $a(\cdot)$ is constant, we obtain

$$\Pi(\sqrt{n}(\Psi(f) - \hat{\Psi}) \mid \boldsymbol{Y}^{(n)}) \rightsquigarrow N(0, \|a\|_L^2),$$

where $\hat{\Psi} = \Psi(f_0) + \frac{1}{\sqrt{n}} W_n(a)$.

(ii) For the square L^2-norm functional $\Psi(f)$ in (4.3), we obtain for $\alpha \in (\frac{p}{2}, p)$

$$\Pi(\sqrt{n}(\Psi(f) - \hat{\Psi}) \mid \boldsymbol{Y}^{(n)}) \rightsquigarrow N(0, 4\,\|f_0\|_L^2)$$

where $\hat{\Psi} = \Psi(f_0) + \frac{2}{\sqrt{n}} W_n(f_0)$.

Proof. Reference to Section 4.6.6.

Remark 33. *Similar constraints on the smoothness α have been imposed in other related works (Farrell et al., 2021). However, unlike in other developments (Schmidt-Hieber, 2020; Farrell et al., 2021), the convergence rates we build on are* adaptive *in the sense that, beyond the assumption $\alpha < p$, the exact knowledge of α is* not required. *When the imposed smoothness assumptions do not hold, one could still obtain asymptotic normality via misspecified BvM-type results (Kleijn and van der Vaart, 2012) but uncertainty quantification with the implied credible sets would be problematic.*

Remark 34. *It is worth noting that our results do not hinge on the assumption that f_0 came from the prior. Instead, f_0 is an arbitrary Hölder smooth function, not necessarily a neural network. While the model is ultimately mis-specified, our results are attainable due to the expressibility of deep ReLU networks where one can approximate f_0 with deep learning mappings with a rapidly vanishing error. The fact that our posterior concentrates around the truth at the optimal rate makes the derivation of BvM and valid inference feasible.*

4.5 Discussion

In this paper, we obtained asymptotic normality results for linear and squared L^2-norm functionals for deep, sparse ReLU networks. These results can be used as a basis for semi-parametric inference and can be extended in various ways.

First, one could obtain similar formulations for general smooth linear functionals by verifying the *no bias* condition in (4.19). This relates to the approximation ability of random ReLU features mentioned in Remark 30. The ReLU features act similarly as random rotational trees. However, the nested nature of partitions and local linear functions make the analysis difficult. Random features have gained much attention recently. For instance, Rahimi and Recht (2008a) show how random features can be connected to kernel methods.

Sun et al. (2019) discuss the universal approximation bounds for compositional ReLU features. Huang et al. (2006) and Huang (2014) provide similar results and they propose an implementation of the extreme learning machine implementation, where only the top layer is trained while deep layers are sampled randomly from some distribution. A time-series variant of this algorithm is the Deep Echo State Network (Sun et al., 2017; McDermott and Wikle, 2019).

Another way to obtain BvM for smooth linear functionals would be to construct a less-restrictive projection of the first-order term $\Psi_0^{(1)}$. Schmidt-Hieber (2020) shows that parallelization can be realized using embedding networks. The shifted function f_t could be constructed as an embedding network that simultaneously represents $(f, \Psi_0^{(1)})$. This representation could leverage the approximability of smooth functions a with deep neural networks.

To sum up, our semi-parametric BvM results certify that (semi-parametric) inference with Bayesian deep learning is valid and that meaningful uncertainty quantification is attainable. Possible applications of our results include casual inference, whereby embedding our model within a missing data framework (Ray and van der Vaart, 2020), the average functional

can be used for average treatment effect estimation. In this vein, our results are relevant for the development/understanding of the widely sought after machine learning methods for causal inference (Athey and Wager, 2017). In particular, an extension of our work along these lines will constitute a fully-Bayesian variant of the doubly-robust plug-in approach of Farrell et al. (2021). In addition, the main theorems (Theorem 3.1-3) provide foundations for testing hypotheses such as exceedance of a level $\sum_{i=1}^{n} f_0(x_i) > c$. Lastly, an important future direction will be quantifying uncertainty about the *entire function* f_0 (not only its functionals), which was recently formalized for Bayesian CART by Castillo and Ročková (2021).

Our work is primarily concerned with theoretical frequentist study of the posterior distribution. Investigating practical usefulness and computation of our priors is an important future direction. There are various ways to approximate aspects of deep learning posterior distributions under spike-and-slab prior, see Polson and Ročková (2018) for a discussion on possible implementations. In addition, Deng et al. (2019) proposed an adaptive empirical Bayesian method for sparse deep learning with a self-adaptive spike-and-slab prior.

4.6 Appendix

4.6.1 Rudiments

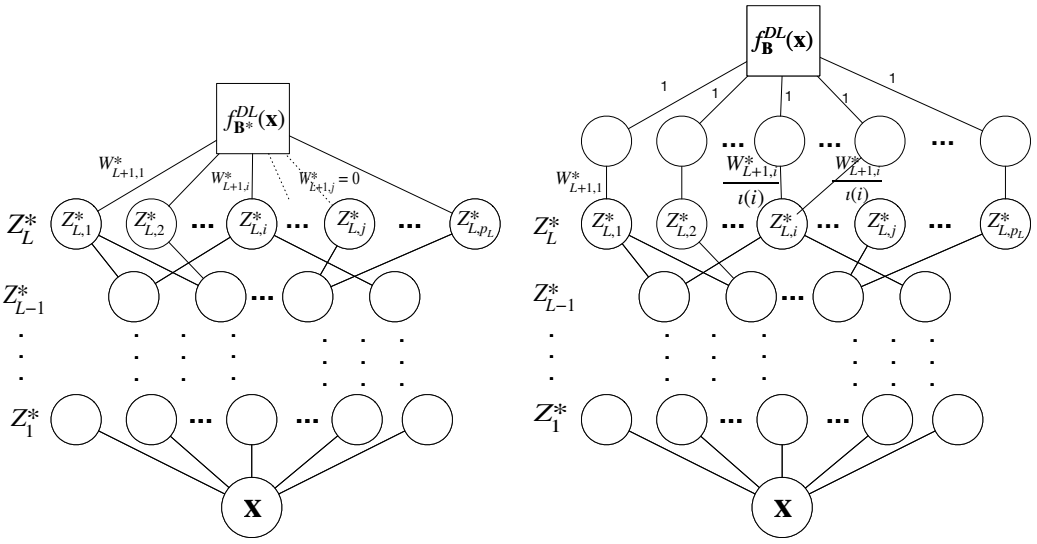
With the prior measure $\Pi(\cdot)$ on $\mathcal{F}(L, \mathbf{p}, s)$, given observed data $\boldsymbol{Y}^{(n)} = (Y_1, \ldots, Y_n)'$, inference about f_0 is carried out via the posterior distribution

$$\Pi(A|\boldsymbol{Y}^{(n)}, \{\boldsymbol{x}_i\}_{i=1}^n) = \frac{\int_A \prod_{i=1}^n \Pi_f(Y_i|\boldsymbol{x}_i) d\Pi(f)}{\int \prod_{i=1}^n \Pi_f(Y_i|\boldsymbol{x}_i) d\Pi(f)}, \forall A \in \mathcal{B}$$

where $\mathcal{B}$ is a σ-field on $\mathcal{F}(L, \mathbf{p}, s)$ and where $\Pi_f(Y_i|\boldsymbol{x}_i)$ is the likelihood function for the output Y_i under f.

4.6.2 Posterior Concentration Rate

First, we show that the posterior concentrates at the optimal (near-minimax) rate. We modify the result in Polson and Ročková (2018) to our prior which differs in two aspects: (1) the top layer is fully connected, (2) the top layer coefficients are assigned a Gaussian prior. First, we show that our fully-connected top layer networks can approximate f_0 as well as the networks considered in Polson and Ročková (2018) (i.e. with a sparse top layer). The following Lemma demonstrates how one can construct a fully connected top layer network from any network considered in PR18 so that their outputs are the same. A graphical illustration of this construction can be found in Figure 4.2.



(a) $f^{DL}_{\boldsymbol{B}}(\boldsymbol{x})$ with sparse top layer (b) $f^{DL}_{\boldsymbol{B}^*}(\boldsymbol{x})$ with fully-connected top layer

Figure 4.2: Network Construction

Lemma 35. *Assume a sparse network $f^{DL}_{\boldsymbol{B}^*} \in \tilde{\mathcal{F}}(L, \mathbf{p}^*, s^*)$ of the form (6) in PR18 with a sparsity pattern γ, where $\tilde{\mathcal{F}}(L, \mathbf{p}^*, s^*)$ is defined in Section 4 of PR18. With $\mathbf{p}^* = (p, p^*_1, \ldots, p^*_L, 1) \in \mathbb{N}^{L+2}$ and $|\gamma| = s^*$, there exists at least one network $f^{DL}_{\boldsymbol{B}} \in \mathcal{F}(L+1, \mathbf{p}, s)$ with $\mathbf{p} = (p, p^*_1, \ldots, p^*_L, p^*_L, 1) \in \mathbb{N}^{L+3}$ and $|\gamma| = s \leq s^* + 2p^*_L$ such that $f^{DL}_{B^*}(\boldsymbol{x}) = f^{DL}_{\boldsymbol{B}}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^p$.*

Proof. We construct one function $f_{\boldsymbol{B}}^{DL}$ that satisfies the stated conditions. We denote $\boldsymbol{B} = \{(W_l, b_l) : 1 \leq l \leq L+2\}$ such that $\mathbf{p} = (p, p_1^*, \ldots, p_L^*, p_L^*, 1) \in \mathbb{N}^{L+3}$ and choose the same deep coefficients $\{W_l, b_l\} = \{W_l^*, b_l^*\}$ for each $1 \leq l \leq L$. The parameters of the top layer are set as $W_{L+2} = 1'_{p_L^*}$ and $b_{L+2} = b_{L+1}^*$. Choosing the matrix W_{L+1} in a way such that $W'_{L+1} 1_{p_L^*} = W_{L+1}^{*\prime}$ we obtain

$$f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}) = W_{L+2} Z_{L+1} + b_{L+2} = W_{L+2} W_{L+1} Z_L^* + b_{L+1}^* = W_{L+1}^* Z_L^* + b_{L+1}^* = f_{\boldsymbol{B}^*}^{DL}(\boldsymbol{x}).$$

The procedure we use to generate W_{L+1} from W_{L+1}^* can be found in Algorithm 6.

Algorithm 6: Network Construction of $\mathcal{F}(L+1, \mathbf{p}, s)$ from $\tilde{\mathcal{F}}(L, \mathbf{p}^*, s^*)$

We assume $W_{L+1,1}^* \neq 0$
Initialize $\{W_l, b_l\}_{l=1}^L = \{W_l^*, b_l^*\}_{l=1}^L, W_{L+1} = 0_{p_L \times p_L}, b_{L+1} = 0, W_{L+2} = \mathbb{I}'_{p_L}, b_{L+2} = b_{L+1}^*$
function $h(j) := \max\{k \leq j : W_{L+1,k} \neq 0\}$
// the index of last connected node (up to j) in layer L+1 of $f_{\boldsymbol{B}^*}^{DL}$
function $\iota(j) := \sum_{i=1}^{p_L} \mathbb{I}(h(i) = h(j))$
// #nodes in layer L+1 in f_B^{DL} that will be connected to $Z_{L,h(j)}$
procedure Generate W_{L+1} from W_{L+1}^*
for *each* $j = 1 : p_L$ **do**
  if $h(j) = j$ // when $Z_{L,j}$ previously connected in $f_{\boldsymbol{B}^*}^{DL}$
  then
    $W_{L+1,i,i} = \frac{W_{L+1,j}^*}{\iota(j)}$ // connect $Z_{L,j}$ to $Z_{L+1,j}$ with the averaged weights
  else
    $W_{L+1,j,h(j)} = \frac{W_{L+1,h(j)}^*}{\iota(j)}$ // connect $Z_{L,h(j)}$ to $Z_{L+1,j}$ with the averaged weights
  end
end

It turns out that the sparsity of this extended network satisfies

$$s = s^* + \|W_{L+2}\|_0 + \|W_{L+1}\|_0 - \left\|W_{L+1}^*\right\|_0 = s^* + 2p_L^* - \left\|W_{L+1}^*\right\|_0 \leq s^* + 2p_L^*. \qquad \square$$

With the construction from Lemma 35, our network class could achieve at least the same approximation error as the one in Schmidt-Hieber (2020). To recover the posterior concentration rate results in Theorem 6.1 in PR18, we impose the following conditions on

(L, s, N)

$$\begin{cases} L^* \propto \log(n) \\ s^* \lesssim n^{p/(2\alpha+p)} \\ N^* \propto n^{p/(2\alpha+p)}/\log(n) \end{cases} \Rightarrow \begin{cases} L = L^* + 1 \propto \log(n) \\ s \leq s^* + 2p_L^* = s^* + 24pN^* \lesssim n^{p/(2\alpha+p)} \\ N = N^* \propto n^{p/(2\alpha+p)}/\log(n) \end{cases}$$

The assumptions on the network structure (depth, width and sparsity) maintain very similar for our new prior.

We formally state the posterior concentration result for our prior below.

Theorem 36. *Assume $f_0 \in \mathcal{H}_p^\alpha$ where $p = O(1)$ as $n \to \infty$, $\alpha < p$ and $\|f_0\|_\infty \leq F$. Let L, s be as in (4.14), and $\mathbf{p} = (p, 12pN, \ldots, 12pN, 1)' \in \mathbb{N}^{L+2}$, where $N = C_N \lfloor n^{p/(2\alpha+p)}/\log(n) \rfloor$ for some $C_N > 0$. Under the priors from Section 2.1, the posterior distribution concentrates at the rate $\epsilon_n = n^{-\alpha/(2\alpha+p)} \log^\delta(n)$ for some $\delta > 1$ in the sense that*

$$\Pi(f_{\boldsymbol{B}}^{DL} \in \mathcal{F}(L, p, s) : \|f - f_0\|_n > M_n \epsilon_n \mid Y^{(n)}) \to 0$$

in $\mathbb{P}_0^n$ probability as $n \to \infty$ for any $M_n \to \infty$.

Proof. The statement can be proved as in Rockova and Polson (2018) by verifying the following three conditions (adopted from Ghosal and Van Der Vaart (2007))

$$\sup_{\epsilon > \epsilon_n} \log \mathcal{E}\left(\frac{\epsilon}{36}; A_{\epsilon,1} \cap \mathcal{F}_n; \|\cdot\|_n\right) \leq n\epsilon_n^2 \tag{4.24}$$

$$\Pi(A_{\epsilon_n,1}) \geq e^{-dn\epsilon_n^2} \tag{4.25}$$

$$\Pi(\mathcal{F} \backslash \mathcal{F}_n) \leq e^{-(d+2)n\epsilon_n^2} \quad \text{for some } d > 2. \tag{4.26}$$

We define $\mathcal{F}_n$, for some $C_n = Cn^{p/(2\alpha+p)} \log^{2\delta}(n)$ and $C > 0$, as

$$\mathcal{F}_n = \{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}(L, \mathbf{p}, s) : \|W_{L+1}\|_2^2 + b_{L+1}^2 \leq C_n\}.$$

Here $\mathcal{F}_n \subset \mathcal{F}(L, \mathbf{p}, s)$ is an approximating space (a sieve) consisting of functions whose top layer weights are contained in a ball of radius $\sqrt{C_n}$ in $\mathbb{R}^{p_L+1}$. We show that this sieve contains most of the prior mass as required in (4.26) for $C > 0$ large enough. Indeed, because $p = \mathcal{O}(1)$ and

$$p_L + 1 = 12\, p\, N + 1 \asymp n^{p/(2\alpha+p)} / \log(n)$$

we have

$$\begin{aligned}\Pi(\mathcal{F} \backslash \mathcal{F}_n) &= \mathbb{P}\left(\|W_{L+1}\|_2^2 + b_{L+1}^2 > C_n \right) \\ &= \mathbb{P}(\chi^2_{p_L+1} > C_n) = \mathbb{P}(e^{\frac{1}{4}\chi^2_{p_L+1}} > e^{\frac{C_n}{4}}) \leq e^{-\frac{C_n}{4}} 2^{(p_L+1)/2} \to 0.\end{aligned}$$

Next, we want to verify the entropy condition (4.24). Because

$$\{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n : \|f\|_\infty \leq \epsilon\} \subset \{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n : \|f\|_n \leq \epsilon\}$$

we have

$$\begin{aligned}\sup_{\epsilon > \epsilon_n} \log \mathcal{E}\left(\frac{\epsilon}{36}; f \in \mathcal{F}_n; \|\cdot\|_\infty \right) &\lesssim \log \left\{ \underbrace{\left(\frac{2}{\frac{\epsilon_n/36}{V(L+1)}} \right)^{s-(p_L+1)}}_{(I)} \underbrace{\left(\frac{\sqrt{C_n}}{\frac{\epsilon_n/36}{V(L+1)}} \right)^{p_L+1}}_{(II)} \right\} \\ &\lesssim (s+1) \log\left(\frac{72}{\epsilon_n}(L+1)(12pN+1)^{2(L+1)} \right) + (p_L+1)\log(n^{p/(2\alpha+p)} \log^{2\delta}(n)) \\ &\lesssim n^{p/(2\alpha+p)} \log(n) \log\left(n/\log^\delta(n) \right) + n^{p/(2\alpha+p)}/\log(n) \log\left(n \log(n) \right) \\ &\lesssim n^{p/(2\alpha+p)} \log^2(n) \lesssim n\epsilon_n^2\end{aligned}$$

for some $\delta > 1$, where

$$V = \prod_{l=0}^{L+1} (P_l + 1) \tag{4.27}$$

and using the fact that $s \lesssim n^{p/(2\alpha+p)}$ and $L \asymp \log(n)$.

The covering number $\mathcal{E}(\frac{\epsilon}{36}; f \in \mathcal{F}_n; \|\cdot\|_\infty)$ consists of two parts. The part (I) stands for the covering number for the deep architecture, while the part (II) is the covering number for the top layer. The calculations of the covering numbers are derived from Lemma 12 of Schmidt-Hieber (2020) which shows

$$\left\| f_{\boldsymbol{B}}^{DL} - f_{\boldsymbol{B}^*}^{DL} \right\|_\infty \leq \|\boldsymbol{B} - \boldsymbol{B}^*\|_\infty V(L+1)$$

with V defined as in (4.27). To make sure $\left\| f_{\boldsymbol{B}}^{DL} - f_{\boldsymbol{B}^*}^{DL} \right\|_\infty \leq \frac{\epsilon_n}{36}$, we want $\|\boldsymbol{B} - \boldsymbol{B}^*\|_\infty \leq \frac{\epsilon_n/36}{2V(L+1)}$. Since all deep parameters are bounded in absolute value by one, we can discretize the unit cube $[-1,1]^{s-p_L-1}$ with a grid of a diameter $\frac{\epsilon_n/36}{2V(L+1)}$ and obtain the covering number in part (I). For the top layer, the weights and the bias term are contained inside a (p_L+1)-dimensional ball with a radius $\sqrt{C_n}$. Part(II) for $\|\cdot\|_\infty$ is bounded by the $\frac{\epsilon_n/36}{2V(L+1)}$-covering number of a Euclidean ball of radius $\sqrt{C_n}$ in (p_L+1)-dimensional space (Edmunds and Triebel, 2008).

Last, we need to show that the prior concentrates enough mass around the truth in the sense of (4.25). From Lemma 35 and Lemma 5.1 in PR18, we know that there exists a neural network $\hat{f}_{\hat{\boldsymbol{B}}} \in \mathcal{F}_n(L, \mathbf{p}, s)$, such that

$$\left\| \hat{f}_{\hat{\boldsymbol{B}}} - f_0 \right\|_n \leq \epsilon/2.$$

We denote the connectivity pattern of $\hat{f}_{\hat{\boldsymbol{B}}}$ as $\hat{\gamma}$ (with $\hat{s} = |\hat{\gamma}|$) and the corresponding set of coefficients as $\hat{\boldsymbol{B}}$. Following the same arguments as in PR18, we have

$$\{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n(L, \mathbf{p}, s) : \left\| f_{\boldsymbol{B}}^{DL} - f_0 \right\|_n \leq \epsilon_n\} \supset \{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n(L, \mathbf{p}, \hat{\gamma}) : \left\| f_{\boldsymbol{B}}^{DL} - f_0 \right\|_n \leq \epsilon_n/2\}.$$

We now denote with $\boldsymbol{\beta} \in \mathbb{R}^T$ and $\hat{\boldsymbol{\beta}} \in \mathbb{R}^T$ the vectorized nonzero coefficients in $\boldsymbol{B}$ and $\hat{\boldsymbol{B}}$ that have the sparsity pattern $\hat{\gamma}$. We use $\gamma(\boldsymbol{\beta})$ to pin down the sparsity pattern of $\boldsymbol{\beta}$. Using

Lemma 12 of Schmidt-Hieber (2020) we have

$$\left\{f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n(L, \mathbf{p}, \hat{\gamma}) : \left\|f_{\boldsymbol{B}}^{DL} - f_0\right\|_n \leq \epsilon_n/2\right\} \supset \left\{\boldsymbol{\beta} \in \mathbb{R}^T : \gamma(\beta) = \hat{\gamma} \text{ and } \left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_\infty \leq \frac{\epsilon_n}{2V(L+1)}\right\}. \tag{4.28}$$

Altogether, we can write

$$\Pi(f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n(L, \mathbf{p}, \hat{s}) : \left\|f_{\boldsymbol{B}}^{DL} - f_0\right\|_n \leq \epsilon_n) > \frac{\Pi(f_{\boldsymbol{B}}^{DL} \in \mathcal{F}_n(L, \mathbf{p}, \hat{\gamma}) : \left\|f_{\boldsymbol{B}}^{DL} - f_0\right\|_n \leq \epsilon_n/2)}{\binom{T-p_L-1}{\hat{s}-p_L-1}}$$
$$> \frac{1}{\binom{T-p_L-1}{\hat{s}-p_L-1}} \Pi\left(\boldsymbol{\beta} \in \mathbb{R}^T : \gamma(\beta) = \hat{\gamma} \text{ and } \left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_\infty \leq \frac{\epsilon_n}{2V(L+1)}\right).$$

We note that with $\hat{s} \asymp n^{p/(2\alpha+p)}, L \asymp \log(n)$ and $N \asymp n^{p/(2\alpha+p)}/\log(n)$

$$\frac{1}{\binom{T-p_L-1}{\hat{s}-p_L-1}} \geq e^{-(L+1)\hat{s}\log(12pN)} > e^{-D_1 \log^2(n) n^{p/(2\alpha+p)}}$$

for some $D_1 > 0$. In addition, under the uniform prior on the deep coefficients and the standard normal prior on the top layer, we can write

$$\Pi\left(\boldsymbol{\beta} \in \mathbb{R}^T : \gamma(\beta) = \hat{\gamma} \text{ and } \left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_\infty \leq \frac{\epsilon_n}{2V(L+1)}\right) \tag{4.29}$$
$$\geq \left(\frac{\epsilon_n}{2V(L+1)}\right)^{\hat{s}-p_L-1} \prod_{j>T-p_L-1} \Pi\left(\left|\beta_j - \hat{\beta}_j\right| \leq \frac{\epsilon_n}{2V(L+1)}\right)$$
$$= \left(\frac{\epsilon_n}{2V(L+1)}\right)^{\hat{s}-p_L-1} \prod_{j>T-p_L-1} \int_{-\frac{\epsilon_n}{2V(L+1)}}^{\frac{\epsilon_n}{2V(L+1)}} d\Pi(\beta_j - \hat{\beta}_j). \tag{4.30}$$

where the last $T - p_L - 1$ coefficients in $\boldsymbol{\beta}$ are the top layer weights and bias as shown in (4.9).

We want to recenter the normal distribution at 0 rather than $\hat{\beta}_j$ by using the following

inequality

$$\frac{dN(\hat{\beta}_j, 1)}{dN(0, \frac{1}{2})} = e^{-\frac{1}{2}(\beta_j - \hat{\beta}_j)^2 + \beta_j^2} = e^{\frac{1}{2}(\beta_j + \hat{\beta}_j)^2 - \hat{\beta}_j^2} \geq e^{-\hat{\beta}_j^2}.$$

Then we can continue with the lower bound for (4.30) as follows

$$\begin{aligned}
(4.30) &\geq \left(\frac{\epsilon_n}{2V(L+1)}\right)^{\hat{s}-p_L-1} e^{-\sum_{j>T-p_L-1}\hat{\beta}_j^2} \left(\int_{-\frac{\epsilon_n}{2V(L+1)}}^{\frac{\epsilon_n}{2V(L+1)}} dN\left(0, \frac{1}{2}\right)\right)^{p_L+1} \\
&\geq \left(\frac{\epsilon_n}{2V(L+1)}\right)^{\hat{s}-p_L-1} e^{-C_n} \left(e^{-(\frac{\epsilon_n}{2V(L+1)})^2} \frac{\epsilon_n}{\sqrt{\pi}V(L+1)}\right)^{p_L+1} \\
&\geq \left(\frac{2}{\sqrt{2\pi}}\right)^{p_L+1} \left(\frac{\epsilon_n}{2V(L+1)}\right)^{\hat{s}} e^{-C_n} e^{-\frac{(p_L+1)\epsilon_n}{4(12pN+1)^{(L+1)}(L+1)}} \geq e^{-D_2 n^{p/(2\alpha+p)} \log^2(n)}
\end{aligned}$$

for some $D_2 > 0$ and recall that $C_n = Cn^{p/(2\alpha+p)} \log^{2\delta}(n)$.Thus we can combine the bounds and conclude that $e^{-(D_1+D_2)n^{p/(2\alpha+p)} \log^2(n)} \geq e^{-dn\epsilon_n^2}$ for some $\delta > 1$ and $d > D_1 + D_2$. The proof is now complete.

□

It is worth noting that the same concentration rate still holds if we use $N(0, 1)$ prior on *all* parameters. We could define

$$\mathcal{F}_n = \{\|\beta\|_2^2 \leq C_n\}.$$

The prior mass condition in (4.26) is

$$\Pi(\mathcal{F} \backslash \mathcal{F}_n) = \mathbb{P}(\chi_s^2 > C_n) \leq e^{-C_1 n^{p/(2\alpha+p)} \log^{2\delta}(n)}.$$

The entropy condition in (4.24) is

$$\begin{aligned}\sup_{\epsilon>\epsilon_n} \log \mathcal{E}(\frac{\epsilon}{36}, f \in \mathcal{F}_n; \|\cdot\|_\infty) &\lesssim \log \left\{ \left(\frac{\sqrt{C_n}}{\frac{\epsilon_n/36}{V(L+1)}} \right)^s \right\} \\ &\lesssim (s+1) \log \left(\frac{72}{\epsilon_n}(L+1)(12pN+1)^{2(L+1)} \right) + s \log(Cn^{p/(2\alpha+p)} \log^{2\delta}(n)) \\ &\lesssim n^{p/(2\alpha+p)} \log(n) \log\left(n/\log^\delta(n)\right) + n^{p/(2\alpha+p)} \log(n \log(n)) \\ &\lesssim n\epsilon_n^2\end{aligned}$$

for some $\delta > 1$, using the fact that $s \lesssim n^{p/(2\alpha+p)}$ and $L \asymp \log(n)$.

The prior concentration condition in (4.25) can be proved by changing (4.30) into

$$\begin{aligned}&\Pi(\beta \in \mathbb{R}^T : \gamma(\beta) = \hat{\gamma}, \sum_j \beta_j^2 \leq C_n \text{ and } \left\|\beta - \hat{\beta}\right\|_\infty \leq \frac{\epsilon_n}{2V(L+1)}) \\ &\geq e^{-\sum_j \hat{\beta}_j^2} \left(\int_{-\frac{\epsilon_n}{2V(L+1)}}^{\frac{\epsilon_n}{2V(L+1)}} dN(0, \frac{1}{2}) \right)^{\hat{s}} \\ &\geq e^{-C_n} \left(e^{-(\frac{\epsilon_n}{2V(L+1)})^2} \frac{\epsilon_n}{\sqrt{\pi}V(L+1)} \right)^{\hat{s}} \\ &\geq e^{-C_n} \left(\frac{\epsilon_n}{\sqrt{\pi}V(L+1)} \right)^{\hat{s}} e^{-\frac{\hat{s}\epsilon_n}{4(12pN+1)^{(L+1)}(L+1)}} \geq e^{-Dn^{p/(2\alpha+p)} \log^2(n)}. \qquad \square\end{aligned}$$

Theorem 37 (adaptive priors)**.** *Assume $f_0 \in \mathcal{H}_p^\alpha$, where $p = O(1)$ as $n \to \infty$, $\alpha < p$, and $\|f_0\|_\infty \leq F$. Let $L \asymp \log(n)$ and assume priors for N and s as in (4.20) and (4.21). Assume the prior of f as given by (4.7) and (4.8). Then the posterior distribution concentrates at the rate $\xi_n = n^{-\alpha/(2\alpha+p)} \log^\delta(n)$ for $\delta > 1$ in the sense that*

$$\Pi(f \in \mathcal{F}(L) : \|f - f_0\|_L > M_n \xi_n \mid \mathbf{Y}^{(n)}) \to 0$$

in $\mathbb{P}_0^n$ probability as $n \to \infty$ for any $M_n \to \infty$.

The proof for Theorem 37 follows the same techniques used in Theorem 6.2 of PR18.

And this adaptive results also hold for networks with standard normal priors on all weights.

4.6.3 Preparations for Main Theorems

The general framework for first-order approximation of functionals is as follows

Theorem 38. *(Castillo and Rousseau, 2015) Consider the model $\mathbb{P}_0^n$, a real-valued functional $f \to \Psi(f)$ and $\langle \cdot, \cdot \rangle_L, \Psi_0^{(1)}, W_n$, as defined above. Suppose that (4.16) is satisfied, and denote*

$$\hat{\Psi} = \Psi(f_0) + \frac{W_n(\Psi_0^{(1)})}{\sqrt{n}}, \qquad V_0 = \left\| \Psi_0^{(1)} \right\|_L^2.$$

Let Π be a prior distribution on f. Let A_n be any measurable set such that

$$\Pi(A_n \mid \boldsymbol{Y}^{(n)}) = 1 + o_P(1), \quad as\ n \to \infty.$$

Then for any real t with f_t as

$$f_t = f - \frac{t\Psi_0^{(1)}}{\sqrt{n}},$$

we could write

$$\mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n] = e^{o_P(1)+t^2V_0/2} \frac{\int_{A_n} e^{\ell_n(f_t)-\ell_n(f_0)} d\Pi(f)}{\int_{A_n} e^{\ell_n(f)-\ell_n(f_0)} d\Pi(f)}.$$

Moreover, if

$$\frac{\int_{A_n} e^{\ell_n(f_t)-\ell_n(f_0)} d\Pi(f)}{\int_{A_n} e^{\ell_n(f)-\ell_n(f_0)} d\Pi(f)} = 1 + o_P(1),\ \forall t \in \mathbb{R} \tag{4.31}$$

is satisfied, then the posterior distribution of $\sqrt{n}(\Psi(f) - \hat{\Psi})$ is asymptotically normal and mean-zero, with variance V_0.

Proof. Set $R_n(\cdot, \cdot) = 0, \Psi_0^{(2)} = 0, \mu_n = 0$ in Theorem 2.1 of Castillo and Rousseau (2015). □

4.6.3.0.1 Projection of Functions The intuition of our projection conditional on (γ, Z) is to maintain the same partitions for the shifted function in (4.17) and perform the *change of measure* locally. We first give the notation for Z^L, which are the nodes in the top layer. Let $Z_{Lj}, j = 1, \ldots, p_L$ denote the j^{th} node in L^{th} layer, which can be written as a sum of local linear functions, respectively:

$$Z_{Lj}(\boldsymbol{x}) = \sum_{k=1}^{K_L} \mathbb{I}(\boldsymbol{x} \in \Omega_k^j)\{\tilde{\beta}_k^{j'} \boldsymbol{x} + \tilde{\alpha}_k^j\}$$

here the partitions $\{\Omega_k^j\}_{k=1}^{K_L}$ and coefficients $\{\tilde{\beta}_k^j, \tilde{\alpha}_k^j\}_{k=1}^{K_L}$ are determined by $\{W_l, b_l\}_{l=1}^L$.

For simplicity of notation, we denote $W_{L+1} = (w_1, \ldots, w_{p_L})'$. Then the output can be written as:

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{j=1}^{p_L} w_j Z_{Lj}(\boldsymbol{x}) + b_{L+1} \\ &= \sum_{k_1=1}^{K_L} \cdots \sum_{k_{p_L}=1}^{K_L} \mathbb{I}\left(\boldsymbol{x} \in \bigcap_{j=1}^{p_L} \Omega_{k_j}^j\right) \left\{ \left(\sum_{j=1}^{p_L} w_j \tilde{\beta}_{k_j}^{j'}\right) \boldsymbol{x} + \left(\sum_{j=1}^{p_L} w_j \tilde{\alpha}_{k_j}^j + b_{L+1}\right) \right\}. \end{aligned}$$

We denote the projection of function $a(\boldsymbol{x})$ conditional on $\{W_l, b_l\}_{l=1}^L$ with $a_{[Z]}^\gamma$, since conditional on $\{W_l, b_l\}_{l=1}^L$ is equivalent to conditional on (γ, Z):

$$\begin{aligned} (W^a, b^a) &= \arg\min_{W_{L+1}, b_{L+1} \in \mathcal{F}_n(L, \mathbf{p}, \gamma, Z)} \|W Z_L(\boldsymbol{x}) + b - a(\boldsymbol{x})\|_L, \\ a_{[Z]}^\gamma(\boldsymbol{x}) &= W^a Z_L(\boldsymbol{x}) + b^a. \end{aligned}$$

The projection $a_{[Z]}^\gamma$ can also be viewed as the best approximation to a conditional on (γ, Z).

Similarly, we denote projection of f_0 onto $\{W_l, b_l\}_{l=1}^L$ as $f_{0[Z]}^\gamma$:

$$(W^0, b^0) = \arg\min_{W_{L+1}, b_{L+1} \in \mathcal{F}_n(L, \mathbf{p}, \gamma, Z)} \|WZ_L(\boldsymbol{x}) + b - f_0(\boldsymbol{x})\|_L \,, \tag{4.32}$$

$$f_{0[Z]}^\gamma(\boldsymbol{x}) = W^0 Z_L(\boldsymbol{x}) + b^0. \tag{4.33}$$

Note that $f \in \{WZ_L(\boldsymbol{x}) + b : W \in \mathbb{R}^{p_L}, b \in \mathbb{R}\}$, so naturally we have $\left\| f_{0[Z]}^\gamma - f_0 \right\|_L \leq \|f - f_0\|_L$.

4.6.4 Proof of Theorem 29

We will perform the analysis locally on the sets $A_n \equiv A_n^{M_n}$ from (4.15) for some $M_n \to \infty$. We use the fact that convergence of Laplace transforms for all t in probability implies convergence in distribution in probability (Castillo and Rousseau, 2015). The posterior decomposes into a mixture of laws with weights $\Pi(\gamma \mid \boldsymbol{Y}^{(n)})$, where γ is the vector encoding the connectivity pattern with prior in (4.10). We denote with $I_{n,\gamma} = \mathbb{E}^\Pi[e^{t\sqrt{n}(\Psi(f) - \hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma]$ and write

$$I_n := \mathbb{E}^\Pi[e^{t\sqrt{n}(\Psi(f) - \hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n] = \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n) I_{n,\gamma}.$$

Next, we want to show that on the event A_n and uniformly for all $\gamma \in \mathcal{V}^{\mathbf{p},s}$

$$I_{n,\gamma} = e^{o_P(1) + t^2 V_0/2}(1 + o(1)) \quad \text{as } n \to \infty$$

so that $I_n = e^{o_P(1) + t^2 V_0/2}(1 + o(1))$.

We choose γ such that $\mathcal{F}(L, \mathbf{p}, \gamma) \cap A_n \neq \emptyset$ and for $f \in \mathcal{F}(L, \mathbf{p}, \gamma) \cap A_n$ we expand the

linear functional as $\Psi(f) - \Psi(f_0) = \langle a, f - f_0 \rangle_L$ which yields

$$\Psi_0^{(1)} = a,$$
$$r(f, f_0) = 0.$$

The remainder condition (4.16) is thus trivially satisfied. To verify the second condition (4.17), we choose the shifted function f_t as

$$f_t = f - \frac{ta}{\sqrt{n}}.$$

Due to the fact that our class of neural networks has a top linear layer, the function f_t shares the same deep connectivity structure as f where only the top layer intercepts b_{L+1}^t have been shifted. The *change of measure* thus only influences b_{L+1} where $b_{L+1}^t = b_{L+1} - \frac{ta}{\sqrt{n}}$. Next, we can write

$$I_{n,\gamma} = e^{\frac{t^2}{2}\|a\|_L^2} \times \frac{\int_{A_n} e^{\ell_n(f_t) - \ell_n(f_0)} d\Pi(f \mid \gamma)}{\int_{A_n} e^{\ell_n(f) - \ell_n(f_0)} d\Pi(f \mid \gamma)} \tag{4.34}$$
$$= e^{\frac{t^2}{2}\|a\|_L^2} \times \frac{\int_{f_t + \frac{ta}{\sqrt{n}} \in A_n} e^{\ell_n(f_t) - \ell_n(f_0)} d\Pi(f_t \mid \gamma) \frac{d\Pi(f|\gamma)}{d\Pi(f_t|\gamma)}}{\int_{A_n} e^{\ell_n(f) - \ell_n(f_0)} d\Pi(f \mid \gamma)}. \tag{4.35}$$

Next, we show that the ratio above converges to 1 as $n \to \infty$. We have

$$\frac{d\Pi(f \mid \gamma)}{d\Pi(f_t \mid \gamma)} = \frac{d\Pi(\{W_l, b_l\}_{l=1}^L, W_{L+1}, b_{L+1} \mid \gamma)}{d\Pi(\{W_l, b_l\}_{l=1}^L, W_{L+1}, b_{L+1}^t \mid \gamma)} = \frac{d\Pi(\{W_l, b_l\}_{l=1}^L \mid \gamma) d\Pi(W_{L+1}) d\Pi(b_{L+1})}{d\Pi(\{W_l, b_l\}_{l=1}^L \mid \gamma) \Pi(W_{L+1}) \Pi(b_{L+1}^t)} = \frac{d\Pi(b_{L+1})}{d\Pi(b_{L+1}^t)}$$
$$\frac{d\Pi(b_{L+1})}{d\Pi(b_{L+1}^t)} = \frac{\phi(b_{L+1})}{\phi(b_{L+1} - \frac{ta}{\sqrt{n}})} = \exp\left\{-\frac{1}{2}\left[b_{L+1}^2 - (b_{L+1} - \frac{ta}{\sqrt{n}})^2\right]\right\} = \exp\left(-\frac{atb_{L+1}}{\sqrt{n}} + \frac{t^2a^2}{2n}\right)$$

Next, we note (from the definition of the sieve $\mathcal{F}_n$ and C_n in the proof of Theorem 36)

$$\frac{|b_{L+1}|}{\sqrt{n}} \leq \frac{\sqrt{C_n}}{\sqrt{n}} \lesssim n^{-\frac{\alpha}{2\alpha+p}} \log^{\delta}(n).$$

Going back to (4.34), we now have for some $c > 0$

$$e^{-cn^{-\frac{\alpha}{2\alpha+p}}\log^{\delta}(n)+\frac{t^2a^2}{2n}+\frac{t^2}{2}\|a\|_L^2} \times \frac{\Pi\left(f+\frac{ta}{\sqrt{n}} \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma\right)}{\Pi\left(f \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma\right)} \leq I_{n,\gamma}$$
$$\leq e^{cn^{-\frac{\alpha}{2\alpha+p}}\log^{\delta}(n)+\frac{t^2a^2}{2n}+\frac{t^2}{2}\|a\|_L^2} \times \frac{\Pi\left(f+\frac{ta}{\sqrt{n}} \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma\right)}{\Pi\left(f \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma\right)}. \quad (4.36)$$

Next, from

$$\|f-f_0\|_L - \left\|\frac{ta}{\sqrt{n}}\right\|_L \leq \left\|f+\frac{ta}{\sqrt{n}}-f_0\right\|_L \leq \|f-f_0\|_L + \left\|\frac{ta}{\sqrt{n}}\right\|_L$$

it is clear that

$$\left\{f : \|f-f_0\|_L \leq M_n\xi_n - \left\|\frac{ta}{\sqrt{n}}\right\|_L\right\} \subset \left\{f : \left\|f+\frac{ta}{\sqrt{n}}-f_0\right\|_L \leq M_n\xi_n\right\} \subset \left\{f : \|f-f_0\|_L \leq M_n\xi_n + \left\|\frac{ta}{\sqrt{n}}\right\|_L\right\}$$

This yields

$$\Pi\left(f : \|f-f_0\|_L \leq \xi_n - \left\|\frac{ta}{\sqrt{n}}\right\|_L \mid \boldsymbol{Y}^{(n)}, \gamma\right) \leq \Pi\left(f : f+\frac{ta}{\sqrt{n}} \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma\right)$$
$$\leq \Pi\left(f : \|f-f_0\|_L \leq \xi_n + \left\|\frac{ta}{\sqrt{n}}\right\|_L \mid \boldsymbol{Y}^{(n)}, \gamma\right).$$

Since the concentration rate is slower than $1/\sqrt{n}$, i.e. $\xi_n = n^{-\alpha/(2\alpha+p)}\log^{\delta}(n) \gtrsim n^{-1/2}$, we have $\Pi(f+\frac{ta}{\sqrt{n}} \in A_n) \to \Pi(f \in A_n)$, as $n \to \infty$. From the sandwich inequality (4.36), we have $I_{n,\gamma} \to e^{\frac{t^2\|a\|_L^2}{2}}$ for any $t \in \mathbb{R}$ as $n \to \infty$.

4.6.5 Proof of Theorem 31

Similar to the linear functional case, the posterior decomposes into a mixture of laws with weights $\Pi(\gamma \mid \boldsymbol{Y}^{(n)})$, where γ is the vector encoding the connectivity pattern with a prior in

(4.10). We can write

$$I_n := \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n] = \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n) I_{n,\gamma} \tag{4.37}$$

where

$$I_{n,\gamma} := \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma].$$

We further decompose each $I_{n,\gamma}$ by conditioning on the deep weights $\{W_l, b_l\}_{l=1}^{L}$. We can write

$$\begin{aligned}\Pi(\{W_l, b_l\}_{l=1}^{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) &= \Pi(W_{L+1}, b_{L+1} \mid \{W_l, b_l\}_{l=1}^{L}, \boldsymbol{Y}^{(n)}, A_n, \gamma)\Pi(\{W_l, b_l\}_{l=1}^{L} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) \\ &= \Pi(W_{L+1}, b_{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma, Z)\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma),\end{aligned}$$

since $Z = \{Z_l\}_{l=1}^{L}$ is fully determined by $\{W_l, b_l\}_{l=1}^{L}$ and we can thereby replace conditioning on $\{W_l, b_l\}_{l=1}^{L}$ by conditioning on Z. We can further dissect $I_{n,\gamma}$ by conditioning on Z

$$I_{n,\gamma} = \int I_{n,\gamma}^{Z} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma), \quad \text{where} \quad I_{n,\gamma}^{Z} := \int e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} d\Pi(W_{L+1}, b_{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma, Z).$$

In the rest of the proof, we show that $I_{n,\gamma}^{Z} \to \exp(-t^2 V_0/2)$ uniformly for all γ and Z such that $f \in A_n$. This can be done in two steps. First, we show that conditional on $(\boldsymbol{Y}^{(n)}, A_n, \gamma, Z)$, $\Psi(f)$ asymptotically centers at a local (γ, Z)-dependent centering point $\hat{\Psi}_Z^{\gamma}$ with a local (γ, Z)-dependent variance V_Z^{γ} (both defined later). In the second step, we show that the local centering points $\hat{\Psi}_Z^{\gamma}$ are close to the global centering point $\hat{\Psi}$ and that the local variances V_Z^{γ} converge to V_0 uniformly for all γ and Z such that $f \in A_n$.

We define the (γ, Z)-dependent local centering point and variance as

$$\hat{\Psi}_Z^\gamma = \Psi(f_0) + \frac{W_n(2f_{0[Z]}^\gamma)}{\sqrt{n}} \quad \text{and} \quad V_Z^\gamma = 4\left\|f_{0[Z]}^\gamma\right\|_L^2, \tag{4.38}$$

where $f_{0[Z]}^\gamma$ is the $\|\cdot\|_L$ projection of f_0 on the set of deep learning networks f with a connectivity pattern γ and hidden nodes Z defined in (4.33).

For any $f \in \mathcal{F}(L, \mathbf{p}, \gamma)$, the squared L^2-norm functional can be expanded as

$$\begin{aligned}\Psi(f) - \Psi(f_0) &= 2\langle f_0, f - f_0\rangle_L + \|f - f_0\|_L^2 \\ &= 2\langle f_{0[Z]}^\gamma, f - f_0\rangle_L + \|f - f_0\|_L^2 + 2\langle f_0 - f_{0[Z]}^\gamma, f - f_0\rangle_L.\end{aligned}$$

Note that $\left\|f_{0[Z]}^\gamma - f_0\right\|_L \leq \|f - f_0\|_L$ for any f which has a connectivity pattern γ and hidden nodes Z.

This expansion yields the first-order and remainder terms

$$\begin{aligned}\Psi_0^{(1)} &= 2f_{0[Z]}^\gamma, \\ r(f, f_0) &= \|f - f_0\|_L^2 + 2\langle f_0 - f_{0[Z]}^\gamma, f - f_0\rangle_L.\end{aligned}$$

To ensure asymptotical normality of $\Psi(f)$, we first need to ensure the local shape condition in (4.16). Assuming that the smoothness α satisfies

$$\alpha > p/2 \tag{4.39}$$

we have for $f \in A_n$ with a connectivity γ and hidden nodes Z

$$\begin{aligned} r(f, f_0) &= \|f - f_0\|_L^2 + 2\langle f_0 - f_{0[Z]}^{\gamma}, f - f_0 \rangle_L \\ &\leq 2\|f - f_0\|_L^2 + \left\| f_0 - f_{0[Z]}^{\gamma} \right\|_L^2 \\ &\leq 3\|f - f_0\|_L^2 \lesssim \xi_n^2 = n^{-\frac{2\alpha}{2\alpha+p}} \log^{2\delta} = o\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

Next, to verify the second sufficient condition (4.17) we define the shifted function f_t as

$$f_t = f - \frac{2t f_{0[Z]}^{\gamma}}{\sqrt{n}}.$$

Then we use the local centering point $\hat{\Psi}_Z^{\gamma}$ in (4.38) to define

$$\begin{aligned} \tilde{I}_{n,\gamma}^{Z} :=& \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f) - \hat{\Psi}_Z^{\gamma})} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma, Z] \\ =& e^{2t^2 \left\| f_{0[Z]}^{\gamma} \right\|_L^2} \times \frac{\int_{A_n} e^{\ell_n(f_t) - \ell_n(f_0)} d\Pi(f \mid \gamma, Z)}{\int_{A_n} e^{\ell_n(f) - \ell_n(f_0)} d\Pi(f \mid \gamma, Z)} \\ =& e^{2t^2 \left\| f_{0[Z]}^{\gamma} \right\|_L^2} \times \frac{\int_{f_t + \frac{2t f_{0[Z]}^{\gamma}}{\sqrt{n}} \in A_n} e^{\ell_n(f_t) - \ell_n(f_0)} d\Pi(f_t \mid \gamma, Z) \frac{d\Pi(f|\gamma,Z)}{d\Pi(f_t|\gamma,Z)}}{\int_{A_n} e^{\ell_n(f) - \ell_n(f_0)} d\Pi(f \mid \gamma, Z)} \end{aligned} \tag{4.40}$$

For simplicity of notation, we first denote $\zeta = (W_{L+1}, b_{L+1})' \in \mathbb{R}^{p_L+1}$ and $\zeta^t = (W_{L+1}^t, b_{L+1}^t)' \in \mathbb{R}^{p_L+1}$ and $\Delta = (W^0, b^0)'$ as defined in (4.32). Then we can simply write $\zeta^t = \zeta - \frac{2t}{\sqrt{n}}\Delta$.

Since all parameters are a-priori independent and there is no sparsity structure placed

on $\{W_{L+1}, b_{L+1}\}$, the prior ratio $\frac{d\Pi(f|\gamma,Z)}{d\Pi(f_t|\gamma,Z)}$ can be calculated as

$$\begin{aligned}\frac{d\Pi(f \mid \gamma, Z)}{d\Pi(f_t \mid \gamma, Z)} &= \frac{d\Pi(W_{L+1})}{d\Pi(W_{L+1}^t)} \frac{d\Pi(b_{L+1})}{d\Pi(b_{L+1}^t)} = \frac{d\Pi(\zeta)}{d\Pi(\zeta^t)} \\ &= \prod_{i=1}^{p_{L+1}} \exp\left\{-\frac{1}{2}\left[\zeta^2 - (\zeta_i - \frac{2t}{\sqrt{n}}\Delta_i)^2\right]\right\} \\ &= \exp\left\{\sum_{i=1}^{p_{L+1}}\left[-\zeta_i \frac{\Delta_i t}{\sqrt{n}} + \frac{2t^2\Delta_i^2}{n}\right]\right\}.\end{aligned}$$

Similar to our previous proof, we have under the assumption $\alpha > p/2$

$$\left|\sum_{i=1}^{p_{L+1}} \zeta_i \frac{\Delta_i t}{\sqrt{n}}\right| \leq \frac{t}{\sqrt{n}} \|\zeta\|_2 \|\Delta\|_2 \lesssim \frac{C_n}{\sqrt{n}} = o(1), \tag{4.41}$$

where we used the fact that both f and $f_{0[Z]}^{\gamma}$ are contained in A_n and thereby have their top coefficients contained in a ball of radius $\sqrt{C_n}$ (recall the definition of C_n in the proof of Theorem 36).

Now, using the fact that

$$\|f - f_0\|_L - 2\left\|\frac{t f_{0[Z]}^{\gamma}}{\sqrt{n}}\right\|_L \leq \left\|f + \frac{2t f_{0[Z]}^{\gamma}}{\sqrt{n}} - f_0\right\|_L \leq \|f - f_0\|_L + 2\left\|\frac{t f_{0[Z]}^{\gamma}}{\sqrt{n}}\right\|_L$$

we have

$$\begin{aligned}&\Pi\left(f : \|f - f_0\|_L \leq \xi_n - 2\left\|\frac{t f_{0[Z]}^{\gamma}}{\sqrt{n}}\right\|_L \mid \boldsymbol{Y}^{(n)}, \gamma, Z\right) \\ &\leq \Pi\left(f + \frac{2t f_{0[Z]}^{\gamma}}{\sqrt{n}} \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma, Z\right) \leq \Pi\left(f : \|f - f_0\|_L \leq \xi_n + 2\left\|\frac{t f_{0[Z]}^{\gamma}}{\sqrt{n}}\right\|_L \mid \boldsymbol{Y}^{(n)}, \gamma, Z\right).\end{aligned}$$

Again, since the concentration rate is slower than $1/\sqrt{n}$, i.e. $\xi_n = n^{-\alpha/(2\alpha+p)} \log^{\delta}(n) \gtrsim$

$n^{-1/2}$, we have

$$\frac{\Pi(f + \frac{2tf^{\gamma}_{0[Z]}}{\sqrt{n}} \in A_n \mid \boldsymbol{Y}^{(n)}, \gamma, Z)}{\Pi(A_n \mid \boldsymbol{Y}^{(n)}, \gamma, Z)} \to 1, \forall t \in \mathbb{R}. \tag{4.42}$$

Hence, with (4.39), (4.41) and (4.42), one concludes $\tilde{I}^{Z}_{n,\gamma} \to e^{2t^2 \left\| f^{\gamma}_{0[Z]} \right\|^2_L}$ as $n \to \infty$ using a similar sandwich inequality in (4.36). In other words, we have

$$\tilde{I}^{Z}_{n,\gamma} = e^{t^2 V^{\gamma}_Z/2}(1 + o(1)). \tag{4.43}$$

Recall the definition of a local centering point $\hat{\Psi}^{\gamma}_Z$ and a local variance V^{γ}_Z in (4.38). Then we can write

$$\begin{aligned}
I^{Z}_{n,\gamma} &= \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f) - \hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma, Z] \\
&= \mathbb{E}^{\Pi}[e^{t\sqrt{n}[(\Psi(f) - \hat{\Psi}^{\gamma}_Z) + (\hat{\Psi}^{\gamma}_Z - \hat{\Psi})]} \mid \boldsymbol{Y}^{(n)}, A_n, \gamma, Z] \\
&= \tilde{I}^{Z}_{n,\gamma} \times e^{t\sqrt{n}(\hat{\Psi}^{\gamma}_Z - \hat{\Psi})} \\
&= (1 + o(1)) e^{t^2 V^{\gamma}_Z/2 + t\sqrt{n}(\hat{\Psi}^{\gamma}_Z - \hat{\Psi})} \\
&= (1 + o(1)) e^{t^2 V_0/2 + t^2 (V^{\gamma}_Z - V_0)/2 + t\sqrt{n}(\hat{\Psi}^{\gamma}_Z - \hat{\Psi})}.
\end{aligned}$$

The proof will be complete once we show the following condition uniformly for all γ such that $f \in A_n$

$$\begin{aligned}
I_{n,\gamma} &= \int I^{Z}_{n,\gamma} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) \\
&= (1 + o(1)) e^{t^2 V_0/2} \int e^{t^2 (V^{\gamma}_Z - V_0)/2 + t\sqrt{n}(\hat{\Psi}^{\gamma}_Z - \hat{\Psi})} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) \to e^{t^2 V_0/2}, \text{ as } n \to \infty.
\end{aligned}$$

This is equivalent to showing

$$\int e^{t^2 (V^{\gamma}_Z - V_0)/2 + t\sqrt{n}(\hat{\Psi}^{\gamma}_Z - \hat{\Psi})} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) = 1 + o_P(1). \tag{4.44}$$

Since we work conditionally on the set A_n, we have $\|f^{\gamma}_{0[Z]} - f_0\|_L \lesssim \xi_n$ and thereby

$$\begin{aligned}
\sqrt{n}(\hat{\Psi} - \hat{\Psi}^{\gamma}_Z) &= W_n(f^{\gamma}_{0[Z]} - f_0) = o_P(1), \\
\left|V^{\gamma}_z - V\right| &= 4\left|\left\|f^{\gamma}_{0[Z]}\right\|^2_L - \|f_0\|^2_L\right| \\
&\lesssim 2\,\|f_0\|_L \left\|f^{\gamma}_{0[Z]} - f_0\right\|_L + \left\|f^{\gamma}_{0[Z]} - f_0\right\|^2_L \\
&\lesssim \left\|f^{\gamma}_{0[Z]} - f_0\right\|_L \leq \xi_n
\end{aligned}$$

under the assumption that $\|f_0\|_L \leq F$.

Using the smoothness assumption (4.39), we have $\xi_n^2 = o\left(\frac{1}{\sqrt{n}}\right)$. We can bound the integral in (4.44) using the uniform bounds on $\sqrt{n}(\hat{\Psi} - \hat{\Psi}^{\gamma}_Z)$ and $\left|V^{\gamma}_z - V\right|$ as

$$\begin{aligned}
(4.44) &= \int e^{t^2\xi_n/2 + t \times o_P(1)} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, \gamma) \\
&= e^{t^2\xi_n/2 + t \times o_P(1)} = e^{o_P(1)} = 1 + o_P(1).
\end{aligned}$$

Putting the pieces together, we write I_n from (4.37) as

$$I_n = \sum_{\gamma \in \mathcal{V}^{\mathbf{p}}, s} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n) I_{n,\gamma} = \sum_{\gamma \in \mathcal{V}^{\mathbf{p}}, \gamma} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n) e^{t^2 V_0/2}(1 + o_P(1)) = e^{t^2 V_0/2}(1 + o_P(1))$$

which completes the proof. □

4.6.6 Proof of Theorem 32

For our proof for Theorem 32, the analysis is locally conducted on the set

$$A^M_n = \{f \in \mathcal{F}(L) : \|f - f_0\|_L \leq M_n \xi_n\} \tag{4.45}$$

with $\xi_n = n^{-\alpha/(2\alpha+p)} \log^{\delta}(n)$ for some $M > 0$ and $\delta > 0$. And from the results in Theorem 37, we know $\Pi(A_n^M \mid Y^{(n)}) = 1 + o_p(1)$ for any $M_n \to \infty$.

Conditioning on A_n in (4.45), the posterior consists of a mixture of laws conditional on N, s and γ

$$\begin{aligned} I_n &= \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n] \\ &= \sum_{N=1}^{\infty} \Pi(N \mid \boldsymbol{Y}^{(n)}, A_n) \sum_{s=1}^{T} \Pi(s \mid \boldsymbol{Y}^{(n)}, A_n, N) \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n, N, s) I_{n,s,\gamma} \\ &= \sum_{N=1}^{N_n} \Pi(N \mid \boldsymbol{Y}^{(n)}, A_n) \sum_{s=1}^{s_n} \pi(s \mid \boldsymbol{Y}^{(n)}, A_n, N) \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n, N, s) I_{n,s,\gamma} + o_p(1) \end{aligned}$$

where we denote with

$$I_{n,s,\gamma} = \mathbb{E}^{\Pi}[e^{t\sqrt{n}(\Psi(f)-\hat{\Psi})} \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma].$$

The second equality follows from the fact that $\Pi(N > N_n \mid \boldsymbol{Y}^{(n)}) \to 0$ and $\Pi(s > s_n \mid \boldsymbol{Y}^{(n)}) \to 0$ in $\mathbb{P}_0^n$ probability as $n \to \infty$, using Corollary 6.1 of Polson and Ročková (2018). Thereby the set A_n eventually excludes all the deep learning mappings outside the sieve.

4.6.6.0.1 Linear functionals For $\Psi(f) = \langle a, f \rangle_L$, when $a(\cdot)$ is a constant function, following the same strategy as in the proof of Theorem 29, we have

$$I_{n,s,\gamma} = e^{t^2 \|a\|_L^2/2}(1 + o(1))$$

and thereby the BvM holds.

4.6.6.0.2 Squared L^2-norm functionals For $\Psi(f) = \|f\|_2^2$, we use the same strategy as in the proof of Theorem 31. For $\alpha \in (\frac{p}{2}, p)$, we have

$$\left\| f_{0[Z]}^{N,s,\gamma} - f_0 \right\|_L^2 \leq \|f - f_0\|_L^2 = o\left(\frac{1}{\sqrt{n}}\right) \tag{4.46}$$

here $f_{0[Z]}^{N,s,\gamma}$ denotes the projection of f_0 onto deep learning networks with a fixed sparsity and hidden structure (γ, Z) where $|\gamma| = s$ and the width equals N (similarly as in (4.33)). The inequality (4.46) holds for all f with a deep structure determined by (γ, Z).

The following arguments are similar to the proof of Theorem 31 but will be conditional on N and s. Since

$$\Pi(\{W_l, b_l\}_{l=1}^{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma) = \Pi(W_{L+1}, b_{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma, Z) d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma)$$

we can rewrite $I_{n,s,\gamma}$ as

$$\begin{aligned} I_{n,s,\gamma} &= \int \left(\int e^{t\sqrt{n}(\Psi(f) - \hat{\Psi})} d\Pi(W_{L+1}, b_{L+1} \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma, Z) \right) d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma) \\ &= (1 + o(1)) e^{2t^2 \|f_0\|_L^2} \int e^{t^2 (V_Z^{N,s,\gamma} - V_0)/2 + t\sqrt{n}(\hat{\Psi}_Z^{N,s,\gamma} - \hat{\Psi})} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma) \end{aligned}$$

where

$$\hat{\Psi}_Z^{N,s,\gamma} = \Psi(f_0) + \frac{1}{\sqrt{n}} W_n(2 f_{0[Z]}^{N,s,\gamma}), \quad V_Z^{N,s,\gamma} = 4 \left\| f_{0[Z]}^{N,s,\gamma} \right\|_L^2.$$

and the term $(1 + o(1))$ comes from similar considerations as in (4.43).

Now we need to show $I_{n,s,\gamma} \to e^{2t^2 \|f_0\|_L^2}$ for all N, s and γ in the local neighborhood A_n. In other words,

$$\sup_{N \leq N_n} \sup_{s \leq s_n} \sup_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \int e^{t^2 (V_Z^{N,s,\gamma} - V_0)/2 + t\sqrt{n}(\hat{\Psi}_Z^{N,s,\gamma} - \hat{\Psi})} d\Pi(Z \mid \boldsymbol{Y}^{(n)}, A_n, N, s, \gamma) = o_P(1). \tag{4.47}$$

Then we can write for $\alpha > p/2$

$$\begin{aligned}
\sqrt{n}(\hat{\Psi}^{N,s,\gamma} - \hat{\Psi}) &= W_n(f_{0[Z]}^{N,s,\gamma} - f_0) = o_P(1), \\
\left|V_{N,s,\gamma} - V_0\right| &= 4\left|\left\|f_{0[Z]}^{N,s,\gamma}\right\|_L^2 - \|f_0\|_L^2\right| \\
&\lesssim 2\,\|f_0\|_L \left\|f_{0[Z]}^{N,s,\gamma} - f_0\right\|_L + \left\|f_{0[Z]}^{N,s,\gamma} - f_0\right\|_L^2 \\
&\lesssim \left\|f_{0[Z]}^{N,s,\gamma} - f_0\right\|_L \leq \xi_n.
\end{aligned}$$

With $\alpha > p/2$, (4.47) is satisfied. Aggregating the sum of $I_{N,s,\gamma}$ over N, s and γ, we have

$$\begin{aligned}
I_n &= \sum_{N=1}^{N_n} \Pi(N \mid \boldsymbol{Y}^{(n)}, A_n) \sum_{s=1}^{s_n} \Pi(s \mid \boldsymbol{Y}^{(n)}, A_n, N) \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n, N, s) I_{n,s,\gamma} + o_P(1) \\
&= \sum_{N=1}^{N_n} \Pi(N \mid \boldsymbol{Y}^{(n)}, A_n) \sum_{s=1}^{s_n} \Pi(s \mid \boldsymbol{Y}^{(n)}, A_n, N) \sum_{\gamma \in \mathcal{V}^{\mathbf{p},s}} \Pi(\gamma \mid \boldsymbol{Y}^{(n)}, A_n, N, s)(1 + o(1)) e^{2t^2 \|f_0\|_L^2 + o_P(1)} + o_P(1).
\end{aligned}$$

As a result, we have $I_n \to e^{2t^2\|f_0\|_L^2}$ for all $t \in \mathbb{R}$ as $n \to \infty$, which concludes the proof for the L^2-norm functional case.

CHAPTER 5

DATA AUGMENTATION FOR BAYESIAN DEEP LEARNING

Deep Learning (DL) methods have emerged as one of the most powerful tools for functional approximation and prediction. While the representation properties of DL have been well studied, uncertainty quantification remains challenging and largely unexplored. Data augmentation techniques are a natural approach to provide uncertainty quantification and to incorporate stochastic Monte Carlo search into stochastic gradient descent (SGD) methods. The purpose of our paper is to show that training DL architectures with data augmentation leads to efficiency gains. We use the theory of scale mixtures of normals to derive data augmentation strategies for deep learning. This allows variants of the expectation-maximization and MCMC algorithms to be brought to bear on these high dimensional nonlinear deep learning models. To demonstrate our methodology, we develop data augmentation algorithms for a variety of commonly used activation functions: logit, ReLU, leaky ReLU and SVM. Our methodology is compared to traditional stochastic gradient descent with back-propagation. Our optimization procedure leads to a version of iteratively re-weighted least squares and can be implemented at scale with accelerated linear algebra methods providing substantial improvement in speed. We illustrate our methodology on a number of standard datasets. Finally, we conclude with directions for future research.

5.1 Introduction

Deep neural networks (DNNs) have become a central tool for Artificial Intelligence (AI) applications such as, image processing (ImageNet, Krizhevsky et al. (2012)), object recognition

. Adopted from Yuexi Wang, Nicholas Polson, and Vadim O Sokolov. Data augmentation for Bayesian deep learning. *Bayesian Analysis*, 1(1):1–29, 2022b.

(ResNet, He et al. (2016)) and game intelligence (AlphaGoZero, Silver et al. (2016)). The approximability (Poggio et al., 2017; Bauer and Kohler, 2019) and rate of convergence of deep learning, either in the frequentist fashion (Schmidt-Hieber, 2020) or from a Bayesian predictive point of view (Polson and Ročková, 2018; Wang and Ročková, 2020), have been well-explored and understood. Fan et al. (2021) provides a selective overview of deep learning. However, training deep learners is challenging due to the high dimensional search space and the non-convex objective function. Deep neural networks have also suffered from issues such as local traps, miscalibration and overfitting. Various efforts have been made to improve the generalization performance and many of their roots lie in Bayesian modeling. For example, Dropout (Wager et al., 2013) is commonly used and can be viewed as a deterministic ridge ℓ_2 regularization. Sparsity structure via spike-and-slab priors (Polson and Ročková, 2018) on weights helps DNNs adapt to smoothness and avoid overfitting. Rezende et al. (2014) propose stochastic back-propagation through the use of latent Gaussian variables.

In this paper, following the spirit of hierarchical Bayesian modeling, we develop data augmentation strategies for deep learning with a complete data likelihood function equivalent to weighted least squares regression. By using the theory of mean-variance mixtures of Gaussians, our latent variable representation brings all of the conditionally linear model theory to deep learning. For example, it allows for the straightforward specification of uncertainty at each layer of deep learning and for a wide range of regularization penalties. Our method applies to commonly used activation functions such as ReLU, leaky ReLU, logit (see also Gan et al. (2015)), and provides a general framework for training and inference in DNNs. It inherits the advantages and disadvantages of data augmentation schemes. For *approximation* methods like Expectation-Maximization (EM) and Minorize-Maximization (MM), they are stable as they increase the objective but can be slow in the neighborhood of the maximum point even with acceleration methods such as Nesterov acceleration available and the performance is highly dependent on the properties of the objective function. Stochastic

exploratory methods like MCMC have the main advantage of addressing uncertainty quantification (UQ) and are stable in the sense they require no tuning. Hyper-parameter estimation is immediately available using traditional Bayesian methods. DA augments the objective function with extra hidden units which allow for efficient step size selection for the gradient descent search. In some of the applications, data augmentation methods can be formulated in terms of complete data sufficient statistics, a considerable advantage when dealing with large datasets where most of the computational expense comes from repeatedly iterating over the data. By combing the MCMC methods with the J-copies trick (Jacquier et al., 2007), we can move faster towards posterior mode and avoid local maxima. Traditional methods for training deep learning models such as stochastic gradient descent (SGD) have none of the above advantages. We also note that we exploit the advantages of SGD and accelerated linear algebra methods when we implement our weighted least squares regression step.

Data augmentation strategies are commonplace in statistical algorithms and accelerated convergence (Nesterov, 1983; Green, 1984) is available. Our goal is to show similar efficiency improvements for deep learning. Our work builds on Deng et al. (2019) who provide adaptive empirical Bayes methods. In particular, we show how to implement standard activation functions, including ReLU (Polson and Ročková, 2018), logistic (Zhou et al., 2012; Hernández-Lobato and Adams, 2015) and SVM (Mallick et al., 2005) activation functions and provide specific data augmentation strategies and algorithms. The core subroutine of the resulting algorithms solves a least squares problem. Scalable linear algebra libraries such as Compute Unified Device Architecture (CUDA) and accelerated linear algebra (XLA) are available for implementation. To illustrate our approach, empirically we experiment with two benchmark datasets using Pólya-Gamma data augmentation for logit activation functions. For the deep architecture embedded in our approach, we adopt deep ReLU networks. Deep networks are able to achieve the same level of approximation accuracy with exponentially fewer parameters for compositional functions (Mhaskar et al., 2017). Poggio et al. (2017)

further show how deep networks can avoid the curse of dimensionality. The ReLU function is favored due to its ability to avoid vanishing gradients and its expressibility and inherent sparsity. Approximation properties of deep ReLU networks have been developed in Montufar et al. (2014), Telgarsky (2017), and Liang and Srikant (2017). Yarotsky (2017) and Schmidt-Hieber (2020) show that deep ReLU networks can yield a rate-optimal approximation of smooth functions of an arbitrary order. Polson and Ročková (2018) provide posterior rates of convergence for sparse deep learning.

There is another active area of research that revives traditional statistical models with the computational power of DL (Bhadra et al., 2021). Examples include Gaussian Process models (Higdon et al., 2008; Gramacy and Lee, 2008b), Generalized Linear Models (GLM) and Generalized Linear Mixed Models (GLMM) (Tran et al., 2020) and Partial Least Squares (PLS) (Polson et al., 2021). Our method benefits from the computation efficiency and flexibility of expression of the deep neural network. In addition, our work builds on the sampling optimization literature (Pincus, 1968, 1970) which now uses MCMC methods. Other examples include Ma et al. (2019) who study that sampling can be faster than optimization and Neelakantan et al. (2017) showing that gradient noise can improve learning for very deep networks. Gan et al. (2015) implements data augmentation inside learning deep sigmoid belief networks. Neal (2011) and Chen et al. (2014) provide Hamitonian Monte Carlo (HMC) algorithms for MCMC. Duan et al. (2018) proposes a family of calibrated data-augmentation algorithms to increase the effective sample size.

The rest of our paper is outlined as follows. Section 5.2 provides the general setting of deep neural networks and shows how DA can be integrated into deep learning using the duality between Bayesian simulation and optimization. Section 5.3 describes our data augmentation (DA) schemes and two approaches to implement them. Section 5.4 provides applications to Gaussian regression, support vector machines and logistic regression using Pólya-Gamma augmentation (Polson et al., 2013). Section 5.5 provides the experiments of

DA on both regression and classification problems. Section 5.6 concludes with directions for future research.

5.2 Bayesian Deep Learning

In deep learning we wish to recover a multivariate predictive map $f_{\boldsymbol{\theta}}(\cdot)$ denoted by

$$\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x}),$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)', y_i \in \mathbb{R}$ denotes a univariate output and $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$, $\boldsymbol{x}_i \in \mathbb{R}^p$ a high-dimensional set of inputs. Using training data of input-output pairs $\{y_i, \boldsymbol{x}_i\}_{i=1}^n$ that generalizes well out-of-sample, the goal is to provide a predictive rule for a new input variable $\boldsymbol{x}_\star$

$$y_\star = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_\star),$$

where $\hat{\boldsymbol{\theta}}$ is estimated from training data typically using SGD. The interest in deep learners lies in their ability to perform better than the additive rule for those interpolation or prediction problems. Other statistical alternatives include Gaussian processes but they often have difficulty in handling higher dimensions.

Deep learners use compositions (Kolmogorov, 1957; Vitushkin, 1964) of ridge functions rather than additive functions that are commonplace in statistical applications. With $L \in \mathbb{N}$ we denote the number of hidden layers and with $p_l \in \mathbb{N}$ the number of neurons at the l^{th} layer. Setting $p_{L+1} = p, p_0 = p_1 = 1$, we denote with $\mathbf{p} = (p_0, p_1, \ldots, p_{L+1}) \in \mathbb{N}^{L+2}$ the vector of neuron counts for the entire network. Imagine composing L layers, a deep predictor then takes the form

$$\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = (f_{W_0,b_0} \circ f_{W_1,b_1} \circ \cdots \circ f_{W_L,b_L})(\boldsymbol{x}), \tag{5.1}$$

where $b_l \in \mathbb{R}^{p_l}$ is a shift vector, $W_l \in \mathbb{R}^{p_{l-1} \times p_l}$ is a weight matrix that links neurons between $(l-1)^{th}$ and l^{th} layers and $f_{W_l,b_l}(x) = f_l(W_l x + b_l)$ is a semi-affine function. We denote with $\boldsymbol{\theta} = \{(W_0, b_0), (W_1, b_1), \ldots, (W_L, b_L)\}$ as the stacked parameters. We can rewrite the compositions in (5.1) with a set of latent variables $Z = (Z_1, Z_2, \ldots, Z_L)'$ as

$$\begin{aligned} \boldsymbol{y} &= f_0(Z_1 W_0 + b_0), \\ Z_l &= f_l(Z_{l+1} W_l + b_l), \quad l = 1, \ldots, L, \\ Z_{L+1} &= \boldsymbol{x}, \end{aligned} \tag{5.2}$$

where $Z_l \in \mathbb{R}^{n \times p_l}$ is the matrix of hidden nodes in l-th layer. We only consider the case $p = 1$ and $Z_1 \in \mathbb{R}^n$ in our work. We provide discussion on extensions to cases $p > 1$ for some of our applications in Section 5.4.

5.2.1 Bayesian Simulation and Regularization Duality

The problem of deep learning regularization (Polson and Sokolov, 2017) is to find a set of parameters $\boldsymbol{\theta}$ which minimizes a combination of a negative log-likelihood $\ell(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ and a penalty function $\phi(\boldsymbol{\theta})$ defined by

$$\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \sum_{j=1}^{\#\boldsymbol{\theta}} \phi(\theta_j), \tag{5.3}$$

where λ controls regularization and $\#\boldsymbol{\theta}$ denotes the number of parameters in $\boldsymbol{\theta}$.

When the function $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is a deep learner defined as (5.1), we can specify different amount of penalty λ_l and form of regularization function $\phi_l(\cdot)$ for each layer. Then the objective function can be written as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i)) + \sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l). \tag{5.4}$$

Commonly used regularization techniques for deep learners include L^2(weight decay), spike-and-slab regularization (Polson and Ročková, 2018) and dropout (Wager et al., 2013), which can also be viewed as a variant of L^2-regularization.

As such the optimization problem in (5.4) of training a deep learner $f_{\boldsymbol{\theta}}(\cdot)$ involves a highly nonlinear objective function. Stochastic gradient descent (SGD) is a popular tool based on back-propagation (a.k.a. the chain rule), but it often suffers from local traps and overfitting due to the non-convex nature of the problem. We propose data augmentation techniques which can be seamlessly applied in this context and provide efficiency gains. This is achieved via the hierarchical duality between optimization with regularization and finding the *maximum a posteriori* (MAP) estimate (Polson and Scott, 2011), as described in the following lemma.

Lemma 39. *The regularization problem*

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i)) + \sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l) \right\}$$

is equivalent to finding the the Bayesian MAP estimator defined by

$$\arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}} \exp \left\{ -\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i)) - \sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l) \right\},$$

which corresponds to the mode of a posterior distribution characterized as

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) / p(\boldsymbol{y}),$$

$$p(\boldsymbol{y}|\boldsymbol{\theta}) \propto \exp\{-\sum_{i=1}^{n} \ell\big(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big)\}, \quad p(\boldsymbol{\theta}) \propto \exp\{-\sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l)\}.$$

Here $p(\boldsymbol{\theta})$ can be interpreted as a prior probability distribution and the log-prior as the regularization penalty.

5.2.2 A Stochastic Top Layer

By exploiting the duality from Lemma 39, we wish to use a Bayesian framework to add stochastic layers – so as to fully account for the uncertainty in estimating the predictive rule $f_{\boldsymbol{\theta}}(\cdot)$. Thus, we convert the sequence of composite functions in the deep learner specified in (5.2) to a stochastic version given by

$$\begin{aligned} \boldsymbol{y} \mid Z_1 &\sim p(\boldsymbol{y} \mid Z_1), \\ Z_l &\sim N(f_l(W_l Z_{l+1} + b_l), \tau_l^2), \quad l = 1, 2, \ldots, L, \\ Z_{L+1} &= \boldsymbol{x}. \end{aligned} \tag{5.5}$$

Now the hidden variables $Z = (Z_1, \ldots, Z_L)'$ can be viewed as data augmentation variables and hence will also allow the contribution of fast scalable algorithms for inference and prediction.

For the ease of computation, we only replace the top layer of the DNN with a stochastic layer. We denote network structure below the top layer with $\boldsymbol{B} = \{(W_1, b_1), \ldots, (W_L, b_L)\}$, and the network structure can be rewritten as

$$\boldsymbol{y} = f_0(Z_1 W_0 + b_0), \quad Z_1 = f_{\boldsymbol{B}}(\boldsymbol{x}),$$

where the function $f_0(Z_1 W_0 + b_0)$ is the top layer structure and function $f_{\boldsymbol{B}}(\boldsymbol{x})$ is the network architecture below the top layer. Considering the objective function in (5.4), we implement the solutions with a two-step iterative search. At iteration t, we have

1. DA-update for the top layer W_0, b_0 as the MAP estimator of the distribution

$$\begin{aligned} p(W_0, b_0 \mid Z_1^{(t)}, \boldsymbol{y}) &\propto p(\boldsymbol{y}, Z_1^{(t)} \mid W_0, b_0) p(W_0, b_0) \\ &\propto \exp\left\{ -\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\boldsymbol{\theta}}(x_i) \mid \boldsymbol{B}^{(t)}) + \lambda_0 \phi_0(W_0, b_0) \right\} \end{aligned} \tag{5.6}$$

2. SGD-update for the deep architecture $\boldsymbol{B}$

$$\begin{aligned}\boldsymbol{B}^{(t+1)} &= \arg\min_{\boldsymbol{B}} \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, f_{\boldsymbol{\theta}}(x_i) \mid (W_0, b_0)^{(t+1)}\big) + \sum_{l=1}^{L} \lambda_l \phi_l(W_l, b_l) \\ &= \arg\min_{\boldsymbol{B}} \frac{1}{n}\sum_{i=1}^{n} \ell\big(Z_1^{(t)}, f_{\boldsymbol{B}}(x_i)\big) + \sum_{l=1}^{L} \lambda_l \phi_l(W_l, b_l).\end{aligned}$$

3. Sample $Z_1^{(t+1)}$ from a normal distribution $\mathcal{N}(\mu_z^{(t)}, \sigma_z^{(t)})$ where $\mu_z^{(t)}$ and $\sigma_z^{(t)}$ are determined jointly by $\{\theta^{(t)}, \boldsymbol{x}, \boldsymbol{y}\}$.

The main contribution of our work comes from two aspects: (1) we update top layer weights $\{W_0, b_0\}$ conditional on $\boldsymbol{B}$ as in (5.6), which is also equivalent to conditioning on Z_1, with data augmentation techniques as later shown in Section 5.3; (2) the latent variables Z_1 is sampled from a normal distribution rather than optimized by gradient descent methods. Z_1 serves as a bridge that connects a weighted L^2-norm model f_0 and a deep learner $f_{\boldsymbol{B}}$. Commonly used activation functions $\{f_l\}_{l=1}^{L}$ are linear affine functions, rectified linear units (ReLU), sigmoid, hyperbolic tangent (tanh), and etc. We illustrate our methods with a deep ReLU network, i.e., $\{f_l\}_{l=1}^{L}$ are ReLU functions, due to its expressibility and inherent sparsity. In the next section, we introduce our data augmentation strategies and show how the stochastic layers can be achieved via data augmentation.

5.3 Data Augmentation for Deep Learning

Data augmentation introduces a vector of auxiliary variables, denoted by $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)'$ with $\omega_i \in \mathbb{R}$, such that the posterior can be written as

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = E_{\boldsymbol{\omega}}\Big[p(\boldsymbol{\theta}, \boldsymbol{\omega} \mid \boldsymbol{y})\Big],$$

where the augmented auxiliary distribution, $p(\boldsymbol{\theta}, \boldsymbol{\omega} \mid \boldsymbol{y})$ factorizes nicely into complete conditionals $p(\boldsymbol{\theta} \mid \boldsymbol{\omega}, \boldsymbol{y})$ and $p(\boldsymbol{\omega} \mid \boldsymbol{\theta}, \boldsymbol{y})$. A crucial ingredient is that $p(\boldsymbol{\theta} \mid \boldsymbol{\omega}, \boldsymbol{y})$ is easily managed typically via conditional Gaussians.

Data augmentation tricks allow us to express the likelihood as an expectation of a weighted L^2-norm. Specifically, we write

$$\begin{aligned}\exp\Big\{ -\ell\big(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\boldsymbol{x})\big)\Big\} &= E_{\omega}\Big\{ \exp\Big(-Q\big(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega}\big)\Big)\Big\} \\ &= \int_0^{\infty} \exp\Big(-Q\big(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega}\big)\Big) p(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}\end{aligned}$$

where $p(\boldsymbol{\omega})$ is the prior on the auxiliary variables $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)'$ and the function $Q\big(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega}\big)$ is designed to be a quadratic form, given the data augmentation variables $\boldsymbol{\omega}$. The function $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = (f_0 \circ \cdots \circ f_L)(\boldsymbol{x})$ is a deep learner.

Table 5.1 shows that standard activation functions such as ReLU, logit, lasso and check can be expressed in the form of (5.7). Commonly used activation functions for deep learning, with an appropriate stochastic assumptions for w (for notation of simplicity, we derive the standard form for the single observation case) can be expressed as

$$\begin{aligned}\exp(-\max(1-x, 0)) &= E_{\omega}\left\{ \frac{1}{\sqrt{2\pi\omega}} \exp\Big(-\frac{1}{2\omega}(x-1-\omega)^2\Big)\right\}, && \text{where } \omega \sim \mathcal{GIG}(1,0,0), \\ \exp(-\log(1+e^x)) &= E_{\omega}\left\{ \exp(-\frac{1}{2}\omega x^2)\right\}, && \text{where } \omega \sim \mathcal{PG}(1,0), \\ \exp(-|x|) &= E_{\omega}\left\{ \frac{1}{\sqrt{2\pi\omega}} \exp\Big(-\frac{1}{2\omega}x^2\Big)\right\}, && \text{where } \omega \sim \mathcal{E}\big(\frac{1}{2}\big).\end{aligned}$$

Here $\mathcal{GIG}$ denotes the Generalized Inverse Gaussian distribution, $\mathcal{PG}$ represents the Pólya Gamma distribution (Polson et al., 2013), and $\mathcal{E}$ represents the exponential distribution.

Using the data augmentation strategies, the objectives are represented as mixtures of Gaussians. DA can perform such an optimization with only the use of a sequence of iteratively re-weighted L^2-norms. This allows us to use XLA techniques to accelerate the training.

$l(W,b)$	$Q(W,b,\omega)$	$p(\omega)$
ReLU: $\max(1-z_i,0)$	$\int_0^\infty \frac{1}{\sqrt{2\pi c\lambda}} \exp\left\{-\frac{(x+a\lambda)^2}{2c\lambda}\right\} \mathrm{d}\lambda = \frac{1}{a}\exp\left(-\frac{2\max(ax,0)}{c}\right)$	$\mathcal{GIG}(1,0,0)$
Logit: $\log(1+e^{z_i})$	$\frac{1}{2^b} e^{(a-b/2)\psi} \int_0^\infty e^{-\omega\psi^2/2} p(\omega)\mathrm{d}\omega = \frac{(e^\psi)^a}{(1+e^\psi)^b}$	$\mathcal{PG}(b,0)$
Lasso: $\lvert\frac{z_i}{\sigma}\rvert$	$\int_0^\infty \frac{1}{\sqrt{2\pi c\lambda}} \exp\left\{-\frac{x^2}{2c\lambda}\right\} e^{-\frac{1}{2}\lambda}\mathrm{d}\lambda = \frac{1}{c}\exp\left(-\frac{\lvert x\rvert}{c}\right)$	$\mathcal{E}(\frac{1}{2})$
Check: $\lvert z_i\rvert + (2\tau-1)z_i$	$\int_0^\infty \frac{1}{\sqrt{2\pi c\lambda}} \exp\left\{-\frac{(x+(2\tau-1)\lambda)^2}{2c^2\lambda}\right\} e^{-2\tau(1-\tau)\lambda}\mathrm{d}\lambda = \frac{1}{c}\exp\left(-\frac{2}{c}\rho_\tau(x)\right)$	$\mathcal{GIG}(1,0,2\sqrt{\tau-\tau^2})$

x

Table 5.1: Data augmentation strategies. Here $\rho_\tau(x) = \frac{1}{2}|x| + \left(\tau - \frac{1}{2}\right)x$ is the check function.

Remark 40. *The log-posterior is optimized given the training data, $\{y_i, \boldsymbol{x}_i\}_{i=1}^n$. Deep learning possesses the key property that $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}|\theta, \boldsymbol{x})$ is computationally inexpensive to evaluate using tensor methods for very complicated architectures and fast implementation on large datasets. One caveat is that the posterior is highly multi-modal and providing good hyperparameter tuning can be expensive. This is clearly a fruitful area of research for state-of-the-art stochastic MCMC algorithms to provide more efficient algorithms. For shallow architectures, the alternating direction method of multipliers (ADMM) is an efficient solution to the optimization problem.*

Similarly we can represent the regularization penalty $\exp(-\lambda\phi(\boldsymbol{\theta}))$ in data augmentation form. Hence, we can then replace the optimization problem in (5.4) with

$$\hat{\boldsymbol{\theta}} := \arg\max_{\boldsymbol{\theta}} E_{\boldsymbol{\omega}}\Big[\exp\Big(-\frac{1}{n}\sum_{i=1}^n Q\big(y_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{\omega}\big) - \sum_{l=0}^L \lambda_l \phi_l(W_l, b_l)\Big)\Big], \tag{5.7}$$

using the duality in Lemma 39.

There are two approaches to Monte Carlo optimization which could handle our data augmentation (Geyer, 1996), missing data methods like Expectation-Maximization (EM) algorithms or stochastic search methods like Markov Chain Monte Carlo (MCMC). The first approach is based on a probabilistic *approximation* of the objective function (5.7) and is less

concerned with exploring $\boldsymbol{\Theta}$. The second type is more *exploratory* which aims to optimize the objective function by visiting the entire range of $\boldsymbol{\Theta}$ and is less tied to the properties of the function.

For EM algorithms, we consider constructing a surrogate optimization problem which has the same solution to (5.7) (Lange et al., 2000). Specifically, we define a new objective function as

$$H(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\omega}|\boldsymbol{\theta}}\Big[\exp\Big(-\frac{1}{n}\sum_{i=1}^{n} Q\big(y_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{\omega}\big) - \sum_{l=0}^{L}\lambda_l\phi_l(W_l, b_l)\Big)\Big],$$

which is a concave function to be maximized. A natural choice of the surrogate function can be constructed using Jensen's inequality as

$$G\big(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\big) = -\mathbb{E}_{\boldsymbol{\omega}|\boldsymbol{\theta}^{(t)}}\left[\frac{1}{n}\sum_{i=1}^{n} Q\big(y_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{\omega}\big) + \sum_{l=0}^{L}\lambda_l\phi_l(W_l, b_l)\right],$$

where each ω_i is drawn from conditional distribution $p(\omega_i \mid \boldsymbol{\theta}) \propto p(\omega_i, \boldsymbol{\theta})$ and the minorization is satisfied as

$$\log H(\boldsymbol{\theta}) \geq G\big(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\big).$$

Maximizing $G\big(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\big)$ with respect to $\boldsymbol{\theta}$ drives $H(\boldsymbol{\theta})$ uphill. The ascent property of the EM algorithm relies on the nonnegativity of the Kullback-Leibler divergence of two conditional probability densities (Hunter and Lange, 2004; Lange, 2013a). The EM algorithm enjoys the numerical stability as it steadily increases the likelihood without wildly overshooting or undershooting. It simplifies the optimization problem by (1) avoiding large matrix inversion; (2) linearizing the objective function; (3) separating the variables of the optimization problem (Lange, 2013b). In Section 5.4.3 we show how Pólya-Gamma augmentation leads to an EM algorithm for logistic regression.

The exploratory alternative to solve (5.7) is stochastic search methods such as MCMC.

The data augmentation strategies enable us to sample from the joint posterior

$$\begin{aligned} p(\boldsymbol{\theta} \mid \boldsymbol{y}) &\propto \exp\Big(-\frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big) - \sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l)\Big) \\ &= E_{\boldsymbol{\omega}}\Big[\exp\Big(-\frac{1}{n}\sum_{i=1}^{n} Q\big(y_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{\omega}\big) - \sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l)\Big)\Big] \\ &= \int_0^{\infty} \exp\Big(-\frac{1}{n}\sum_{i=1}^{n} Q(y_i \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{\omega})\Big) p(\boldsymbol{\omega}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\omega} \end{aligned}$$

where the prior is related to the regularization penalty, via $p(\boldsymbol{\theta}) \propto \exp\big(-\sum_{l=0}^{L} \lambda_l \phi_l(W_l, b_l)\big)$.

Hence, we can provide an MCMC algorithm in the augmented space $(\boldsymbol{\theta}, \boldsymbol{\omega})$ and simulate from the joint posterior distribution, denoted by $p(\boldsymbol{\theta}, \boldsymbol{\omega} \mid \boldsymbol{y})$, namely

$$p(\boldsymbol{\theta}, \boldsymbol{\omega} \mid \boldsymbol{y}) \propto \exp\Big(- Q(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega})\Big) p(\boldsymbol{\theta}) p(\boldsymbol{\omega}).$$

A sequence can be simulated using MCMC Gibbs conditionals,

$$\begin{aligned} p\big(\boldsymbol{\theta}^{(t)} \mid \boldsymbol{\omega}^{(t)}, \boldsymbol{y}\big) &\propto \exp\Big(- Q(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega}^{(t)})\Big) p(\boldsymbol{\theta}), \\ p\big(\boldsymbol{\omega}^{(t+1)} \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{y}\big) &\propto \exp\Big(- Q(\boldsymbol{y} \mid f_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}), \boldsymbol{\omega})\Big) p(\boldsymbol{\omega}). \end{aligned}$$

Then we recover stochastic draws $\boldsymbol{\theta}^{(t)} \sim p(\boldsymbol{\theta} \mid \boldsymbol{y})$ from the marginal posterior. These draws can be used in prediction to account for *predictive uncertainty*, namely

$$p\big(y_\star \mid f(\boldsymbol{x}_\star)\big) = \int p\big(y_\star \mid \boldsymbol{\theta}, f_{\boldsymbol{\theta}}(\boldsymbol{x}_\star)\big) p(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\theta} \approx \frac{1}{T}\sum_{t=1}^{T} p\big(y_\star \mid \boldsymbol{\theta}^{(t)}, f_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}_\star)\big). \tag{5.8}$$

As $Q(\boldsymbol{y} \mid f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{\omega})$ is conditionally quadratic, the update step for $\boldsymbol{\theta} \mid \boldsymbol{\omega}, \boldsymbol{y}$ can be achieved using SGD or a weighted L^2-norm – the weights $\boldsymbol{\omega}$ are adaptive and provide an automatic choice of the learning rate, thus avoiding backtracking which can be computationally ex-

pensive. And the performance of MCMC search is less tied to the statistical properties (i.e. convexity or concavity) of the objective function. We provide examples of how Gaussian regression and SVMs can be implements in Section 5.4.1 and Section 5.4.2.

5.3.1 MCMC with J-copies

The MCMC methods offer a full description of the objective function (5.7) over the entire space $\boldsymbol{\Theta}$. Inspired by the simulated annealing algorithm (Metropolis et al., 1953), we introduce a scaling factor J to allow for faster moves on the surface of (5.7) to maximize. It also helps avoiding the trapping attraction of local maxima. In addition, the corresponding posterior is connected to the Boltzmann distribution, whose density is prescribed by the energy potential $f(\theta)$ and temperature J as

$$\pi_J(\boldsymbol{\theta}) = \exp\left\{-Jf(\boldsymbol{\theta})\right\}/Z_J \quad \text{for} \quad \boldsymbol{\theta} \in \boldsymbol{\Theta} \tag{5.9}$$

where $Z_J = \int_{\boldsymbol{\Theta}} \exp\left\{-Jf(\boldsymbol{\theta})\right\} d\boldsymbol{\theta}$ is an appropriate normalizing constant.

To simulate the posterior mode without evaluating the likelihood directly (Jacquier et al., 2007), we sample J independent copies of hidden variable Z_1 . Denoted the copies with $Z_1^1, \ldots, Z_1^J$, we sample them simultaneously and independently from the posterior distribution

$$Z_1^j | \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y} \overset{iid}{\sim} \mathcal{N}(\mu_z, \sigma_z^2), \quad j = 1, \ldots, J,$$

where μ_z, σ_z are determined by $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}\}$. And we stack the J copies as

$$\boldsymbol{y}^{(S)} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y} \\ \boldsymbol{y} \\ \vdots \\ \boldsymbol{y} \end{bmatrix}, \quad Z_1^{(S)} = \begin{bmatrix} Z_1^1 \\ Z_1^2 \\ Z_1^3 \\ \vdots \\ Z_1^J \end{bmatrix}, \quad f_{\boldsymbol{B}}(\boldsymbol{x}^{(S)}) = \begin{bmatrix} f_{\boldsymbol{B}}(\boldsymbol{x}) \\ f_{\boldsymbol{B}}(\boldsymbol{x}) \\ f_{\boldsymbol{B}}(\boldsymbol{x}) \\ \vdots \\ f_{\boldsymbol{B}}(\boldsymbol{x}) \end{bmatrix} \tag{5.10}$$

where $\boldsymbol{y}^{(S)}$, $Z_1^{(S)}$ and $f_{\boldsymbol{B}}(\boldsymbol{x}^{(S)})$ are $(n \times J)$-dimensional vectors. We use $Z_1^{(S)}$ to amplify the information in $\boldsymbol{y}$, which is especially useful in the finite sample problems. Figure 5.1 illustrates our network architecture.

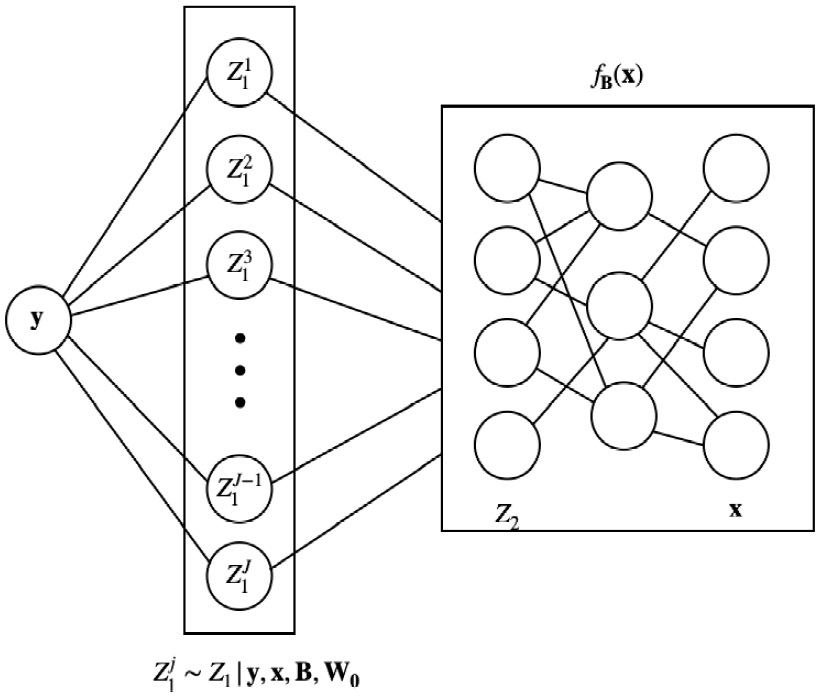


Figure 5.1: J-copies network architecture

With the stacked system, the joint distribution of the parameters $\boldsymbol{\theta}$ and the augmented

hidden variables $Z_1^{(S)}$ given data $\boldsymbol{y}, \boldsymbol{x}$ can be written as

$$\pi_J(\boldsymbol{\theta}, Z_1^{(S)} \mid \boldsymbol{x}, \boldsymbol{y}) \propto \prod_{j=1}^{J} p(\boldsymbol{y} \mid \boldsymbol{\theta}, Z_1^j) p(Z_1^j \mid \boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) p(\theta).$$

Hence, the marginal joint posterior

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y}) = \int \pi_J(\boldsymbol{\theta}, Z_1^{(S)} \mid \boldsymbol{x}, \boldsymbol{y}) \mathrm{d}Z_1^{(S)}$$

concentrates on the density proportional to $p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta})^J p(\boldsymbol{\theta})$ and provides us with a simulation solution to finding the MAP estimator (Pincus, 1968, 1970).

Another alternative to simulate from the posterior mode is Hamiltonian Monte Carlo (Neal, 2011), which is a modification of Metropolis-Hastings (MH) sampler. Adding an additional momentum variable $\boldsymbol{\nu}$ to the Boltzmann distribution in (5.9), and generating draws from joint distribution

$$\pi_J(\boldsymbol{\theta}, \boldsymbol{\omega}) \propto \exp\left(-Jf(\boldsymbol{\theta}) - (1/2)\boldsymbol{\nu}^T M^{-1} \boldsymbol{\nu}\right),$$

where M is a mass matrix. Chen et al. (2014) adopt this approach in a deep learning setting.

5.3.2 Connection to Diffusion Theory

An alternative to the MCMC algorithm can be derived from diffusion theory (Phillips and Smith, 1996). For example, we can approximate the random walk Metropolis-Hastings algorithm with the Langevin diffusion L_t defined by the stochastic differential equation $dL_t = dB_t + \frac{1}{2}\nabla \log f(L_t)dt$, where B_t is the standard Brownian motion. More specifically, let $d := |\boldsymbol{\theta}|$, we write the random walk like transition as

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \frac{\sigma^2}{2}\nabla \log f(\boldsymbol{\theta}^{(t)}) + \sigma \boldsymbol{\epsilon}_t,$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}_d(0, I_d)$ and σ^2 corresponds to the discretization size.

This can also be derived by taking a second-order approximation of $\log(f)$, namely

$$\begin{aligned}\log f(\boldsymbol{\theta}^{(t+1)}) = &\log f(\boldsymbol{\theta}^{(t)}) + \big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big)'\nabla \log f(\boldsymbol{\theta}^{(t)}) \\ &- \frac{1}{2}\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big)' H(\boldsymbol{\theta}^{(t)})\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big),\end{aligned}$$

where $H(\boldsymbol{\theta}^{(t)}) = -\nabla^2 \log f(\boldsymbol{\theta}^{(t)})$ is the Hessian matrix. By taking exponential transformation on both sides, the random walk type approximation to $f(\boldsymbol{\theta}^{(t+1)})$ is

$$\begin{aligned}f(\boldsymbol{\theta}^{(t+1)}) &\propto \exp\Big\{\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big)'\nabla \log f(\boldsymbol{\theta}^{(t)}) - \frac{1}{2}\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big)' H(\boldsymbol{\theta}^{(t)})\big(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\big)\Big\} \\ &\propto \exp\Big\{-\frac{1}{2}\big(\boldsymbol{\theta}^{(t+1)} - \widetilde{\boldsymbol{\theta}}^{(t)}\big)' H(\boldsymbol{\theta}^{(t)})\big(\boldsymbol{\theta}^{(t+1)} - \widetilde{\boldsymbol{\theta}}^{(t)}\big)\Big\}.\end{aligned}$$

where $\widetilde{\boldsymbol{\theta}}^{(t)} = \boldsymbol{\theta}^{(t)} + H^{-1}(\boldsymbol{\theta}^{(t)})\nabla \log f(\boldsymbol{\theta}^{(t)})$. If we simplify this approximation by replacing $H(\boldsymbol{\theta}^{(t)})$ with $\sigma^{-2} I_p$, the Taylor approximation leads to updating step as

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \sigma^2 \nabla \log f(\boldsymbol{\theta}^{(t)}) + \sigma \boldsymbol{\epsilon}_t.$$

Roberts and Rosenthal (1998) give further discussion on the choice of σ that would yield an acceptance rate of 0.574 to achieve optimal convergence rate.

Mandt et al. (2017) show that SGD can be interpreted as a multivariate Ornstein-Uhlenbeck process

$$d\boldsymbol{\theta}^{(t)} = -\eta A \boldsymbol{\theta}^{(t)} dt + \eta \sqrt{\frac{C}{S}} dW^{(t)},$$

here η is the constant learning rate, A is the symmetric Hessian matrix at the optimum and $\frac{C}{S}$ is the covariance of the mini-batch (of size S) gradient noise, which is assumed to be approximately constant near the local optimum of the loss. They also provide results on discrete-time dynamics on other Stochastic Gradient MCMC algorithms, such as Stochastic Gradient Langevin dynamics (SGLD) by Welling and Teh (2011) and Stochastic Gradient

Fisher Scoring by Ahn et al. (2012).

Combing their results and the Langevin dynamics of MCMC algorithms, we can write the approximation of our DA-DL updating scheme as

$$\begin{bmatrix} W_0 \\ b_0 \end{bmatrix}^{(t+1)} = \begin{bmatrix} W_0 \\ b_0 \end{bmatrix}^{(t)} + \sigma^2 \nabla \log f_0(Z_1^{(t)} W_0^{(t)} + b_0^{(t)}) + \sigma \boldsymbol{\epsilon}_{0t},$$

$$\boldsymbol{B}^{(t+1)} = \boldsymbol{B}^{(t)} - \eta \nabla^2 f_{\boldsymbol{B}^*}(\boldsymbol{x}) \boldsymbol{B}^{(t)} + \frac{C}{\sqrt{S}} \eta \boldsymbol{\epsilon}_{\boldsymbol{B}t}.$$

Similar adaptive dynamics are also observed in other methods. Geman and Hwang (1986) show the convergence of the annealing process using Langevin equations. Slice sampling (Neal, 2003) adaptively chooses the step size based on the local properties of the density function. By constructing local quadratic approximations, it could adapt to the dependencies between variables. Murray et al. (2010) further propose elliptical slice sampling that operates on the ellipse of states.

5.4 Applications

To illustrate our methodology, we provide three examples: (1) a standard Gaussian regression model with squared loss; (2) a binary classification model under the support vector machine framework; (3) a logistic regression model paired with a Pólya mixing distribution. For the Gaussian regression and SVM models, we implement with J-copies stacking strategy to provide full posterior modes.

Before diving into the examples, we introduce the notations we use throughout this section. We continue to denote the output with $\boldsymbol{y} = (y_1, \ldots, y_n)', y_i \in \mathbb{R}$, the input with $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)', \boldsymbol{x}_i \in \mathbb{R}^p$, the latent variable of the top layer with $Z_1 = (z_{1,1}, \ldots, z_{1,n})', z_{1,i} \in \mathbb{R}$ and the stacked version as in (5.10). We introduce stochastic noises $\boldsymbol{\epsilon}_0 = (\epsilon_{0,1}, \ldots, \epsilon_{0,n})'$ in the top layer and $\boldsymbol{\epsilon}_z = (\epsilon_{z,1}, \ldots, \epsilon_{z,n})'$ in the second layer, where $\epsilon_{0,i} \overset{iid}{\sim} \mathcal{N}(0, \tau_0^2)$ and

$\epsilon_{z,i} \overset{iid}{\sim} \mathcal{N}(0, \tau_z^2)$. The scale parameters τ_0 and τ_z are pre-specified and determine the level of randomness or uncertainty for the DA-update and SGD-update respectively. We use η to denote the learning rate used in the SGD updates and T is number of training epochs. We use $\|\cdot\|$ to denote ℓ_2-norm such that $\|\boldsymbol{y}\| = \sqrt{\sum_{i=1}^n y_i^2}$ and the matrix-type norm as $\|\boldsymbol{y}\|_\Sigma = \sqrt{y^T \Sigma y}$.

Our models differ from standard deep learning models and some newly proposed Bayesian approaches in the adoption of stochastic noises $\boldsymbol{\epsilon}_0$ and $\boldsymbol{\epsilon}_z$. It distinguishes our model from other deterministic neural networks. By letting $\boldsymbol{\epsilon}_z$ follow a spiky distribution that puts most of its mass around zero, we can control the estimation approximating to posterior mode instead of posterior mean. The randomness allows us to adopt a stacked system and make the best use of data especially when the dataset is small.

5.4.1 Gaussian Regression

We consider the regression model as

$$
\begin{aligned}
y_i &= z_{1,i} W_0 + b_0 + \epsilon_{0,i}, && \text{where } y_i \in (-\infty, \infty),\ \epsilon_{0,i} \overset{i.i.d}{\sim} \mathcal{N}(0, \tau_0^2), \\
z_{1,i} &= f_{\boldsymbol{B}}(x_i) + \epsilon_{z,i}, && \text{where } \epsilon_{z,i} \overset{i.i.d}{\sim} \mathcal{N}(0, \tau_z^2).
\end{aligned}
$$

The posterior updates are given by

$$
\hat{W}_0 = \mathrm{Cov}(Z_1, \boldsymbol{y}) / \mathrm{Var}(Z_1), \tag{5.11}
$$

$$
\hat{b}_0 = \bar{\boldsymbol{y}} - W_0 \bar{Z}_1, \tag{5.12}
$$

$$
p(Z_1 \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = C_z \exp\left\{ -\frac{1}{2\tau_0^2} \|\boldsymbol{y} - Z_1 W_0 - b_0\|^2 - \frac{1}{2\tau_z^2} \|Z_1 - f_{\boldsymbol{B}}(\boldsymbol{x})\|^2 \right\},
$$

where $\bar{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and C_z is a normalizing constant. The latent variable Z_1 is drawn from following normal distribution $Z_1 \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$ with the mean and variance specified as

$$\mu_Z = \frac{\tau_z^2 W_0(\boldsymbol{y} - b_0) + \tau_0^2 f_{\boldsymbol{B}}(\boldsymbol{x})}{W_0^2\tau_z^2 + \tau_0^2}, \quad \sigma_Z^2 = \frac{\tau_0^2\tau_z^2}{W_0^2\tau_z^2 + \tau_0^2}. \tag{5.13}$$

The J copies of Z_1 are simulated and stacked as

$$Z_1^j \overset{iid}{\sim} \mathcal{N}(\mu_Z, \sigma_Z^2), \quad Z_1^{(S)} = (Z_1^1, \ldots, Z_1^J)'.$$

The updating scheme for this Gaussian regression is summarized in Algorithm 7.

Algorithm 7: Data Augmentation with J-copies for Gaussian Regression (DA-GR)

Initialize $\boldsymbol{B}^{(0)}, W_0^{(0)}, b_0^{(0)}$
For epoch $t = 1, \ldots, T$ **do**

1. Update the weights in the top layer with $\{\boldsymbol{y}^{(S)}, Z_1^{(t,S)}\}$
   $W_0^{(t)} = \mathrm{Cov}(Z_1^{(t,S)}, \boldsymbol{y}^{(S)})/\mathrm{Var}(Z_1^{(t,S)})$
   $b_0^{(t)} = \bar{\boldsymbol{y}}^{(S)} - W_0^{(t)}\bar{Z_1}^{(t,S)}$
2. Update the deep learner $f_{\boldsymbol{B}}$ with $\{Z_1^{(t,S)}, \boldsymbol{x}^{(S)}\}$
   $\boldsymbol{B}^{(t)} = \boldsymbol{B}^{(t-1)} - \eta\nabla f_{\boldsymbol{B}^{(t-1)}}\big(\boldsymbol{x}^{(S)} \mid Z_1^{(t,S)}\big)$; `// SGD`
3. Update $Z_1^{(S)}$ jointly from deep learner $f_{\boldsymbol{B}}$ and sampling layer f_0
   ${Z_1}^{j,(t+1)} \mid W_0^{(t)}, b_0^{(t)}, \boldsymbol{y}, f_{\boldsymbol{B}^{(t)}}(\boldsymbol{x}) \overset{iid}{\sim} \mathcal{N}\big(\mu_z^{(t)}, {\sigma_z^{(t)}}^2\big)$, $j = 1, \ldots, J$

Return $\hat{\boldsymbol{y}} = W_0^{(T)} f_{\boldsymbol{B}^{(T)}}(\boldsymbol{x}) + b_0^{(T)}$

The model can also be generalized to multivariate y. Let y_i be a q-dimension vector, we denote each dimension as $y_{ik}, k = 1, \ldots, q$, and the model is written as

$$\begin{aligned} y_{ik} &= z_{1,i}W_{0k} + b_{0k} + \epsilon_{0,ik}, && \text{where } y_{ik} \in (-\infty, \infty),\ \epsilon_{0,ik} \overset{iid}{\sim} \mathcal{N}(0, \tau_0^2), \\ z_{1,i} &= f_{\boldsymbol{B}}(x_i) + \epsilon_{z,i}, && \text{where } \epsilon_{z,i} \overset{iid}{\sim} \mathcal{N}(0, \tau_z^2), \end{aligned}$$

where $W_0 = (W_{01}, \ldots, W_{0q})'$ is now a q-dimensional vector with W_{0k} computed similarly to (5.11), $b_0 = (b_{01}, \ldots, b_{0q})'$ is also q-dimensional with b_{0k} calculated as (5.12). The posterior

update for Z_1 becomes

$$p(Z_1 \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = C_z \exp\left\{-\frac{1}{2\tau_0^2}\sum_{k=1}^{q}\|\boldsymbol{y}_k - Z_1 W_{0k} - b_{0k}\|^2 - \frac{1}{2\tau_z^2}\|Z_1 - f_{\boldsymbol{B}}(\boldsymbol{x})\|^2\right\},$$

which is a multivariate normal distribution with the mean and variance as

$$\mu_Z = \frac{\tau_z^2\sum_{k=1}^{q} W_{0k}(\boldsymbol{y}_k - b_{0k}) + \tau_0^2 f_{\boldsymbol{B}}(\boldsymbol{x})}{\tau_z^2\sum_{k=1}^{q} W_{0k}^2 + \tau_0^2}, \sigma_Z^2 = \frac{\tau_0^2\tau_z^2}{\tau_z^2\sum_{j=k}^{q} W_{0k}^2 + \tau_0^2}.$$

5.4.2 Support Vector Machines (SVMs)

Support vector machines require data augmentation for rectified linear unit (ReLU) activation functions. Polson and Scott (2011) and Mallick et al. (2005) write the support vector machine model as

$$\boldsymbol{y} = Z_1 W_0 + \boldsymbol{\lambda} + \sqrt{\boldsymbol{\lambda}}\boldsymbol{\epsilon}_0, \text{ where } \boldsymbol{\lambda} \sim p(\boldsymbol{\lambda}),$$

where $p(\boldsymbol{\lambda})$ follows a flat uniform prior. The augmentation variable $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)'$ can be regarded as slacks admitting fuzzy boundaries between classes.

By incorporating the augmentation variable $\boldsymbol{\lambda}$, the ReLU deep learning model can be written as

$$\begin{aligned} y_i &= z_{1,i} W_0 + \lambda_i + \sqrt{\lambda_i}\epsilon_{0,i}, \text{ where } y_i \in \{-1, 1\}, \epsilon_{0,i} \overset{i.i.d}{\sim} \mathcal{N}(0, \tau_0^2), \\ z_{1,i} &= f_{\boldsymbol{B}}(\boldsymbol{x}_i) + \epsilon_{z,i}, \text{ where } \epsilon_{z,i} \overset{i.i.d}{\sim} \mathcal{N}(0, \tau_z^2). \end{aligned}$$

From a probabilistic perspective, the likelihood function for the output $\boldsymbol{y}$ is given by

$$\begin{aligned} p(y_i \mid W_0, z_{1,i}) &\propto \exp\left\{-\frac{2}{\tau_0^2}\max(1 - y_i z_{1,i} W_0, 0)\right\} \\ &\propto \int_0^\infty \frac{1}{\tau_0\sqrt{2\pi\lambda_i}} \exp\left(-\frac{1}{2\tau_0^2}\frac{(1 + \lambda_i - y_i z_{1,i} W_0)^2}{\lambda_i}\right) \mathrm{d}\lambda_i. \end{aligned}$$

Derived from this augmented likelihood function, the conditional updates are

$$W_0 \mid \boldsymbol{y}, Z_1, \boldsymbol{\lambda} \propto \left[\prod_{i=1}^{n} \frac{1}{\tau_0\sqrt{\lambda_i}}\right]\left[\exp\left\{-\frac{1}{2\tau_0^2}\sum_{i=1}^{n}\frac{(1+\lambda_i - y_i z_{1,i} W_0)^2}{\lambda_i}\right\}\right]$$

$$Z_1 \mid \boldsymbol{y}, \boldsymbol{x}, W_0, \boldsymbol{B} \propto \exp\left\{-\frac{1}{2\tau_0^2}\left\|\boldsymbol{y} - Z_1 W_0\right\|^2_{\Lambda^{-1}} - \frac{1}{2\tau_z^2}\left\|Z_1 - f_{\boldsymbol{B}}(\boldsymbol{x})\right\|^2\right\}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of the augmentation variables.

In order to generate the latent variables, we use conditional Gibbs sampling as

$$\lambda_i^{-1} \mid W_0, y_i, z_{1,i} \sim \mathcal{IG}(\left|1 - y_i z_{1,i} W_0\right|^{-1}, \tau_0^{-2}) \tag{5.14}$$

$$W_0 \mid \boldsymbol{y}, Z_1, \lambda \sim \mathcal{N}(\mu_w, \sigma_w^2) \tag{5.15}$$

$$Z_1 \mid \boldsymbol{y}, \boldsymbol{x}, W_0, \boldsymbol{B} \sim \mathcal{N}(\mu_z, \sigma_z^2) \tag{5.16}$$

with the means and variances given by

$$\mu_w = \frac{\sum_{i=1}^{n} y_i z_{1,i}\frac{1+\lambda_i}{\lambda_i}}{\tau_0^2\sum_{i=1}^{n}\frac{y_i^2 z_{1,i}^2}{\lambda_i}}, \sigma_w^2 = \frac{1}{\tau_0^2\sum_{i=1}^{n}\frac{y_i^2 z_{1,i}^2}{\lambda_i}}, \mu_z = \frac{W_0\tau_z^2\boldsymbol{y} + \tau_0^2 f_{\boldsymbol{B}}(\boldsymbol{x})\Lambda\mathbf{1}}{W_0\tau_z^2 + \tau_0^2\Lambda\mathbf{1}}, \sigma_z^2 = \frac{\tau_0^2\tau_z^2\Lambda\mathbf{1}}{W_0^2\tau_z^2 + \tau_0^2\Lambda\mathbf{1}},$$

where $\mathcal{IG}$ denotes the Inverse Gaussian distribution and $\mathbf{1} = (1, \ldots, 1)'$ is a n-dimensional unit vector.

The J-copies strategy can also be adopted here. Z_1^j and $\boldsymbol{\lambda}^j$ needs to be sampled independently for $j = 1, \ldots, J$. Algorithm 8 summarizes the updating scheme with J-copies for SVMs.

5.4.3 Logistic Regression

The aim of this example is to show how EM algorithm can be implemented via a weighted L^2-norm in deep learning. Adopting the logistic regression model from Polson and Scott

Algorithm 8: Data Augmentation with J-copies for SVM (DA-SVM)

Initialize $\boldsymbol{B}^{(0)}, W_0^{(0)}, \boldsymbol{\lambda}^{(0)}$

For epoch $t = 1, \ldots, T$ **do**

1. Update the weights in the top layer with $\{\boldsymbol{y}^{(S)}, Z_1^{(t,S)}\}$
   $\{\boldsymbol{\lambda}^{(t,S)}\}^{-1} \mid W_0^{(t-1)}, \boldsymbol{y}^{(S)}, Z_1^{(t,S)} \sim \mathcal{IG}(|1 - \boldsymbol{y}^{(S)} Z_1^{(t,S)} W_0^{(t-1)}|^{-1}, \frac{1}{\tau_0^2})$
   $W_0^{(t)} \mid \boldsymbol{y}^{(S)}, Z_1^{(t,S)}, \boldsymbol{\lambda}^{(t,S)} \sim \mathcal{N}(\mu_\omega^{(t)}, {\sigma_\omega^{(t)}}^2)$
2. Update the deep learner $f_{\boldsymbol{B}}$ with $\{Z_1^{(t,S)}, \boldsymbol{x}^{(S)}\}$
   $\boldsymbol{B}^{(t)} = \boldsymbol{B}^{(t-1)} - \eta \nabla f_{\boldsymbol{B}^{(t-1)}}\big(\boldsymbol{x}^{(S)} \mid Z_1^{(t,S)}\big)$; `// SGD`
3. Update $Z_1^{(S)}$ jointly from deep learner $f_{\boldsymbol{B}}$ and sampling layer f_0
   $Z_1^{j,(t+1)} \mid W_0^{(t)}, \boldsymbol{\lambda}^{j,(t)}, \boldsymbol{y}, f_{\boldsymbol{B}^{(t)}}(\boldsymbol{x}) \overset{iid}{\sim} \mathcal{N}(\mu_z^{(t)}, {\sigma_z^{(t)}}^2)$, $j = 1, \ldots, J$

Return $\hat{y} = \begin{cases} 1, \text{ if } W_0^{(T)} f_{\boldsymbol{B}^{(T)}}(\boldsymbol{x}) > 0 \\ -1, \text{ otherwise.} \end{cases}$

(2013), we focus on the penalization of W_0, with parameter optimization given by

$$\hat{W}_0 = \arg\min_{W_0} \left[\frac{1}{n} \sum_{i=1}^{n} \log\Big(1 + \exp\big(-y_i f_{\boldsymbol{B}}^{DL}(\boldsymbol{x}_i) W_0\big)\Big) + \phi(W_0 \mid \tau) \right],$$

The outcomes y_i are coded as ± 1, and τ is assumed fixed.

For likelihood function ℓ and regularization penalty ϕ, we assume

$$p(y_i \mid \sigma) \propto \int_0^\infty \frac{\sqrt{\omega}_i}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{\omega_i}{2\sigma^2} \big(y_i f_{\boldsymbol{B}}(\boldsymbol{x}_i) W_0 - \frac{1}{2\omega_i}\big)^2 \right\} p(\omega_i) \mathrm{d}\omega_j, \tag{5.17}$$

$$p(W_0 \mid \tau) = \int_0^\infty \frac{\sqrt{\lambda}}{\sqrt{2\pi}\tau} \exp\left\{ -\frac{\lambda}{2\tau^2} (W_0 - \mu_W - \kappa_W \lambda^{-1})^2 \right\} p(\lambda) \mathrm{d}\lambda, \tag{5.18}$$

where μ_W, κ_W are pre-specified terms controlling the prior of the penalty term and λ is endowed with a Pólya distribution prior $P(\lambda)$. Let ω_i^{-1} have a Pólya distribution with $\alpha = 1, \kappa = 1/2$, the following three updates will generate a sequence of estimates that

converges to a stationary point of posterior

$$W_0^{(t+1)} = (\tau^{-2}\Lambda^{(t)} + \boldsymbol{x}_*^T\Omega^{(t)}\boldsymbol{x}_*)^{-1}(\frac{1}{2}\boldsymbol{x}_*^T\mathbf{1}),$$
$$\omega_i^{(t+1)} = \frac{1}{z_i^{(t+1)}}\left(\frac{e^{z_i^{(t+1)}}}{1+e^{z_i^{(t+1)}}} - \frac{1}{2}\right), \lambda^{(t+1)} = \frac{\kappa_W + \tau^2\phi'(W_0^{(t)} \mid \tau)}{W_0^{(t)} - \mu_W},$$

where $z_i^{(t)} = y_i z_{1,i}^T W_0^{(t)} = y_i \text{logit}(\hat{y}_i^t)$, $\boldsymbol{x}_*$ is a matrix with rows $x_i^* = y_i z_{1,i}$, $\Omega = \text{diag}(\omega_1, \ldots, \omega_n)$ and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$ are diagonal matrices. $\boldsymbol{x}_*$ can be written as $\boldsymbol{x}_* = \text{diag}(\boldsymbol{y})Z_1$, $\phi'(\cdot)$ denotes the derivative of standard normal density function.

In the non-penalized case, with $\lambda_i = 0$ for every i, the updates can be simplified as weighted least squares

$$W_0^{(t+1)} = ({Z_1^{(t)}}^T\text{diag}(\boldsymbol{y})\Omega^{(t)}\text{diag}(\boldsymbol{y})Z_1^{(t)})^{-1}(\frac{1}{2}\boldsymbol{y}^T Z_1^{(t)}),$$
$$\omega_i^{(t+1)} = \frac{1}{z_i^{(t+1)}}\left(\frac{e^{z_i^{(t+1)}}}{1+e^{z_i^{(t+1)}}} - \frac{1}{2}\right).$$

We focus on the non-penalized binary classification case and Algorithm 9 summarizes our approach. Further generalizations are available. For example, a ridge-regression penalty, along with the generalized double-pareto prior (Armagan et al., 2013) can be implemented by adding a sample-wise L^2-regularizer. A multinomial generalization of this model can be found in Polson and Scott (2013).

5.5 Experiments

We illustrate the performance of our methods on both synthetic and real datasets, compared to the deep ReLU networks without the data augmentation layer. We refer to the latter as DL in our results. We denote the data augmented gaussian regression in Algorithm 7 as DA-GR, the SVM implementation in Algorithm 8 as DA-SVM and the logistic regression in

Algorithm 9: Data Augmentation for Logistic Regression (DA-logit)

Initialize $W_0^{(0)}, b_0^{(0)} \boldsymbol{B}^{(0)}$

For epoch $t = 1, \ldots, T$ **do**

1. Retrieve the input and output of the top layer
   $Z_1^{(t)} = f_{\boldsymbol{B}^{(t-1)}}(\boldsymbol{x})$; `// input`
   $\boldsymbol{y}^{(t)} = \text{sigmoid}(W_0^{(t-1)} Z_1^{(t)} + b_0^{(t-1)})$; `// output`
2. Calculate the sample-wise weights
   $\boldsymbol{z}^{(t)} = \boldsymbol{y} \cdot \text{logit}(\boldsymbol{y}^{(t)})$; `// transformed responses`
   $\boldsymbol{\omega}^{(t)} = \frac{1}{\boldsymbol{z}^{(t)}}(\text{sigmoid}(\boldsymbol{z}^{(t)}) - \frac{1}{2})$; `// weights`
3. Update the entire deep learner $f_{\boldsymbol{\theta}}$ with $\{\boldsymbol{y}, \boldsymbol{x}\}$
   $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \eta \nabla f_{\boldsymbol{\theta}^{(t-1)}}(\boldsymbol{x} \mid \boldsymbol{y}, \text{sample\_weights} = \boldsymbol{\omega}^{(t)})$;
   `// SGD`

Return $\hat{\boldsymbol{y}} = \begin{cases} 1, \text{ if } f_{\boldsymbol{\theta}^{(T)}}(\boldsymbol{x}) > \frac{1}{2} \\ -1, \text{ otherwise.} \end{cases}$

Algorithm 9 as DA-logit. For appropriate comparison, we adopt the same network structures, such as the number of layers, the number of hidden nodes, and regularizations like dropout rates, for DL and our methods. The differences between our methods and DL are that (1) the top layer weights W_0, b_0 of DL are updated via SGD optimization, while the weights W_0, b_0 of our methods are updated via MCMC or EM; (2) for binary classification, DA-logit and DL adopt a sigmoid activation function in the top layer to produce a binary output, while DA-SVM uses a linear function in the top layer and the augmented sampling layer transforms the continuous value into a binary output. For all experiments, the datasets are partitioned into 70% training and 30% testing randomly. For the optimization we use a modification of the SGD algorithm, the Adaptive moment estimation (Adam, Kingma and Ba (2015)) algorithm. The Adam algorithm combines the estimate of the stochastic gradient with the earlier estimate of the gradient, and scales this using an estimate of the second moment of the unit-level gradient. We have also explored RMSprop (Tieleman and Hinton, 2012) optimizer and we observe similar decreases in regression or classification errors.

To illustrate how the choice of J could affect the speed of convergence, we include different implementations of DA-GR and DA-SVM with $J = 2, 5, 10$. We have explored different

sampling noise variance τ_0, τ_Z, but the choices, in general, do not affect the results significantly.

5.5.1 Friedman Data

The benchmark (Friedman, 1991) setup uses a regression of the form

$$y_i = 10\sin(\pi x_{i1} x_{i2}) + 20(x_{i3} - 0.5)^2 + 10x_{i4} + 5x_{i5} + \epsilon_i, \quad \text{with} \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2),$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i1}, \ldots, x_{ip})$ and only the first 5 covariates are predictive of y_i. We run the experiments with $n = 100, 1\,000$ and $p = 10, 50, 100, 1\,000$ to explore the performance in both low dimensional and high dimensional scenarios. We implement both one-layer ($L = 1$) and two-layer ($L = 2$) ReLU networks with 64 hidden units in each layer. For DA-GR model, we let $\tau_0 = 0.1, \tau_z = 1$. The experiments are repeated 50 times with different random seeds.

Figure 5.2 reports the three quartiles of the out-of-sample squared errors (MSEs). The top row is the performance of the one-layer networks and the bottom row is the performance of the two-layer networks. The two-layer networks perform better and converge faster. For DA-GR, when $J = 5$ or $J = 10$, it converges significantly faster and the prediction errors are also smaller. When $J = 2$, the performance of DA-GR is relatively similar to the deep learning model with only SGD updates. This is due to the fact that DA-GR with J-copies learns the posterior mode which is equivalent to the minimization point of the objective function, and it concentrates on the mode faster when J becomes larger.

The computation costs of DA are higher as shown in Figure 5.3. This is not entirely unexpected since we introduce sampling steps. When J increases, the computation costs also increase slightly. Given the improvement in convergence speed and prediction errors, our data augmentation strategies are still worthwhile even with some extra computation costs. In addition, for each epoch, we can draw the sample-wise posteriors in parallel and

the gap between the computation time can be further mitigated.

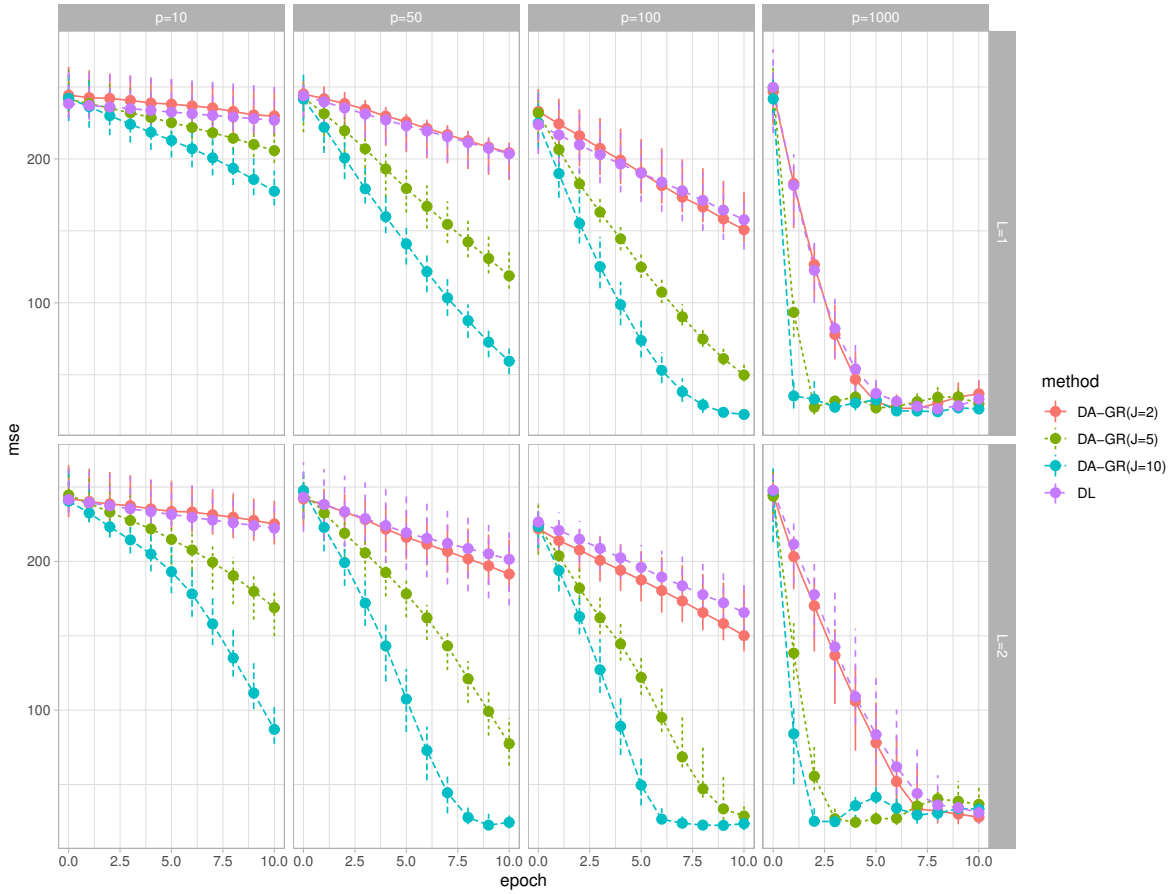


Figure 5.2: Quartiles of out-of-sample MSEs under the Friedman Setup. We explore cases where $n = 1\,000$ and $p = 10, 50, 100, 1\,000$. The tests are repeated 50 times. The medians of out-of-sample MSEs after training for 1 to 10 epochs are plotted with lines and the vertical bars mark the 25 % and 75% quantiles of the MSEs. DA-GR refers to DA Gaussian regression shown in Algorithm 7 and DL stands for the ReLU networks without the data-augmentation layer.

5.5.2 Boston Housing Data

Another classical regression benchmark dataset is the Boston Housing dataset[1], see, for example, Hernández-Lobato and Adams (2015). The data contains $n = 506$ observations with 13 features. To show the robustness of DA, we repeat the experiment 20 times with

1. https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

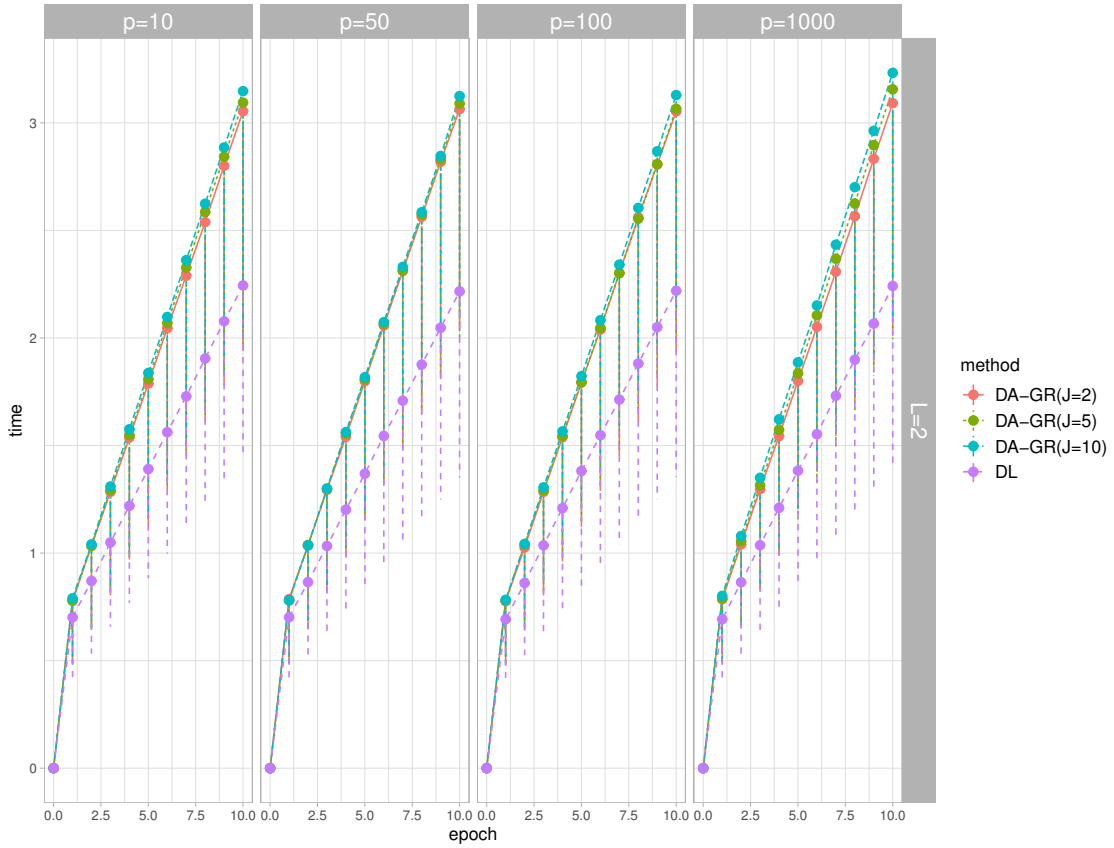


Figure 5.3: Computation time under the Friedman setup. The setups are $n = 1\,000$ and $p = 10, 50, 100, 1\,000$. The averaged time (over 50 repetitions) for computing 1 to 10 epochs is plotted with lines and the vertical bars mark the 25% and 75% quantiles of the computation time collected. We only include one figure of computation time comparison here since the scale is relatively the same for all cases.

different training subsets. We adopt the ReLU networks with one hidden layer of 64 units and set the dropout rate to be 0.5. For the DA-GR model, we let $\tau_0 = 0.1, \tau_Z = 1$.

Figure 5.4 shows the prediction errors of all methods. DA-GR with $J = 10$ performs significantly better than the others, in terms of both prediction errors and convergence rates. Meanwhile, DA-GR with $J = 2$ behaves similarly to SGD at the beginning, but it converges significantly faster than SGD after a few epochs. This again, shows that with the J-copies strategy, our method helps the optimization converge at a faster speed, and injecting the noise helps the model generalize well out-of-sample.

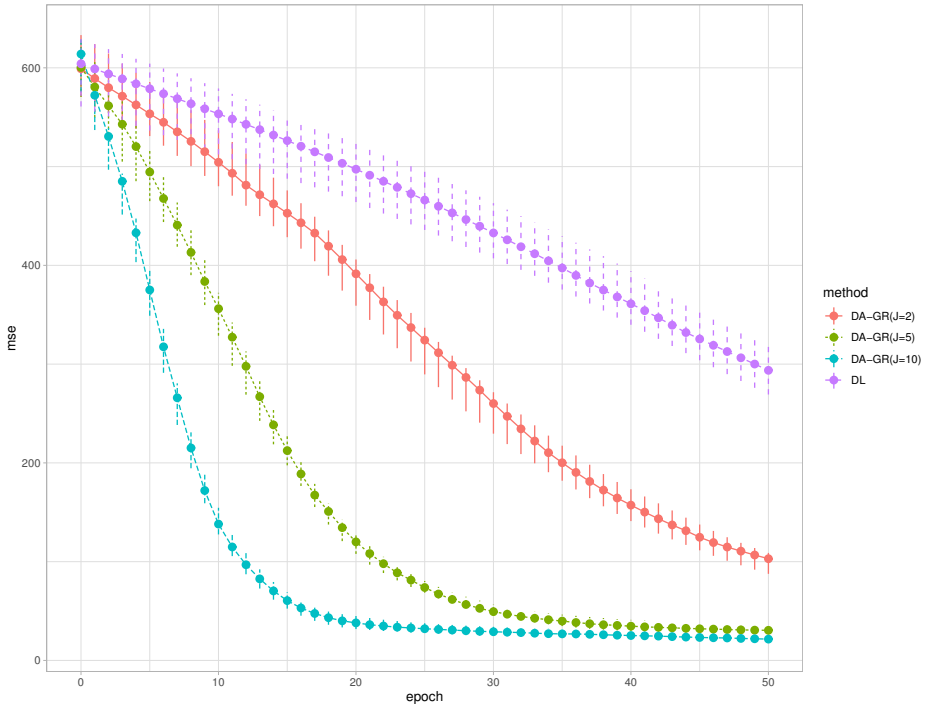


Figure 5.4: Out-of-sample MSEs for the Boston housing dataset. The experiment is repeated 20 times with different training subsampling. The medians of MSEs after training for 1 to 50 epochs are provided, with the vertical bars marking the 25% and 75% quantiles of the errors. DA-GR refers to the data augmentation strategy in Algorithm 7 and DL stands for the ReLU networks without the data-augmentation layer.

5.5.3 Wine Quality Data Set

The Wine Quality Data Set [2] contains 4 898 observations with 11 features. The output wine rating is an integer variable ranging from 0 to 10 (the observed range in the data is from 3 to 9). The frequency of each rating is reported in Table 5.2.

rating	3	4	5	6	7	8	9
frequency	20	163	1457	2198	880	175	5

Table 5.2: Frequencies of different wine ratings

The most frequent ratings are 5 and 6. Since we focus on binary classification problems, we provide two types of classifications, both of which have relatively balanced categories:

2. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, 'Wine Quality Data Set', UCI Machine Learning Repository.

(1) wine with a rating of 5 or 6 (Test 1); (2) wine with a rating of ≤ 5 or > 5 (Test 2). We use the same network architectures adopted in Friedman's example with $\tau_0 = \tau_z = 0.1$.

Figure 5.5 provides results for the two types of binary classifications. In both cases, DA-SVM performs better than DA-logit and DL. The advantage of large J is still significant and helps converge especially in the early phase. DA-logit outperforms DL in Test 1 when the network is shallow (L=1), while in other cases performs similarly to DL.

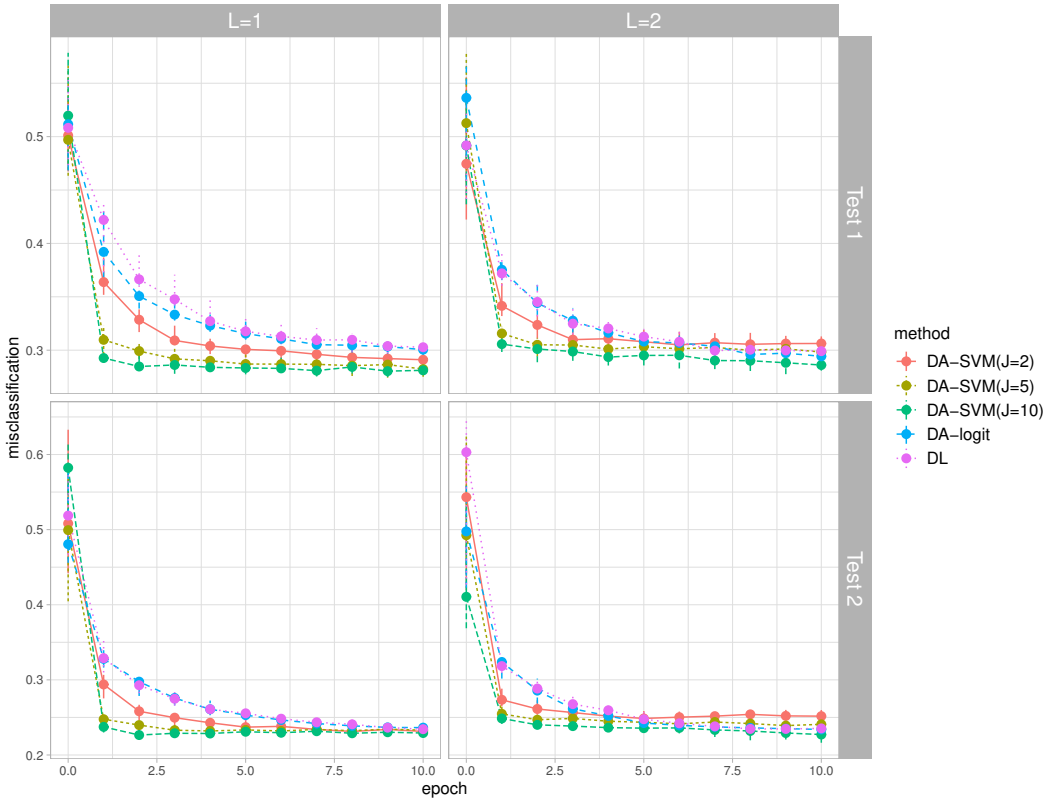


Figure 5.5: Binary classifications on the wine quality dataset. Two types of binary classifications are considered here. The experiment is repeated 20 times with different training subsampling. We compare the misclassification rates of DA-SVM in Algorithm 8 with $J = 2, 5, 10$, DA-logit in Algorithm 9 and the ReLU networks without the data augmentation layer (DL), after training for 1 to 10 epochs.

5.5.4 Airbnb Data Set

The Airbnb Kaggle competition[3] provides a more challenging application with 213451 observations in total, and classified by destination into 12 classes: 10 most popular countries, other and no destination found (NDF), where *other* corresponds to any other country which is not among the top 10 and NDF corresponds to situations that no booking was made. The countries are denoted with their standard codes, as 'AU' for Australia, 'CA' for Canada, 'DE' for Germany, 'ES' for Spain, 'FR' for France, 'UK' for United Kingdom, 'IT' for Italy, 'NL' for Netherlands, 'PT' for Portugal, 'US' for United States. Table 5.3 reports the percentage of each class. We follow the preprocessing steps in Polson and Sokolov (2017). The list of variables contains information from the sessions records (number of sessions, summary statistics of action types, device types and session duration), and user tables such as gender, language, affiliate provider etc. All categorical variables are converted to binary dummies, which leads to 661 features in total. For the neural network architecture, we use a two-layer ReLU network with 64 hidden units on each layer and set the dropout rate to be 0.3. For the SVM model, we let $\tau_0 = \tau_z = 0.1$.

	AU	CA	DE	ES	FR	UK	IT	NDF	NL	PT	US	other
% obs	0.25	0.67	0.50	1.05	2.35	1.09	1.33	58.35	0.36	0.10	29.22	4.73

Table 5.3: Percentage of each class (#obs = 213451)

Our goal is to test the binary classification models on this dataset. We consider two types of binary responses, both of which have relatively balanced amounts of observations in each category.

1. Spain (1.05%) vs United Kingdom(1.09%)

2. United Kingdom (1.09%) vs Italy (1.33%)

3. https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings

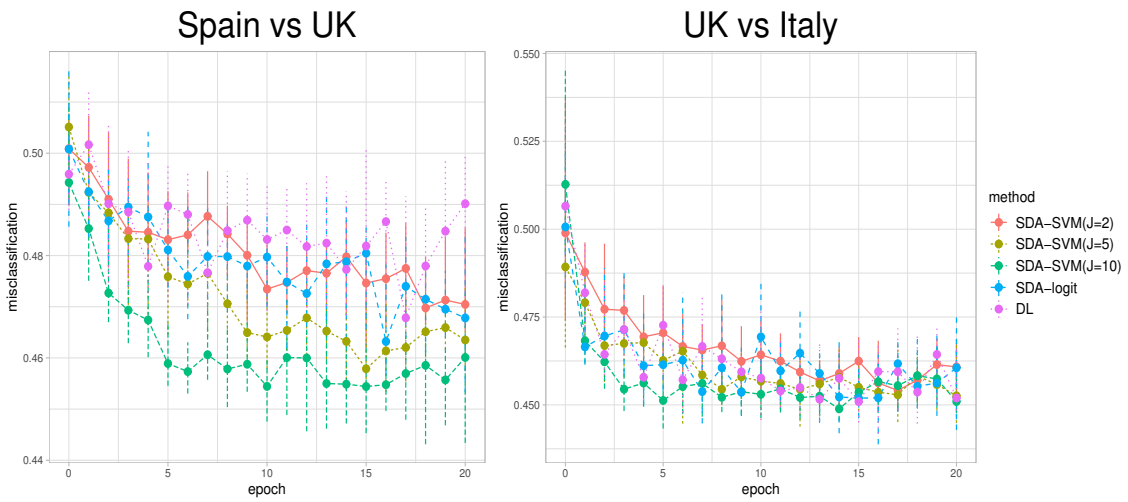


Figure 5.6: Binary classifications on the Airbnb booking dataset. Two types of binary classifications are considered here. The experiment is repeated 20 times with different training subsampling. We compare the misclassification rates of DA-SVM in Algorithm 8 with $J = 2, 5, 10$, DA-logit in Algorithm 9 and the ReLU networks without the data augmentation layer, after training for 1 to 20 epochs.

Figure 5.6 demonstrates the binary classifications for Spain versus UK and UK versus Italy. For both cases, the out-of-sample misclassification rates are not small and the fluctuations over epochs are big, suggesting that a better model structure may be needed. However, we still observe that DA-SVM with $J = 5$ or $J = 10$ has smaller classification errors over epochs and the out-of-sample errors decrease faster during earlier phase of training.

5.5.5 Summary of Experiment Results

From the above examples, we observe that DA-logit which is implemented under the EM principle does not show an obvious advantage over the vanilla neural network. It shows some improvements on the convergence speed when the network is shallow in the Wine Quality dataset case as in Figure 5.5. This could be partially due to the fact that we did not apply regularization on the DA layer for our logit implementation. More importantly, the performance of the EM algorithm is contingent on the statistical properties of the objective function. Although the surrogate function is constructed via only the top layer whose

quadratic form ensures concavity, the property of the objective function as a whole becomes complicated when the deep network architecture is more complex. Since our method also inherits the negative side of EM and MM algorithms, convergence to the global maximum is not guaranteed in the absence of concavity. However, this observation could open the possibility of future research where we can combine the EM algorithms with shape-constrained neural networks (Gupta et al., 2020).

On the contrary, the MCMC methods with the J-copies strategy significantly improve the prediction errors and convergence speed of the neural networks for both regression and classification problems. And the advantages become more outstanding when J is larger. The phenomenon suggests that the stochastic exploratory methods are preferable when the statistical property of the objective function is unknown or too complex. And the J-copies scheme largely relieves the problem of being trapped into local modes.

One concern of using MCMC methods is the extra computation costs induced by the sampling steps. In our current version where $p_1 = 1$, the sample-wise sampling steps can be computed in parallel. If one wishes to introduce a higher dimension latent variable Z_1 such that $p_1 > 1$, the computation costs will increase as it may involve sampling from multivariate distributions. In that case, fast sampling implementation such as Bhattacharya et al. (2016) is recommended to speed up the process.

5.6 Discussion

Various regularization methods have been deployed in neural networks to prevent overfitting, such as early stopping, weight decay, dropout (Hinton et al., 2012b), gradient noise (Neelakantan et al., 2017). Bayesian strategies tackle the regularization problem by proposing probability structures on the weights. We show that data augmentation strategies are available for many standard activation functions (ReLU, SVM, logit) used in deep learning.

Using MCMC provides a natural stochastic search mechanism that avoids procedures

such as back-tracking and provides full descriptions of the objective function over the entire range $\boldsymbol{\Theta}$. Training deep neural networks thus benefits from additional hidden stochastic augmentation units (a.k.a. data augmentation). Uncertainty can be injected into the network through the probabilistic distributions on only one or two layers, permitting more variability of the network. When more data are observed, the level of uncertainty decreases as more information is learned and the network becomes more deterministic. We also exploit the duality between *maximum a posteriori* estimation and optimization. We provide a J-copies stacking scheme to speed up the convergence to posterior mode and avoid trapping attraction of the local modes. Concerning efficiency, DA provides a natural framework to convert the objective function into weighted least squares and is straightforward to implement with the current deep learning training process.

Our three motivational examples illustrated the advantages of data augmentation. Our work has the potential to be generalized to many other data augmentation schemes and different regularization priors. Probabilistic structures on more units and layers are also possible to allow for more uncertainty.

Our DA-DL methods enjoy the benefits of both worlds. On one hand, with the data augmentation on top, it is robust to random weight initialization. Although we still need to specify the learning rates for the deep architecture, the top layer can learn adaptively and the entire network becomes less sensitive to the choice of learning rate. On the other hand, the fast SGD updates from the deep architecture largely alleviate the computation concerns compared to a fully Bayesian hierarchical model.

There are many directions to future research, including adding more sampling layers so the model could accommodate more randomness and flexibility, and using weighted Bayesian bootstrap (Newton et al., 2021) to approximate the unweighted posteriors by assigning random weight to each observation and penalty. Uncertainty quantification for prediction is also possible. Although we focus on the training aspect of deep learning, one can collect

posterior draws $\boldsymbol{\theta}^{(t)}$ from the MCMC procedure when the training process converges. Using (5.8), we can construct predictive intervals and conduct inference.

CHAPTER 6

VARIABLE SELECTION WITH ABC BAYESIAN FORESTS

Few problems in statistics are as perplexing as variable selection in the presence of very many redundant covariates. The variable selection problem is most familiar in parametric environments such as the linear model or additive variants thereof. In this work, we abandon the linear model framework, which can be quite detrimental when the covariates impact the outcome in a non-linear way, and turn to tree-based methods for variable selection. Such variable screening is traditionally done by pruning down large trees or by ranking variables based on some importance measure. Despite heavily used in practice, these ad-hoc selection rules are not yet well understood from a theoretical point of view. In this work, we devise a Bayesian tree-based probabilistic method and show that it is consistent for variable selection when the regression surface is a smooth mix of $p > n$ covariates. These results are the first model selection consistency results for Bayesian forest priors. Probabilistic assessment of variable importance is made feasible by a spike-and-slab wrapper around sum-of-trees priors. Sampling from posterior distributions over trees is inherently very difficult. As an alternative to MCMC, we propose ABC Bayesian Forests, a new ABC sampling method based on data-splitting that achieves higher ABC acceptance rate. We show that the method is robust and successful at finding variables with high marginal inclusion probabilities. Our ABC algorithm provides a new avenue towards approximating the median probability model in non-parametric setups where the marginal likelihood is intractable.

. Adopted from Yi Liu, Veronika Ročková, and Yuexi Wang. Variable selection with abc bayesian forests. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3):453–481, 2021.

6.1 Perspectives on Non-parametric Variable Selection

In its simplest form, variable selection is most often carried out in the context of linear regression (Tibshirani, 1996; George and McCulloch, 1993; Fan and Li, 2001). However, confinement to linear parametric forms can be quite detrimental for variable importance screening, when the covariates impact the outcome in a non-linear way (Turlach, 2004). Rather than first selecting a parametric model to filter out variables, another strategy is to first select variables and then build a model. Adopting this reversed point of view, we focus on developing methodology for the so called "model-free" variable selection (Chipman et al., 2001).

There is a long strand of literature on the fundamental problem of non-parametric variable selection. One line of research focuses on capturing non-linearities and interactions with basis expansions and performing grouped shrinkage/selection on sets of coefficients (Scheipl, 2011; Ravikumar et al., 2009; Lin and Zhang, 2006; Radchenko and James, 2010). Lafferty and Wasserman (2008) propose the RODEO method for sparse non-parametric function estimation through regularization of the derivative expectation operator and provide a consistency result for the selection of the optimal bandwidth. Candes et al. (2018) propose a model-free knock-off procedure, controlling FDR in settings when the conditional distribution of the response is arbitrary. In the Bayesian literature, Savitsky et al. (2011) deploy spike-and-slab priors on covariance parameters of Gaussian processes to erase variables. In this work, we focus on other non-parametric regression techniques, namely trees/forests which have been ubiquitous throughout machine learning and statistics (Breiman, 2001; Chipman et al., 2010). The question we wish to address is whether one can leverage the flexibility of regression trees for effective (consistent) variable importance screening.

While trees are routinely deployed for data exploration, prediction and causal inference (Hill, 2011; Taddy et al., 2011a; Gramacy and Lee, 2008a), they have also been used for dimension reduction and variable selection. This is traditionally done by pruning out variables

or by ranking them based on some importance measure. The notion of variable importance was originally proposed for CART using overall improvement in node impurity involving surrogate predictors (Breiman et al., 1984). In random forests, for example, the importance measure consists of a difference between prediction errors before and after noising the covariate through a permutation in the out-of-bag sample. However, this continuous variable importance measure is on an arbitrary scale, rendering variable selection ultimately ad-hoc. Principled selection of the importance threshold (with theoretical guarantees such as FDR control or model selection consistency) is still an open problem. Simplified variants of importance measures have begun to be understood theoretically for variable selection only very recently (Ishwaran, 2007; Kazemitabar et al., 2017).

Bayesian trees and forests select variables based on probabilistic considerations. The BART procedure (Chipman et al., 2010) can be adapted for variable selection by forcing the number of available splits (trees) to be small, thereby introducing competition between predictors. BART then keeps track of predictor inclusion frequencies and outputs a probabilistic importance measure: an average proportion of all splitting rules inside a tree ensemble that split on a given variable, where the average is taken over the MCMC samples. This measure cannot be directly interpreted as the posterior variable inclusion probability in anisotropic regression surfaces, where wigglier directions require more splits. Bleich et al. (2014) consider a permutation framework for obtaining the null distribution of the importance weights. Zhu et al. (2015) implement reinforcement learning for selection of splitting variables during tree construction to encourage splits on fewer more important variables. All these developments point to the fact that regularization is key to enhancing performance of trees/forests in high dimensions. Our approach differs in that we impose regularization from *outside* the tree/forest through a spike-and-slab wrapper.

Spike-and-slab variable selection consistency results have relied on analytical tractability (approximation availability) of the marginal likelihood (Narisetty and He, 2014; Johnson

and Rossell, 2012; Castillo et al., 2015). Nicely tractable marginal likelihoods are ultimately unavailable in our framework, rendering the majority of the existing theoretical tools inapplicable. For these contexts, Yang and Pati (2017) characterized general conditions for model selection consistency, extending the work of Lember and van der Vaart (2007) to non *iid* setting. Exploiting these developments, we show variable selection consistency of our non-parametric spike-and-slab approach when the regression function is a smooth mix of covariates. Building on Ročková and van der Pas (2020), our paper continues the investigation of missing theoretical properties of Bayesian CART and BART. We show model selection consistency when the smoothness is known as well as joint consistency for both the regularity level *and* active variable set when the smoothness is not known and when $p > n$. These results are the first model selection consistency results for Bayesian forest priors.

The absence of a tractable marginal likelihood complicates not only theoretical analysis, but also computation. We turn to Approximate Bayesian Computation (ABC) (Plagnol and Tavaré, 2004; Marin et al., 2012; Csillery et al., 2010) and propose a procedure for model-free variable selection. Our ABC method *does not* require the use of low-dimensional summary statistics and, as such, it *does not* suffer from the known difficulty of ABC model choice (Robert et al., 2011). Our method is based on sample splitting where at each iteration (a) a random subset of data is used to come up with a proposal draw and (b) the rest of the data is used for ABC acceptance. This new data-splitting approach increases ABC effectiveness by increasing its acceptance rate. ABC Bayesian forests relate to the recent line of work on combining machine learning with ABC (Pudlo et al., 2015; Jiang et al., 2017a). We propose dynamic plots that describe the evolution of marginal inclusion probabilities as a function of the ABC selection threshold.

The paper is structured as follows. Section 6.2 introduces the spike-and-slab wrapper around tree priors. Section 6.3 develops the ABC variable selection algorithm. Section 6.4 presents model selection consistency results. Section 6.5 demonstrates the usefulness of the

ABC method on simulated data and we apply our methods to an analysis on HIV dataset in Section 6.6. Section 6.7 wraps up with a discussion.

Notation. With $\|\cdot\|_n$ we denote the empirical L^2 norm. The class of functions $f(\boldsymbol{x}) : [0,1]^p \to \mathbb{R}$ such that $f(\cdot)$ is constant in all directions excluding $\mathcal{S}_0 \subseteq \{1,\ldots,p\}$ is denoted with $\mathcal{C}(\mathcal{S}_0)$. With $\mathcal{H}_p^\alpha$, we denote α-Hölder continuous functions with a smoothness coefficient α. $a \lesssim b$ denotes a is less or equal to b, up to a multiplicative positive constant, and $a \asymp b$ denotes $a \lesssim b$ and $b \lesssim a$. The ε-covering number of a set Ω for a semimetric d, denoted by $N(\varepsilon;\Omega;d)$, is the minimal number of d-balls of radius ε needed to cover set Ω.

6.2 Bayesian Subset Selection with Trees

We will work within the purview of non-parametric regression, where a vector of continuous responses $\boldsymbol{Y}^{(n)} = (Y_1,\ldots,Y_n)'$ is linked to fixed (rescaled) predictors $\boldsymbol{x}_i = (x_{i1},\ldots,x_{ip})' \in [0,1]^p$ for $1 \le i \le n$ through

$$Y_i = f_0(\boldsymbol{x}_i) + \varepsilon_i \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(0,\sigma^2) \quad \text{for} \quad 1 \le i \le n, \tag{6.1}$$

where $f_0(\cdot)$ is the regression mixing function and $\sigma^2 > 0$ is a scalar. It is often reasonable to expect that only a small subset $\mathcal{S}_0$ of $q_0 = |\mathcal{S}_0|$ predictors actually exert influence on $\boldsymbol{Y}^{(n)}$ and contribute to the mix. The subset $\mathcal{S}_0$ is seldom known with certainty and we are faced with the problem of variable selection. Throughout this paper, we assume that the regression surface is smoothly varying (α-Hölder continuous) along the active directions $\mathcal{S}_0$ and constant otherwise, i.e. we write $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$.

Unlike linear models that capture the effect of a single covariate with a single coefficient, we permit non-linearities/interactions and capture variable importance with (additive) regression trees. By doing so, we hope to recover non-linear signals that could be otherwise missed by linear variable selection techniques.

As with any other non-parametric regression method, regression trees are vulnerable to the curse of dimensionality, where prediction performance deteriorates dramatically as the number of variables p increases. If an oracle were to isolate the active covariates $\mathcal{S}_0$, the fastest achievable estimation rate would be $n^{-\alpha/(2\alpha+|\mathcal{S}_0|)}$. This rate depends only on the intrinsic dimensionality $q_0 = |\mathcal{S}_0|$, not the actual dimensionality p which can be much larger than n. Recently, Ročková and van der Pas (2020) showed that with *suitable regularization*, the posterior distribution for Bayesian CART and BART actually concentrates at this fast rate (up to a log factor), adapting to the intrinsic dimensionality and smoothness. Later in Section 6.4, we continue their theoretical investigation and focus on consistent *variable selection*, i.e. estimation of $\mathcal{S}_0$ rather than $f_0(\cdot)$. Spike-and-slab regularization plays a key role in obtaining these theoretical guarantees.

6.2.1 Trees with Spike-and-Slab Regularization

Many applications offer a plethora of predictors and some form of redundancy penalization has to be incurred to cope with the curse of dimensionality. Bayesian regression trees were originally conceived for prediction rather than variable selection. Indeed, original tree implementations of Bayesian CART (Denison et al., 1998a; Chipman et al., 1998) do not seem to penalize inclusion of redundant variables aggressively enough. As noted by Linero (2018), the prior expected number of active variables under the Bayesian CART prior of Chipman et al. (1998) satisfies $\lim_{p\to\infty} \mathbb{E}[q] = K-1$ as $p \to \infty$ where K is the fixed number of bottom leaves. This behavior suggests that (in the limit) the prior forces inclusion of the maximal number of variables while splitting on them only once. This is far from ideal. To alleviate this issue, we deploy the so-called *spike-and-forest priors*, i.e. spike-and-slab wrappers around sum-of-trees priors (Ročková and van der Pas, 2020). As with the traditional spike-and-slab priors, the specification starts with a prior distribution over the 2^p active variable sets:

$$\mathcal{S} \sim \pi(\mathcal{S}) \quad \text{for each} \quad \mathcal{S} \subseteq \{1, \ldots, p\}. \tag{6.2}$$

We elaborate on the specific choices of $\pi(\mathcal{S})$ later in Section 6.3.2 and Section 6.4.

Given the pool of variables $\mathcal{S}$, a regression tree/forest is grown using *only* variables inside $\mathcal{S}$. This prevents the trees from using too many variables and thereby from overfitting. Recall that each individual regression tree is characterized by two components: (1) a tree-shaped K-partition of $[0,1]^p$, denoted with $\mathcal{T}$, and (2) bottom node parameters (step heights), denoted with $\boldsymbol{\beta} \in \mathbb{R}^K$. Starting with a parent node $[0,1]^p$, each K-partition is grown by recursively dissecting rectangular cells at chosen internal nodes along one of the active coordinate axes, all the way down to K terminal nodes. Each tree-shaped K-partition $\mathcal{T} = \{\Omega_k\}_{k=1}^K$ consists of K partitioning rectangles $\Omega_k \subset [0,1]^p$.

While Bayesian CART approximates $f_0(\boldsymbol{x})$ with a single tree mappings $f_{\mathcal{T},\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{k=1}^K \mathbb{I}(\boldsymbol{x} \in \Omega_k)\beta_k$, Bayesian Additive Regression Trees (BART) use an aggregate of T mappings

$$f_{\mathcal{E},\boldsymbol{B}}(\boldsymbol{x}) = \sum_{t=1}^{T} f_{\mathcal{T}^t,\boldsymbol{\beta}^t}(\boldsymbol{x})$$

where $\mathcal{E} = \{\mathcal{T}^1, \ldots, \mathcal{T}^T\}$ is an ensemble of tree partitions and $\boldsymbol{B} = [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^T]$ is an ensemble of step coefficients. In a fully Bayesian approach, prior distributions have to be specified over the set of tree structures $\mathcal{E}$ and over terminal node heights $\boldsymbol{B}$. The spike-and-forest construction can accommodate various tree prior options.

To assign a prior over $\mathcal{E}$ for a given T, one possibility is to first pick the number of bottom nodes, independently for each tree, from a prior

$$K^t \sim \pi(K) \quad \text{for} \quad K = 1, \ldots, n, \tag{6.3}$$

such as the Poisson distribution (Denison et al., 1998a). Given the vector of tree sizes $\boldsymbol{K} = (K^1, \ldots, K^T)'$ and a set of covariates $\mathcal{S}$, we assign a prior over so-called valid ensembles/forests $\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$. We say that a tree ensemble $\mathcal{E}$ is valid if it consists of trees that have non-empty bottom leaves. One can pick a tree partition ensemble from a uniform prior over

valid forests $\mathcal{E} \in \mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$, i.e.

$$\pi(\mathcal{E} \mid \mathcal{S}, \boldsymbol{K}) = \frac{1}{\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})} \mathbb{I}\left(\mathcal{E} \in \mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}\right), \tag{6.4}$$

where $\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})$ is the number of valid tree ensembles characterized by $\boldsymbol{K}$ bottom leaves and split directions $\mathcal{S}$. The prior (6.3) and (6.4) was deployed in the Bayesian CART implementation of Denison et al. (1998a) (with $T = 1$) and it was studied theoretically by Ročková and van der Pas (2020). Another related Bayesian forest prior (implemented in the BART procedure and studied theoretically by Ročková and Saha (2019) consists of an independent product of branching process priors (one for each tree) with decaying split probabilities (Chipman et al., 1998). The implementation is very similar to the one of Denison et al. (1998a).

Finally, given the partitions $\mathcal{T}^t$ of size K^t for $1 \leq t \leq T$, one assigns (independently for each tree) a Gaussian product prior on the step heights

$$\pi(\boldsymbol{\beta}^t \mid K^t) = \prod_{k=1}^{K^t} \phi(\beta_k^t; \sigma_{\boldsymbol{\beta}}^2), \tag{6.5}$$

where $\phi(x; \sigma_{\boldsymbol{\beta}}^2)$ denotes a Gaussian density with mean zero and variance $\sigma_{\boldsymbol{\beta}}^2 = 1/T$ (as suggested by Chipman et al. (2010)). The prior for σ^2 can be chosen as inverse chi-squared with hyperparameters chosen based on an estimate of the residual standard deviation of the data (Chipman et al., 2010).

The most crucial component in the spike-and-forest construction, which sets it apart from existing BART implementations, is the active set $\mathcal{S}$ which serves to mute variables by restricting the pool of predictors available for splits. The goal is to learn which set $\mathcal{S}$ is most likely (a posteriori) and/or how likely each variables is to have contributed to f_0. Unlike related tree-based variable selection criteria, the spike-and-slab envelope makes it possible to perform variable selection directly by evaluating posterior model probabilities $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$ or

marginal inclusion probabilities $\Pi(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$ for $1 \leq j \leq p$. Random forests (Breiman, 2001) also mute variables, but they do so from within the tree by randomly choosing a small subset of variables for each split. The spike-and-slab approach mutes variables externally rather than internally. Bleich et al. (2014) note that when the number of trees is small, the Gibbs sampler for BART can get trapped in local modes which can destabilize the estimation procedure. On the other hand, when the number of trees is large, there are ample opportunities for the noise variables to enter the model without necessarily impacting the model fit, making variable selection very challenging. Our spike-and-slab wrapper is devised to get around this problem.

The problem of variable selection is fundamentally challenged by the sheer size of possible variable subsets. For linear regression, (a) MCMC implementations exist that capitalize on the availability of marginal likelihood (Narisetty and He, 2014; Guan and Stephens, 2011), (b) optimization strategies exist for both continuous (Ročková and George, 2018; Ročková, 2018) and point-mass spike-and slab priors (Carbonetto and Stephens, 2012). These techniques do not directly translate to tree models, for which tractable marginal likelihoods $\pi(\boldsymbol{Y}^{(n)} \mid \mathcal{S})$ are unavailable. To address this computational challenge, we explore ABC techniques as a new promising avenue for non-parametric spike-and-slab methods.

6.3 ABC for Variable Selection

Performing (approximate) posterior inference in complex models is often complicated by the analytical intractability of the marginal likelihood. Approximate Bayesian Computation (ABC) is a simulation-based inference framework that obviates the need to compute the likelihood directly by evaluating the proximity of (sufficient statistics of) observed data and pseudo-data simulated from the likelihood. Simon Tavaré first proposed the ABC algorithm for posterior inference (Tavaré et al., 1997) in the 1990's and since then it has widely been

used in population genetics, systems biology, epidemiology and phylogeography[1].

Combined with a probabilistic structure over models, marginal likelihoods give rise to posterior model probabilities, a standard tool for Bayesian model choice. When the marginal likelihood is unavailable (our case here), ABC offers a unique computational solution. However, as pointed out by Robert et al. (2011), ABC cannot be trusted for model comparisons when model-wise sufficient summary statistics are not sufficient across models. The ABC approximation to Bayes factors then does not converge to exact Bayes factors, rendering ABC model choice fundamentally untrustworthy. A fresh new perspective to ABC model choice was offered in Pudlo et al. (2015), who rephrase model selection as a classification problem that can be tackled with machine learning tools. Their idea is to treat the ABC reference table (consisting of samples from a prior model distribution and high-dimensional vectors of summary statistics of pseudo-data obtained from the prior predictive distribution) as an actual data set, and to train a random forest classifier that predicts a model label using the summary statistics as predictors. Their goal is to produce a stable model decision based on a classifier rather than on an estimate of posterior model probabilities. Our approach has a similar flavor in the sense that it combines machine learning with ABC, but the concept is fundamentally very different. Here, the fusion of Bayesian forests and ABC is tailored to non-parametric variable selection towards obtaining posterior variable inclusion probabilities. Our model selection approach does not suffer from the difficulty of ABC model choice as we *do not* commit to any summary statistics and use random subsets of observations to generate the ABC reference table.

6.3.1 Naive ABC Implementation

For its practical implementation, our Bayesian variable selection method requires sampling from the analytically intractable posterior distribution over subsets $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$ under the

1. The study of how human beings migrated throughout the world in the past.

spike-and-forest prior (6.4), (6.3) and (6.2). Given a single tree partition $\mathcal{T}$, the (conditional) marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \,|\, \mathcal{T}, \mathcal{S})$ is available in closed form, facilitating implementations of Metropolis-Hastings algorithms (Chipman et al., 1998; Denison et al., 1998a) (see Liu et al. (2021, Section S.3)). However, such MCMC schemes can suffer from poor mixing. Taking advantage of the fact that, despite being intractable, one can *simulate from* the marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \,|\mathcal{S})$, we will explore the potential of ABC as a complementary development to MCMC implementations.

The principle at the core of ABC is to perform approximate posterior inference from a given dataset by simulating from a prior distribution and by comparisons with numerous synthetic datasets. In its standard form, an ABC implementation of model choice creates a reference table, recording a large number of datasets simulated from the model prior and the prior predictive distribution under each model. Here, the table consists of M pairs $(\mathcal{S}_m, \boldsymbol{Y}_m^\star)$ of model indices $\mathcal{S}_m$, simulated from the prior $\pi(\mathcal{S})$, and pseudo-data $\boldsymbol{Y}_m^\star \in \mathbb{R}^n$, simulated from the marginal likelihood $\pi(\boldsymbol{Y}^{(n)} \,|\, \mathcal{S}_m)$. To generate $\boldsymbol{Y}_m^\star$ in our setup, one can hierarchically decompose the marginal likelihood

$$\pi(\boldsymbol{Y}^{(n)} \,|\, \mathcal{S}) = \int_{(f_{\mathcal{E},\boldsymbol{B}},\sigma^2)} \pi(\boldsymbol{Y}^{(n)} \mid f_{\mathcal{E},\boldsymbol{B}}, \sigma^2) \mathrm{d}\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S}) \tag{6.6}$$

and first draw $(f_{\mathcal{E},\boldsymbol{B}}^m, \sigma_m^2)$ from the prior $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S})$ and obtain $\boldsymbol{Y}_m^\star$ from (6.1), given $(f_{\mathcal{E},\boldsymbol{B}}^m, \sigma_m^2)$. ABC sampling is then followed by an ABC rejection step, which extracts pairs $(\mathcal{S}_m, \boldsymbol{Y}_m^\star)$ such that $\boldsymbol{Y}_m^\star$ is close enough to the actual observed data. In other words, one trims the reference table by keeping only model indices $\mathcal{S}_m$ paired with pseudo-observations that are at most ϵ-away from the observed data, i.e. $\|\boldsymbol{Y}^{obs} - \boldsymbol{Y}_m^\star\|_2 \leq \epsilon$ for some tolerance level ϵ. These extracted values comprise an approximate ABC sample from the posterior $\pi(\mathcal{S} \,|\, \boldsymbol{Y}^{(n)})$, which should be informative for the relative ordering of the competing models, and thus variable selection (Grelaud et al., 2009). Note that this particular ABC implementation does not require any use of low-dimensional summary statistics, where rejection is based

solely on $\boldsymbol{Y}^{obs}$. While theoretically justified, this ABC variant has two main drawbacks.

First, with very many predictors, it will be virtually impossible to sample from all 2^p model combinations at least once, unless the reference table is huge. Consequently, relative frequencies of occurrence of a model $\mathcal{S}_m$ in the trimmed ABC reference table *may not* be a good estimate of the posterior model probability $\pi(\mathcal{S}_m \,|\, \boldsymbol{Y}^{(n)})$. While the model with the highest posterior probability $\pi(\mathcal{S}_m \,|\, \boldsymbol{Y}^{(n)})$ is commonly conceived as the right model choice, it may not be the optimal model for prediction. Indeed, in nested correlated designs and orthogonal designs, it is the median probability model that is predictive optimal (Barbieri and Berger, 2004). The median probability model (MPM) consists of those variables whose *marginal* inclusion probabilities $\mathbb{P}(j \in \mathcal{S}_0 \,|\, \boldsymbol{Y}^{(n)})$ are at least 0.5. While simulation-based estimates of posterior model probabilities $\mathbb{P}(\mathcal{S} \,|\, \boldsymbol{Y}^{(n)})$ can be imprecise, we argue (and show) that ABC estimates of marginal inclusion probabilities $\mathbb{P}(j \in \mathcal{S}_0 \,|\, \boldsymbol{Y}^{(n)})$ are far more robust and stable.

The second difficulty is purely computational and relates to the issue of coming up with good proposals $f^m_{\mathcal{E},\boldsymbol{B}}$ such that the pseudo-data are sufficiently close to $\boldsymbol{Y}^{obs}$. Due to the vastness of the tree ensemble space, it would be naive to think that one can obtain solid guesses of f_0 purely by sampling from non-informative priors. This is why we call this ABC implementation naive. These considerations lead us to a new data-splitting ABC modification that uses a random portion of the data to train the prior and to generate pseudo-data with more affinity to the left-out observations.

6.3.2 ABC Bayesian Forests

By sampling directly from noninformative priors over tree ensembles $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S})$, the acceptance rate of the naive ABC can be prohibitively small where huge reference tables would be required to obtain only a few approximate samples from the posterior.

To address this problem, we suggest a sample-splitting approach to come up with draws

that are less likely to be rejected by the ABC method. At each ABC iteration, we first draw a random subsample $\mathcal{I} \subset \{1, \ldots, n\}$ of size $|\mathcal{I}| = s$ with no replacement. Then we split the observed data $\boldsymbol{Y}^{(n)}$ into two groups, denoted with $\boldsymbol{Y}_{\mathcal{I}}^{(n)}$ and $\boldsymbol{Y}_{\mathcal{I}^c}^{(n)}$, and instead of (6.6) we consider the marginal likelihood conditionally on $\boldsymbol{Y}_{\mathcal{I}}^{(n)}$

$$\pi(\boldsymbol{Y}^{(n)} \mid \boldsymbol{Y}_{\mathcal{I}}^{(n)}, \mathcal{S}) = \int_{(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2)} \pi(\boldsymbol{Y}_{\mathcal{I}^c}^{(n)} \mid f_{\mathcal{E},\boldsymbol{B}}, \sigma^2) \mathrm{d}\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) \tag{6.7}$$

where

$$\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}) = \pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}_{\mathcal{I}}^{(n)}, \mathcal{S}). \tag{6.8}$$

This simple decomposition unfolds new directions for ABC sampling based on data splitting. Instead of using all observations $\boldsymbol{Y}^{obs}$ to Accept/Reject each draw, we set aside a random subset of data $\boldsymbol{Y}_{\mathcal{I}^c}^{obs}$ for ABC rejection and use $\boldsymbol{Y}_{\mathcal{I}}^{obs}$ to "train the prior". The key observation is that the samples from the prior $\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$, i.e. the *posterior* $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}_{\mathcal{I}}^{(n)}, \mathcal{S})$, will have seen a part of the data and will produce more realistic guesses of f_0. Such guesses are more likely to yield pseudo-data that match $\boldsymbol{Y}_{\mathcal{I}^c}^{obs}$ more closely, thereby increasing the acceptance rate of ABC sampling. Note that the acceptance step is based solely on the left-out sample $\boldsymbol{Y}_{\mathcal{I}_m^c}^{obs}$, not the entire data. Similarly as the naive ABC outlined in the previous section, we first sample the subset $\mathcal{S}$ from the prior $\pi(\mathcal{S})$ and then obtain draws from the conditional marginal likelihood under an updated prior $\pi_{\mathcal{I}}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S})$. This corresponds to an ABC strategy for sampling from $\pi(\mathcal{S} \mid \boldsymbol{Y}_{\mathcal{I}^c}^{(n)})$ under the priors (6.2) and (6.8). As will be seen later, this posterior is effective for assessing variable importance. Moreover, if $\pi(\mathcal{S})$ is a good proxy for $\pi(\mathcal{S} \mid \boldsymbol{Y}_{\mathcal{I}}^{(n)})$ (when the training set is small relative to the ABC rejection set), this ABC will produce approximate samples from the original target $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$.

The idea of using a portion of the data for training the prior and the rest for model selection goes back to at least Good (1950). The most common prescription for choosing training samples in Bayesian analysis is to convert improper priors into propers ones for

meaningful model selection with Bayes factors (Lempers, 1971; O'Hagan, 1995). Berger and Pericchi (1996) advocated choosing the training set as small as possible subject to yielding proper posteriors (so called minimal training samples). Berger and Pericchi (2004) argue that data can vary widely in terms of their information content and the use of single minimal training samples can be inadequate/ suboptimal. Since there are many possible training samples, it is natural to average the resulting Bayes factors over the training samples in some fashion. While intrinsic Bayes factors (Berger and Pericchi, 1996) average Bayes factors over all possible minimal training samples, expected posterior priors (Pérez and Berger, 2002) average the prior first. In particular, the empirical expected-posterior prior for model $\mathcal{S}$ (Ghosh and Samanta, 2002; Pérez and Berger, 2002) writes as

$$\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S}) = \frac{1}{L} \sum_{l=1}^{L} \pi_{\mathcal{I}_l}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S}), \tag{6.9}$$

where $\pi_{\mathcal{I}_l}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \mathcal{S})$ was defined in (8) and where L is the number of all minimal training samples $\mathcal{I}_l$. The marginal likelihood under this prior can be then written as (equation (3.5) in Pérez and Berger (2002)) $m(\boldsymbol{Y}^{(n)} \,|\, \mathcal{S}) = \frac{1}{L} \sum_{l=1}^{L} \pi(\boldsymbol{Y}^{(n)} \,|\, \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S})$, where $\pi(\boldsymbol{Y}^{(n)} \,|\, \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S})$ was defined in (7). Our ABC analysis with internal data splitting can be thus regarded as arising from the empirical expected posterior prior (6.9). While the motivation for using training samples in Bayesian analysis has been largely to make improper priors proper, here we use this idea in a different context to increase ABC acceptance rate.

The ABC Bayesian Forests algorithm is formally summarized in Table 10. It starts by splitting the dataset into two subsets at each (m^{th}) iteration: $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ for fitting and $\boldsymbol{Y}^{obs}_{\mathcal{I}^c_m}$ for ABC rejection. The algorithm then proceeds by sampling an active set $\mathcal{S}$ from $\pi(\mathcal{S})$. Using the spike-and-slab construction, one can draw Bernoulli indicators $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)'$ where $\mathbb{P}(\gamma_j = 1 \,|\, \theta) = \theta$ for some prior inclusion probability $\theta \in (0, 1)$ and set $\mathcal{S}_m = \{j : \gamma_j = 1\}$. When sparsity is anticipated, one can choose θ to be small or to arise from a beta prior $\mathcal{B}(a, b)$

Algorithm 10: ABC Bayesian Forests

Data: Data $(Y_i^{obs}, \boldsymbol{x}_i)$ for $1 \leq i \leq n$
Result: $\pi_j(\epsilon)$ for $1 \leq j \leq p$ where $\pi_j(\epsilon) = \widehat{\mathbb{P}}(j \in \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)})$
Set M: the number of ABC simulations; s: the subsample size; ϵ: the tolerance threshold; $m = 0$ the counter
while $m \leq M$ **do**
  (a) Split data $\boldsymbol{Y}^{obs}$ into $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ and $\boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}$, where $\mathcal{I}_m \subset \{1, \ldots, n\}$ of size $|\mathcal{I}_m| = s$ is obtained by sampling with no replacement.
  (b) Pick a subset $\mathcal{S}_m$ from $\pi(\mathcal{S})$.
  (c) Sample $(f^m_{\mathcal{E},\boldsymbol{B}}, \sigma^2_m)$ from $\pi_{\mathcal{I}_m}(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \mathcal{S}_m) = \pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \mid \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}_m)$.
  (d) Generate pseudo-data $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ by sampling white noise $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_m)$ and setting $Y_i^{\star} = f^m_{\mathcal{E},\boldsymbol{B}}(\boldsymbol{x}_i) + \varepsilon_i$ for each $i \notin I_m$.
  (e) Compute discrepancy $\epsilon_m = \|\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c} - \boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}\|_2$.
  if $\epsilon_m < \epsilon$ **then**
    Accept $(\mathcal{S}_m, f^m_{\mathcal{E},\boldsymbol{B}})$ and set $m = m + 1$
  else
    Reject $(\mathcal{S}_m, f^m_{\mathcal{E},\boldsymbol{B}})$ and set $m = m + 1$
  end
end
Compute $\pi_j(\epsilon)$ as the proportion of times j^{th} variable is used in the accepted $f^m_{\mathcal{E},\boldsymbol{B}}$'s.

for some $a > 0$ and $b > 0$ (yielding the beta-binomial prior). We discuss other suitable prior model choices in Section 6.4.

In the *(c) step* of ABC Bayesian Forests, one obtains a sample from the posterior of $(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2)$, given $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$. For this step, one can leverage existing implementations of Bayesian CART and BART (e.g. the `BART` R package of McCulloch et al. (2018)). A single draw from the posterior is obtained after a sufficient burn-in. In this vein, one can view ABC Bayesian Forests as a computational envelope around BART to restrict the pool of available variables. The *(d) step* then consists of predicting the outcome $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ for left-out observations $\boldsymbol{x}_i$ using (6.1) for each $i \in \mathcal{I}_m^c$. The last step is ABC rejection based on the discrepancy between $\boldsymbol{Y}^{\star}_{\mathcal{I}_m^c}$ and $\boldsymbol{Y}^{obs}_{\mathcal{I}_m^c}$.

For the computation of marginal inclusion probabilities $\pi_j(\epsilon)$, one could conceivably report the proportion of ABC accepted samples $\mathcal{S}_m$ that contain the j^{th} variable. However, $\mathcal{S}_m$ is a pool of *available* predictors and not all of them are necessarily used in $f^m_{\mathcal{E},\boldsymbol{B}}$. Thereby,

we report the proportion of ABC accepted samples $f^m_{\mathcal{E},\boldsymbol{B}}$ that use the j^{th} variable at least once, i.e.

$$\pi_j(\epsilon) = \frac{1}{M(\epsilon)} \sum_{m:\epsilon_m<\epsilon} \mathbb{I}(j \text{ used in } f^m_{\mathcal{E},\boldsymbol{B}}), \tag{6.10}$$

where $M(\epsilon)$ is the number of accepted ABC samples at ϵ. Each tree ensemble $f^m_{\mathcal{E},\boldsymbol{B}}$ thus performs its own variable selection by picking variables from $\mathcal{S}_m$ rather than from $\{1,\ldots,p\}$. Limiting the pool of predictors prevents from too many false positives. In addition, the inclusion probabilities (6.10) do use the training data $\boldsymbol{Y}^{(n)}_{\mathcal{I}}$ to shrink and update the subset $\mathcal{S}$ by leaving out covariates not picked by $f^m_{\mathcal{E},\boldsymbol{B}}$. In this way, the mechanism for selecting the subsets $\mathcal{S}$ is not strictly sampling from the prior $\pi(\mathcal{S})$ but it seizes the information in the training set $\mathcal{I}$. In this way, $\mathcal{S}_m$'s can be regarded as approximate samples from $\pi(\mathcal{S} \mid \boldsymbol{Y}^{obs})$. When $\mathcal{I} = \emptyset$, we recover the naive ABC as a special case.

6.3.2.1 Dynamic ABC

The estimates of marginal inclusion probabilities $\pi_j(\epsilon)$ obtained with ABC Bayesian Forests unavoidably depend on the level of approximation accuracy ϵ. The acceptance threshold ϵ can be difficult to determine in practice, because it has to accommodate random variation of data around f_0 as well as the error when approximating smooth surfaces f_0 with trees. As $\epsilon \to 0$, the approximations $\pi_j(\epsilon)$ will be more accurate, but the acceptance rate will be smaller. It is customary to pick ϵ as an empirical quantile of ϵ_m (Grelaud et al., 2009), keeping only the top few closest samples. Rather than choosing one value ϵ, we suggest a dynamic strategy by considering a sequence of decreasing values $\epsilon_N > \epsilon_{N-1} > \cdots > \epsilon_1 > 0$. By filtering out the ABC samples with stricter thresholds, we track the evolution of each $\pi_j(\epsilon)$ as ϵ gets smaller and smaller. This gives us a dynamic plot that is similar in spirit to the Spike-and-Slab LASSO (Ročková and George, 2018) or EMVS (Ročková and George, 2014) coefficient evolution plots. However, our plots depict approximations to posterior inclusion probabilities rather than coefficient magnitudes. Other strategies for selecting the threshold

ϵ are discussed in Sunnaaker et al. (2013); Marin et al. (2012); Csillery et al. (2010).

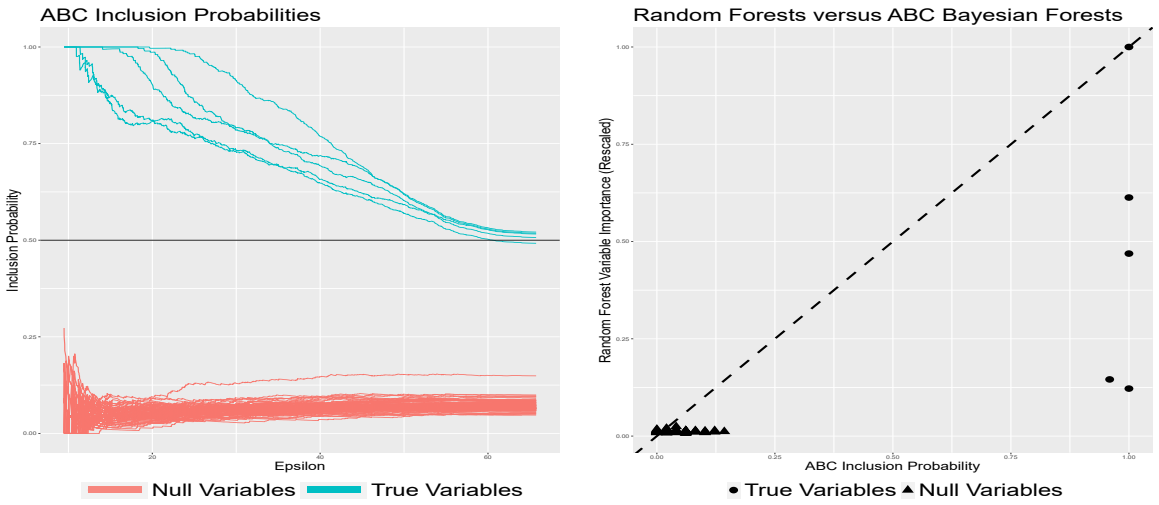
6.3.3 ABC Bayesian Forests in Action

We demonstrate the usefulness of ABC Bayesian Forests on the benchmark Friedman dataset (Friedman, 1991), where the observations are generated from (6.1) with $\sigma = 1$ and

$$f_0(\boldsymbol{x}_i) = 10\,\sin(\pi\, x_{i1}\, x_{i2}) + 20\,(x_{i3} - 0.5)^2 + 10\, x_{i4} + 5\, x_{i5}, \tag{6.11}$$

where $x_i \in [0,1]^p$ are *iid* from a uniform distribution on a unit cube. Because the outcome depends on $x_1, \ldots, x_p$, the predictors $x_6, \ldots, x_p$ are irrelevant, making it more challenging to find $f_0(\boldsymbol{x})$. We begin by illustrating the basic features of ABC Bayesian Forests with $p = 100$ and $n = 500$, assuming the beta-binomial prior $\pi(\mathcal{S} \,|\, \theta)$ with $\theta \sim \mathcal{B}(1,1)$ (see Section 6.3.2). At the m^{th} ABC iteration, we draw one posterior sample $f^m_{\mathcal{E},\boldsymbol{B}}$ after 100 burnin iterations using the BART MCMC algorithm (Chipman et al., 2001) with $T = 10$ trees. We generate $M = 1\,000$ ABC samples (with $s = n/2$) and we keep track of variables used in $f^m_{\mathcal{E},\boldsymbol{B}}$'s to estimate the marginal posterior inclusion probabilities $\pi_j(\epsilon)$. It is worth pointing out that unlike MCMC, ABC Bayesian Forests are embarrassingly parallel, making distributed implementations readily available.

Following the dynamic ABC strategy, we plot the estimates of posterior inclusion indicators $\pi_j(\epsilon)$ as a function of ϵ (Figure 6.1). The true signals are depicted in blue, while the noise covariates are in red. The estimated inclusion probabilities clearly segregate the active and non-active variables, even for large ϵ values. This is because BART itself performs variable selection to some degree, where not all variables in $\mathcal{S}_m$ end up contributing to $f^m_{\mathcal{E},\boldsymbol{B}}$. For small enough ϵ, the inclusion probabilities of true signals eventually cross the 0.5 threshold. Based on the median probability model rule (Barbieri and Berger, 2004), one thereby selects the true model when ϵ is sufficiently small. Because the inclusion probabilities get a bit



(a) $T = 10$ (b) Random Forests versus Bayesian Forests

Figure 6.1: (Left) Dynamic ABC plots for evolving inclusion probabilities as ϵ gets smaller. (Right) Plot of $\pi_j(\epsilon)$ obtained with ABC Bayesian Forests (ϵ is the 5% quantile of ϵ_m's) and the variable importance measure from Random Forests (rescaled to have a maximum at 1).

unstable as ϵ gets smaller (they are obtained from smaller reference tables), we excluded the 10 smallest ϵ values from the plot.

We repeated the experiment with more trees ($T = 50$) and a single tree ($T = 1$). Using more trees, one still gets the separation between signal and noise. However, many more noisy covariates would be included by the MPM rule. This is in accordance with Chipman et al. (2001) who state that BART can over-select with many trees. With a single tree, on the other hand, one may miss some of the low-signal predictors, where deeper trees and more ABC iterations would be needed to obtain a clearer separation.

In this simulation, we observe a curious empirical connection between $\pi_j(\epsilon)$, obtained with ABC Bayesian Forests (taking top 5% ABC samples), and rescaled variable importances obtained with Random Forests (RF). From Figure 6.1(b), we see that the two measures largely agree, separating the signal coefficients (triangles) from the noise coefficients (dots). However, the RF measure is a bit more conservative, yielding smaller normalized importance scores for true signals. While variable importance for RF is yet not understood theoretically,

in the next section we provide conditions under which the posterior distribution is consistent for variable selection.

6.4 Model-Free Variable Selection Consistency

In this section, we develop large sample model selection theory for spike-and-forest priors. As a jumping-off point, we first assume that α (the regularity of f_0) is known, where model selection essentially boils down to finding the active set $\mathcal{S}_0$. Later in this section, we investigate *joint* model selection consistency, acknowledging uncertainty about $\mathcal{S}_0$ and, at the same time, the regularity α.

Several consistency results for non-parametric regression already exist (Zhu et al., 2015; Yang and Pati, 2017). Comminges and Dalalyan (2012) characterized tight conditions on (n, p, q_0), under which it is possible to consistently estimate the sparsity pattern in two regimes. For fixed q_0, consistency is attainable when $(\log p)/n \leq c$ for some $c > 0$. When q_0 tends to infinity as $n \to \infty$, consistency is achievable when $c_1 q_0 + \log\log(p/q_0) - \log n \leq c_2$ for some $c_1, c_2 > 0$. Throughout this section, we will treat q_0 as fixed and show variable selection consistency when $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$. As an overture to our main result, we start with a simpler case when $T = 1$ (a single tree) and when α is known. The full-fledged result for Bayesian forests and unknown α is presented in Section 6.4.3. Throughout this section, we will assume $\sigma^2 = 1$.

6.4.1 The Case of Known α

Spike-and-forest mixture priors are constructed in two steps by (1) first specifying a conditional prior $\Pi_{\mathcal{S}}(f)$ on tree (ensemble) functions expressing a qualitative guess on f_0, and then (2) attaching a prior weight $\pi(\mathcal{S})$ to each "model" (i.e. subset) $\mathcal{S}$. The posterior distribution $\Pi(f \mid \boldsymbol{Y}^{(n)})$ can be viewed as a mixture of individual posteriors for various models $\mathcal{S}$

with weights given by posterior model probabilities $\Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$, i.e.

$$\Pi(f \mid \boldsymbol{Y}^{(n)}) = \sum_{\mathcal{S}} \Pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)}) \Pi_{\mathcal{S}}(f \mid \boldsymbol{Y}^{(n)}).$$

Our aim is to establish "model-free" variable selection consistency in the sense that

$$\Pi(\mathcal{S} = \mathcal{S}_0 \mid \boldsymbol{Y}^{(n)}) \to 1 \quad \text{in } \mathbb{P}_{f_0}^{(n)}\text{-probability} \quad \text{as} \quad n \to \infty,$$

where $\mathbb{P}_{f_0}^{(n)}$ is the distribution of $\boldsymbol{Y}^{(n)}$ under (6.1). The adjective "model-free" merely refers to the fact that we are selecting subsets in a non-parametric regression environment without necessarily committing to a linear model. We start by defining the model index set $\boldsymbol{\Gamma} = \big\{\mathcal{S} : \mathcal{S} \subseteq \{1, \ldots, p\}\big\}$, consisting of all 2^p variable subsets, and we partition it into (a) the true model $\mathcal{S}_0$, (b) models that *overfit* $\boldsymbol{\Gamma}_{\mathcal{S} \supset \mathcal{S}_0}$ (i.e. supersets of the true subset $\mathcal{S}_0$) and (c) models that *underfit* $\boldsymbol{\Gamma}_{\mathcal{S} \not\supset \mathcal{S}_0}$ (i.e. models that miss at least one active covariate). Each model $\mathcal{S} \in \boldsymbol{\Gamma}$ is accompanied by a convergence rate $\varepsilon_{n,\mathcal{S}}$ that reflects the inherent difficulty of the estimation problem. For each model $\mathcal{S}$ of size $|\mathcal{S}|$, we define

$$\varepsilon_{n,\mathcal{S}} = C_\varepsilon \, n^{-\alpha/(2\alpha + |\mathcal{S}|)} \sqrt{\log n} \quad \text{for some} \quad C_\varepsilon > 0, \tag{6.12}$$

the $\|\cdot\|_n$-near-minimax rate of estimation of a $|\mathcal{S}|$-dimensional α-smooth function.

6.4.1.1 Prior Specification

Prior distribution on the model index $\Pi(\mathcal{S})$ has to be chosen carefully for model selection consistency to hold when $p > n$ (Moreno et al., 2015). Traditional spike-and-slab priors introduce $\Pi(\mathcal{S})$ through a prior inclusion probability $\theta = \Pi(i \in \mathcal{S}_0 \mid \theta)$, independently for each $i = 1, \ldots, p$. This prior mixing weight is often endowed with a prior, such as the uniform prior $\pi(\theta) = \mathcal{B}(1, 1)$ (Scott and Berger, 2010), yielding a uniform prior on the model size, or

the "complexity prior" $\pi(\theta) = \mathcal{B}(1, p^c)$ for $c > 2$ (Castillo and van der Vaart, 2012), yielding an exponentially decaying prior on the model size. We propose a different approach, directly assigning a prior on model weights through

$$\pi(\mathcal{S}) \propto \mathrm{e}^{-C\left(n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}\log n \vee |\mathcal{S}|\log p\right)} \tag{6.13}$$

where $C > 0$ is a suitably large constant. When $|\mathcal{S}|\log p \leq n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}$, this prior is proportional to $\mathrm{e}^{-C/C_\varepsilon^2\, n\,\varepsilon_{n,\mathcal{S}}^2}$ and, as such, it puts more mass on models that yield faster rates convergence (similarly as in Lember and van der Vaart (2007)). When $|\mathcal{S}|\log p > n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}\log n$, the implied prior on the effective dimensionality $\pi(|\mathcal{S}|) = \binom{p}{|\mathcal{S}|}\pi(\mathcal{S})$ will be exponentially decaying in the sense that $\pi(|\mathcal{S}|) \lesssim \mathrm{e}^{-(C-1)|\mathcal{S}|\log p}$ for $C > 1$. It was recently noted by Castillo and Mismer (2018) that the complexity prior "penalizes slightly more than necessary". With our prior specification (6.13), however, the exponential decay kicks in *only* when $|\mathcal{S}|$ is sufficiently large.

Assuming that the level of smoothness α is known, the optimal number of steps (i.e. tree bottom leaves K) needed to achieve the rate-optimal performance for estimating f_0 should be of the order $n^{q_0/(2\alpha+q_0)} = 1/C_\varepsilon^2\, n\,\varepsilon_{n,\mathcal{S}_0}^2/\log n$ (Ročková and van der Pas, 2020). For our toy setup with a known α, we thus assume a point-mass prior on K with an atom near the optimal number of steps for each given $\mathcal{S}$, i.e.

$$\pi(K \mid \mathcal{S}) = \mathbb{I}[K = K_{\mathcal{S}}], \quad \text{where} \quad K_{\mathcal{S}} = \lfloor C_K/C_\varepsilon^2\, n\,\varepsilon_{n,\mathcal{S}}^2/\log n \rfloor \tag{6.14}$$

for some $C_K > 0$ such that $K_{\mathcal{S}_0} = 2^{q_0 s}$ for some $s \in \mathbb{N}$. In Section 6.4.2, we allow for more flexible trees with variable sizes.

6.4.1.2 Identifiability

The active variables ought to be sufficiently relevant in order to make their identification possible. To this end, we introduce a non-parametric signal strength assumption, making sure that f_0 is not too flat in active directions (Yang and Pati, 2017; Comminges and Dalalyan, 2012).

We first introduce the notion of an approximation gap. For any given model $\mathcal{S}$, we denote with $\mathcal{F}_{\mathcal{S}}$ a set of approximating functions (only single trees $f_{\mathcal{T},\boldsymbol{\beta}}$ with $K_{\mathcal{S}}$ leaves for now) and define the approximation gap as follows:

$$\delta_n^{\mathcal{S}} \equiv \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}} \|f_0 - f_{\mathcal{T},\boldsymbol{\beta}}\|_n = \|f_0 - f_{\widehat{\mathcal{T}},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}\|_n, \tag{6.15}$$

where $f_{\widehat{\mathcal{T}},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}$ is the $\|\cdot\|_n$-projection of f_0 onto $\mathcal{F}_{\mathcal{S}}$. For identifiability of $\mathcal{S}_0$, we require that those models that miss one of the active covariates have a large separation gap.

Definition 41. *(Identifiability) We say that $\mathcal{S}_0$ is (f_0, ε)-identifiable if, for some $M > 0$,*

$$\inf_{i \in \mathcal{S}_0} \delta_n^{\mathcal{S}_0 \setminus i} > 2M\varepsilon. \tag{6.16}$$

We provide a more intuitive explanation of (6.16) in terms of directional variability of f_0. The best approximating tree $f_{\widehat{\mathcal{T}},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}$ can be written as

$$f_{\widehat{\mathcal{T}},\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}(\boldsymbol{x}) = \sum_{k=1}^{K_{\mathcal{S}}} \mathbb{I}(\boldsymbol{x} \in \widehat{\Omega}_k^{\mathcal{S}}) \widehat{\beta}_k \text{ with } \widehat{\beta}_k = \bar{f}_0(\widehat{\Omega}_k^{\mathcal{S}}) \equiv \frac{1}{n(\widehat{\Omega}_k^{\mathcal{S}})} \sum_{\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}} f_0(\boldsymbol{x}_i),$$

where $\widehat{\mathcal{T}} = \{\widehat{\Omega}_k^{\mathcal{S}}\}_{k=1}^{K_{\mathcal{S}}}$ is the tree-shaped partition of the $\|\cdot\|_n$-projection of f_0 defined in (6.15) with $K_{\mathcal{S}}$ leaves and where $n(\widehat{\Omega}_k^{\mathcal{S}}) = \sum_{i=1}^n \mathbb{I}(\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}) \equiv n\,\mu(\widehat{\Omega}_k^{\mathcal{S}})$. The separation gap

in (6.15) can be then re-written as

$$\delta_n^{\mathcal{S}} = \sqrt{\sum_{k=1}^{K_{\mathcal{S}}} \mu(\widehat{\Omega}_k^{\mathcal{S}}) V[f_0 \,|\, \widehat{\Omega}_k^{\mathcal{S}}]},$$

where

$$V[f_0 \,|\, \widehat{\Omega}_k^{\mathcal{S}}] \equiv \frac{1}{n(\widehat{\Omega}_k^{\mathcal{S}})} \sum_{\boldsymbol{x}_i \in \widehat{\Omega}_k^{\mathcal{S}}} \Big(f_0(\boldsymbol{x}_i) - \bar{f}_0(\widehat{\Omega}_k^{\mathcal{S}}) \Big)^2$$

is the local variability of f_0 inside $\widehat{\Omega}_k^{\mathcal{S}}$. Given this characterization, (6.16) will be satisfied, for instance, when variability of f_0 inside best approximating cells that miss an active direction is too large, i.e. $\inf_{i \in \mathcal{S}_0} \inf_k V[f_0 \,|\, \widehat{\Omega}_k^{\mathcal{S}_0 \setminus i}] > 4M^2\,\varepsilon^2$.

Our identifiability condition is a theoretical assumption on f_0 which indicates how large signal in each direction should be in order to be capturable. It generalizes the more traditional sufficient "beta-min conditions" (Castillo et al., 2015; Zhao and Yu, 2006) for variable selection consistency (see Remark 44). Here, we gauge the amount of signal in terms of local variation in cells that *do not split* on an active covariate. Intuitively, if we do not split on $i \in \mathcal{S}_0$, the "variation" of f_0 inside the cells of the best tree we can get without i will be too large. The following example links our identifiability assumption with beta-min conditions.

Example 3. *Assume for now that $p = 2$ and that f_0 is linear, i.e.*

$$f_0(\boldsymbol{x}_i) = a + bx_{i1} + cx_{i2}.$$

Moreover, assume that $n = 16$ predictor observations are located on a regular grid $\mathcal{X} = \{k/4 : 1 \leq k \leq 4\} \times \{j/4 : 1 \leq j \leq 4\}$, where $\times$ denotes the Cartesian product. Suppose $\mathcal{S}_0 = \{1, 2\}$ and set $\mathcal{S} = \mathcal{S}_0 \setminus \{2\} = \{1\}$ and $K_{\mathcal{S}} = 2$. It can be verified that the partition $\widehat{\mathcal{T}}$ of the best approximating tree that does not *split on the covariate x_2 consists of two rectangles*

$\widehat{\Omega}_1^{\mathcal{S}} = [0, 1/2) \times [0, 1]$ *and* $\widehat{\Omega}_2^{\mathcal{S}} = [1/2, 1] \times [0, 1]$. *Then we have*

$$\bar{f}_0(\widehat{\Omega}_1^{\mathcal{S}}) = a + \frac{3}{2}\left(\frac{b}{4}\right) + \frac{5}{2}\left(\frac{c}{4}\right) \quad \textit{and} \quad \bar{f}_0(\widehat{\Omega}_2^{\mathcal{S}}) = a + \frac{7}{2}\left(\frac{b}{4}\right) + \frac{5}{2}\left(\frac{c}{4}\right)$$

and thereby

$$(\delta_m^{\mathcal{S}})^2 = V(f_0|\widehat{\Omega}_1^{\mathcal{S}}) = V(f_0|\widehat{\Omega}_2^{\mathcal{S}}) = \frac{1}{4}\frac{b^2}{16} + \frac{5}{4}\frac{c^2}{16}. \tag{6.17}$$

From the expression (6.17) *we can immediately see the connection to the beta-min conditions. When the signal in the direction of* x_2 *is large enough, i.e.* $c > 16/\sqrt{5}M\varepsilon$, *our identifiability condition will be satisfied.*

The second sufficient condition needed for methods such as the LASSO to fully recover $\mathcal{S}_0$ is "irrepresentability" (Zhao and Yu, 2006; Van De Geer and Bühlmann, 2009). This condition restricts the amount of correlation between (active and non-active) covariates by imposing a regularization constraint on the magnitudes of regression coefficients of the inactive predictors onto the active ones. Here, we generalize the notion of irrepresentability to the non-parametric setup. Consider an underfitting model $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \not\supset \mathcal{S}_0$, where $\mathcal{S}_1 \subset \mathcal{S}_0$ are true positives and $\mathcal{S}_2$ is a possibly empty set of false positives, i.e. $\mathcal{S}_2 \cap \mathcal{S}_0 = \emptyset$. We define

$$\rho_n^{\mathcal{S}} \equiv \frac{1}{n}\sum_{i=1}^{n}[f_0(\boldsymbol{x}_i) - f_{T,\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}(\boldsymbol{x}_i)][f_{T,\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}(\boldsymbol{x}_i) - f_{T,\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}(\boldsymbol{x}_i)], \tag{6.18}$$

the sample covariance between the surplus signals in f_0 and $f_{T,\widehat{\boldsymbol{\beta}}}^{\mathcal{S}}$ obtained by removing the effect of $f_{T,\widehat{\boldsymbol{\beta}}}^{\mathcal{S}_1}$. This quantity will be large if noise covariates inside $\mathcal{S}_2$ can compensate for the missed true covariates in $\mathcal{S}_0 \backslash \mathcal{S}_1$, i.e. when the true and fake covariates are strongly correlated. To obviate this substitution effect, we introduce the following nonparametric "irrepresentability"condition. Similarly as in Zhao and Yu (2006), we require that "the total amount of an irrelevant covariate represented by the covariates in the true model" is small.

Definition 42. *(Irrepresentability) We say that ε-irrepresentability holds for f_0 and $\mathcal{S}_0$ if, for some $M > 0$, we have $\sup_{\mathcal{S} \not\supset \mathcal{S}_0} |\rho_n^{\mathcal{S}}| < \frac{M}{2}\varepsilon$, where $\rho_n^{\mathcal{S}}$ was defined in* (6.18).

It follows from Lemma 50 that under the irrepresentability and identifiability conditions (Definition 41 and 42), we obtain

$$\inf_{\mathcal{S} \not\supset \mathcal{S}_0} \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}} \|f_{\mathcal{T},\boldsymbol{\beta}} - f_0\|_n > M\,\varepsilon. \tag{6.19}$$

This condition essentially states that *all* models that miss *at least one* active covariate (i.e. not only subsets of the true model) have a large separation gap.

The following theorem characterizes variable selection consistency of spike-and-tree posterior distributions. Namely, the posterior distribution over the model index is shown to concentrate on the true model $\mathcal{S}_0$. One additional assumption is needed to make sure that the (fixed) design $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is sufficiently regular. Ročková and van der Pas (2020) define the notion of a fixed $\mathcal{S}_0$-regular design in terms of cell diameters of a k-d tree partition (Definition 3.3). This assumption essentially excludes outliers, making sure that the data cloud is spread evenly in active directions (while permitting correlation between covariates).

Theorem 43. *Assume $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ for some $\alpha \in (0, 1]$ and $\mathcal{S}_0 \subset \{1, \ldots, p\}$ with $q_0 = |\mathcal{S}_0|$ and $\|f_0\|_\infty \lesssim B$. Denote with $\widetilde{\varepsilon}_n = C_\varepsilon\, n^{-\alpha/(2\alpha+q_n)}\sqrt{\log n}$, where $q_n = C_q\lceil n\,\varepsilon_{n,\mathcal{S}_0}^2/\log p\rceil$ for some $C_q > 0$, and assume $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$ with $2 \leq q_0 = \mathcal{O}(1)$ as $n \to \infty$. Assume that (a) $\mathcal{S}_0$ is $(f_0, \widetilde{\varepsilon}_n)$-identifiable, (b) $\widetilde{\varepsilon}_n$-irrepresentability holds and that (c) the design $\mathcal{X}$ is $\mathcal{S}_0$-regular. Under the spike-and-tree prior comprising (with $T = 1$)* (6.4),(6.5),(6.13) *with $C > 2$ and* (6.14)*, we have*

$$\Pi[\mathcal{S} = \mathcal{S}_0 \,|\, \boldsymbol{Y}^{(n)}] \to 1 \quad \textit{in } \mathbb{P}_{f_0}^{(n)}\textit{-probability as } n \to \infty.$$

Proof. Section 6.8.1.1.

Remark 44. *The assumption of $(f_0, \widetilde{\varepsilon}_n)$-identifiability pertains to the more traditional sufficient beta-min conditions for variable selection consistency in sparse high-dimensional models. For example, Castillo et al. (2015) in their Corollary 1 require that* $\min_{i \in \mathcal{S}_0} |\beta_i^0| \geq M\sqrt{\frac{q_0 \log p}{n}}$, *for some "large enough constant" $M > 0$ that depends on the compatibility number (see e.g. Definition 2.1 in Castillo et al. (2015) of the design matrix X (rescaled to have an $\|\cdot\|_2$ norm $\sqrt{n}$). Our identifiability threshold also depends on the rate of convergence ε_n (similarly as in Castillo et al. (2015)). However, unlike in the linear models we measure the signal strength in a non-parametric way. Lastly, note that the identifiability gap $\widetilde{\varepsilon}_n$ in Theorem 43 is a bit larger than the near-minimax rate $\varepsilon_{n,\mathcal{S}_0}$. This requirement will be relaxed in the next section, where α will be treated as unknown.*

For *iid* models, Ghosal et al. (2008) considered the problem of nonparametric Bayesian model selection and averaging and characterized conditions under which the posterior achieves adaptive rates of convergence. The authors also study the posterior distribution of the model index, showing that it puts a negligible weight on models that are bigger than the optimal one. Yang and Pati (2017) characterized similar conditions for the non-*iid* case, see Section 6.8.1.1 for more details.

Remark 45. *(Theory for ABC) It is worth pointing out that Theorem 43 is obtained for the* actual *posterior $\pi(\mathcal{S} \mid \boldsymbol{Y}^{(n)})$, not the ABC posterior. Theory for ABC recently started emerging with the first results focussing on ABC bias (Barber et al., 2015), consistency and asymptotic normality (Martin et al., 2014; Frazier et al., 2018, 2019) and on convergence of the posterior mean (Li and Fearnhead, 2018). For our non-parametric regression scenario, we can conclude (variable selection) consistency for ABC Bayesian forests under the assumption that the residual variance σ^2 decreases with the sample size (as is typical in the Gaussian sequence model). In particular, Theorem 52 in Section 6.8.1.4 shows that the ABC posterior concentrates at the rate $\lambda_n = 4\epsilon_n^T/3 + 1/\sqrt{n}$, where $\epsilon_n^T = \sqrt{2 \log n/n}$ is the ABC tolerance level.*

This result implies that the ABC posterior will not *reward underfitting model as long as our identifiability and irrepresentability conditions are satisfied with* $\varepsilon = \lambda_n$. *Regarding over-fitting models, an ABC analogue of Lemma 49 in Section 6.8.1.1.2 implies that the ABC posterior probability of over-fitting models goes to zero, which concludes variable selection consistency of a (naive) ABC method. These considerations can be extended to ABC Bayesian Forests with data splitting using the empirical expected posterior prior justification in* (6.9). *More details are in Section 6.8.1.4.*

Remark 46. *(Consistency of the Median Probability Model) In Section 6.3.3, we used the median probability model rule which may not the same as the highest-posterior model whose consistency we have shown in Theorem 43. However, even when* $p \to \infty$ *it can be verified (as in Corollary 4.1 in Narisetty and He (2014)) that the median probability model is* also *consistent under the same assumptions as Theorem 43. In particular,* $\mathbb{P}_{f_0}^{(n)}[\cap_{i=1}^{p} E_i] \to 1$ *as* $n \to \infty$ *where* $E_i = \{\Pi(\gamma_i = \gamma_i^0 \mid \boldsymbol{Y}^{(n)}) > 0.5\}$ *and where* $\gamma_i = \mathbb{I}(i \in \mathcal{S})$ *are binary inclusion indicators and* $\gamma_i^0 = \mathbb{I}(i \in \mathcal{S}_0)$.

6.4.2 The Case of Unknown α

The fact that the level α has to be known for the consistency to hold makes the result in Theorem 43 somewhat theoretical. In this section, we provide a joint consistency result for the unknown regularity level K and, at the same time, the unknown subset $\mathcal{S}_0$. Finding the optimal regularity level K, given $\mathcal{S}_0$, is a model selection problem of independent interest (Lafferty and Wasserman, 2001). Here, we acknowledge uncertainty about *both* K and $\mathcal{S}_0$ by assigning a joint prior distribution on $(K, \mathcal{S})$. Namely, we consider an analogue of (6.13), where $n^{|\mathcal{S}|/(2\alpha+|\mathcal{S}|)}$ is now replaced with $K \log n$ (according to (6.14)), i.e.

$$\pi(K, \mathcal{S}) \propto \mathrm{e}^{-C(K \log n \vee |\mathcal{S}| \log p)} \quad \text{for} \quad 1 \leq K \leq n \quad \text{and} \quad \mathcal{S} \subseteq \{1, \ldots, p\}. \tag{6.20}$$

This prior penalizes models with too many splits or too many covariates. We now regard each model as a *pair of indices* $(K, \mathcal{S})$, where the "true" model is characterized by $\mathbf{\Gamma}_0 = (K_{\mathcal{S}_0}, \mathcal{S}_0)$ with $K_{\mathcal{S}_0}$ defined in (6.14). Again, we partition the model index set $\mathbf{\Gamma} = \{(K, \mathcal{S}) : \mathcal{S} \subseteq \{1, \dots, p\}, 1 \leq K \leq n\}$ into (a) the true model $\mathbf{\Gamma}_0$, (b) models that underfit $\mathbf{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{K < K_{\mathcal{S}_0}\}}$ (i.e. miss at least one covariate *or* use less than the optimal number of splits), and (c) models that overfit $\mathbf{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{K \geq K_{\mathcal{S}_0}\}}$ (i.e. use too many variables and splits).

We combine the identifiability and irrepresentability conditions into one as follows:

$$\inf_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{K < K_{\mathcal{S}_0}\}} \inf_{f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}}(K)} \|f_{\mathcal{T},\boldsymbol{\beta}} - f_0\|_n > M \, \varepsilon_{n,\mathcal{S}_0} \tag{6.21}$$

for some $M > 1$, where $\mathcal{F}_{\mathcal{S}}(K)$ consists of all trees with K bottom leaves and splitting variables $\mathcal{S}$. This condition is an analogue of (6.19), essentially stating that one cannot approximate f_0 with an error smaller than a multiple of the near-minimax rate using underfitting models.

Theorem 47. *Assume $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ for some $\alpha \in (0, 1]$ and $\mathcal{S}_0 \subset \{1, \dots, p\}$ such that $|\mathcal{S}_0| = q_0$ and $\|f_0\|_\infty \lesssim B$. Assume $q_0 \log p \leq n^{q_0/(2\alpha + q_0)}$ and $2 \leq q_0 = \mathcal{O}(1)$ as $n \to \infty$. Furthermore, assume that the design $\mathcal{X}$ is $\mathcal{S}_0$-regular and that* (6.21) *holds. Under the spike-and-tree prior comprising (with $T = 1$)* (6.4), (6.5) *and* (6.20) *for $C > 3$, we have*

$$\Pi\left[\{\mathcal{S} = \mathcal{S}_0\} \cap \{K_{\mathcal{S}_0} \leq K \leq K_n\} \,\middle|\, \boldsymbol{Y}^{(n)}\right] \to 1 \quad \text{in } \mathbb{P}_{f_0}^{(n)}\text{-probability as } n \to \infty,$$

where $K_{\mathcal{S}_0}$ was defined in (6.14) *and $K_n = \lceil \bar{C} \, n \, \varepsilon_{n,\mathcal{S}_0}^2 / \log n \rceil$ for some $\bar{C} > C_K / C_\varepsilon^2$.*

Proof. Section 6.8.1.2.

Note that both $K_{\mathcal{S}_0}$ and K_n are of the same (optimal) order, where the marginal posterior distribution $\Pi(K \mid \boldsymbol{Y}^{(n)})$ squeezes inside these two quantities as $n \to \infty$. Lafferty and

Wasserman (2001) provide a similar result for their RODEO method, without the variable selection consistency part. Yang and Pati (2017) also provide a similar result for Gaussian processes, without the regularity selection consistency part. Here, we characterize *joint* consistency for both subset and regularity model selection.

6.4.3 Variable Selection Consistency with Bayesian Forests

Finally, we provide a variant of Theorem 47 for tree ensembles. Each Bayesian forest (i.e. additive regression tree) model is characterized by a triplet $(\mathcal{S}, T, \boldsymbol{K})$, where $\mathcal{S}$ is the active variable subset, $T \in \mathbb{N}$ is the number of trees and $\boldsymbol{K} = (K^1, \ldots, K^T)' \in \mathbb{N}^T$ is a vector of the bottom leave counts for the T trees. Rate-optimality of Bayesian forests can be achieved for a wide variety of priors, ranging from many weak learners (large T and small K^t's) to a few strong learners (small T and large K^t's) (Ročková and van der Pas, 2020). The optimality requirement is that the *total* number of leaves in the ensemble $\sum_{t=1}^{T} K^t$ behaves like $K_{\mathcal{S}_0}$, defined earlier in (6.14).

We thereby define models in terms of equivalence classes rather than individual triplets $(\mathcal{S}, T, \boldsymbol{K})$. We construct each equivalence class $E(Z)$ by combining ensembles with the same number Z of total leaves, i.e.

$$E(Z) = \bigcup_{T=1}^{\min\{Z,n\}} \left\{ \boldsymbol{K} \in \mathbb{N}^T : \sum_{t=1}^{T} K^t = Z \right\}. \tag{6.22}$$

The cardinality of $E(Z)$, denoted with $\Delta(E(Z))$, satisfies $\Delta(E(Z)) \leq Z!\, p(Z)$, where $p(Z)$ is the partitioning number (i.e. the number of ways one can write Z as a sum of positive integers). The "true" model $\boldsymbol{\Gamma}_0 = (\mathcal{S}_0, E(K_{\mathcal{S}_0}))$ consists of an equivalence class of forests that split on variables inside $\mathcal{S}_0$ with a total number of $K_{\mathcal{S}_0}$ leaves. Similarly as before, we define underfitting model classes $\boldsymbol{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{E(Z): Z < K_{\mathcal{S}_0}\}}$ and overfitting model classes $\boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{E(Z): Z \geq K_{\mathcal{S}_0}\}}$. Regarding the prior on T, similarly as Ročková and van der Pas

(2020), we consider

$$\pi(T) \propto \mathrm{e}^{-C_T\, T}, \quad T = 1, \ldots, n, \quad \text{for} \quad C_T > 0. \tag{6.23}$$

Given T, we assign a joint prior over $\mathcal{S}_0$ and $\boldsymbol{K} \in \mathbb{N}^T$ as follows:

$$\pi(\mathcal{S}, \boldsymbol{K} \mid T) \propto \mathrm{e}^{-C\, \max\left\{|\mathcal{S}| \log p\,;\, \sum_{t=1}^{T} K^t \log n\right\}} \quad \text{for} \quad C > 1. \tag{6.24}$$

We conclude this section with a model selection consistency result for Bayesian forests under the following identifiability condition

$$\inf_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{E(Z) : Z < K_{\mathcal{S}_0}\}} \; \inf_{f_{\mathcal{E},\boldsymbol{B}} \in \mathcal{F}_{\mathcal{S}}(\boldsymbol{K})} \|f_{\mathcal{E},\boldsymbol{B}} - f_0\|_n > M\, \varepsilon_{n,\mathcal{S}_0}, \tag{6.25}$$

where $\mathcal{F}_{\mathcal{S}}(\boldsymbol{K})$ denotes all forests $f_{\mathcal{E},\boldsymbol{B}}$ that split on variables $\mathcal{S}$ and consist of T trees with $\boldsymbol{K} = (K^1, \ldots, K^T)'$ bottom leaves.

Theorem 48. *Assume $f_0 \in \mathcal{H}_p^\alpha \cap \mathcal{C}(\mathcal{S}_0)$ for some $\alpha \in (0, 1]$ and $\mathcal{S}_0 \subset \{1, \ldots, p\}$ such that $|\mathcal{S}_0| = q_0$ and $\|f_0\|_\infty \lesssim B$. Assume $q_0 \log p \leq n^{q_0/(2\alpha + q_0)}$, where $2 \leq q_0 = \mathcal{O}(1)$ as $n \to \infty$. Furthermore, assume that the design is $\mathcal{S}_0$-regular and that* (6.25) *holds. Under the spike-and-forest prior comprising* (6.4), (6.5), (6.23) *and* (6.24)*, we have*

$$\Pi\left[\{\mathcal{S} = \mathcal{S}_0\} \cap \left\{K_{\mathcal{S}_0} \leq \sum_{t=1}^{T} K^t \leq K_n\right\} \,\middle|\, \boldsymbol{Y}^{(n)}\right] \to 1 \quad \textit{in } \mathbb{P}_{f_0}^{(n)}\textit{-probability as } n \to \infty,$$

where $K_{\mathcal{S}_0}$ was defined in (6.14) *and $K_n = \lceil \bar{C}\, n\, \varepsilon_{n,\mathcal{S}}^2 / \log n \rceil$ for some $\bar{C} > C_K / C_\varepsilon^2$.*

Proof. Section 6.8.1.3.

6.5 Simulation Study

We evaluate the performance of ABC Bayesian Forests on simulated data. We consider the following performance criteria: Precision $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, Power $= \frac{\text{TP}}{\text{TP+FN}}$(defined as the proportion of true signals discovered as such), Hamming Distance (HD)$=$ FP$+$FN (where FP and FN denotes the number of false positives and false negatives, respectively) and the area under the ROC curve (AUC). Traditionally, AUC assesses how well a classification method can differentiate between two classes in the absence of a clear decision boundary. We use this criterion to assess variable importance since many of the considered selection methods are based on an importance measure and, as such, do not have a clear decision boundary.

The synthetic data are generated from the model (6.1), where $\boldsymbol{x}_i$'s for $i = 1, \ldots, n$ are drawn independently from $N_p(0, \Sigma)$ with $\Sigma = (\rho_{ij})_{i,j=1}^{p,p}$. We make our comparisons under different combinations of f_0, σ and Σ. In particular, we consider a relatively large noise level with $\sigma = 5$ ($\sigma = \sqrt{5}$ for the linear setup) and

1. medium equi-correlation $\rho_{ij} = 0.5$ for $i \neq j$ with $\rho_{ii} = 1$,

2. high auto-correlation $\rho_{ij} = 0.9^{|i-j|}$.

Regarding the mean function f_0, we consider four choices: (1) a linear setup with $f_0(\boldsymbol{x}_i) = x_{i1} + 2x_{i2} + 3x_{i3} - 2x_{i4} - x_{i5}$; (2) the Friedman setup as described in (6.11); (3) a CART (tree-based) function $f_0(\boldsymbol{x}_i)$ generated from the first 5 covariates using the `rpart` function in R; (4) a simulated example from Liang et al. (2018) (denoted with LLS hereafter) with $f_0(\boldsymbol{x}_i) = \frac{10x_{i2}}{1+x_{i1}^2} + 5\sin(x_{i3}x_{i4} + 2x_{i5})$. For the auto-correlation case, we permuted the covariates so that signals are not next to each other.

For each combination of settings, we repeat our simulation over 20 different datasets assuming $n = 500$ and $p \in \{100, 1\,000\}$. We compare ABC Bayesian Forests with Random Forests (RF), Dynamic Trees (DT) of Taddy et al. (2011b), BART (Chipman et al., 2010),

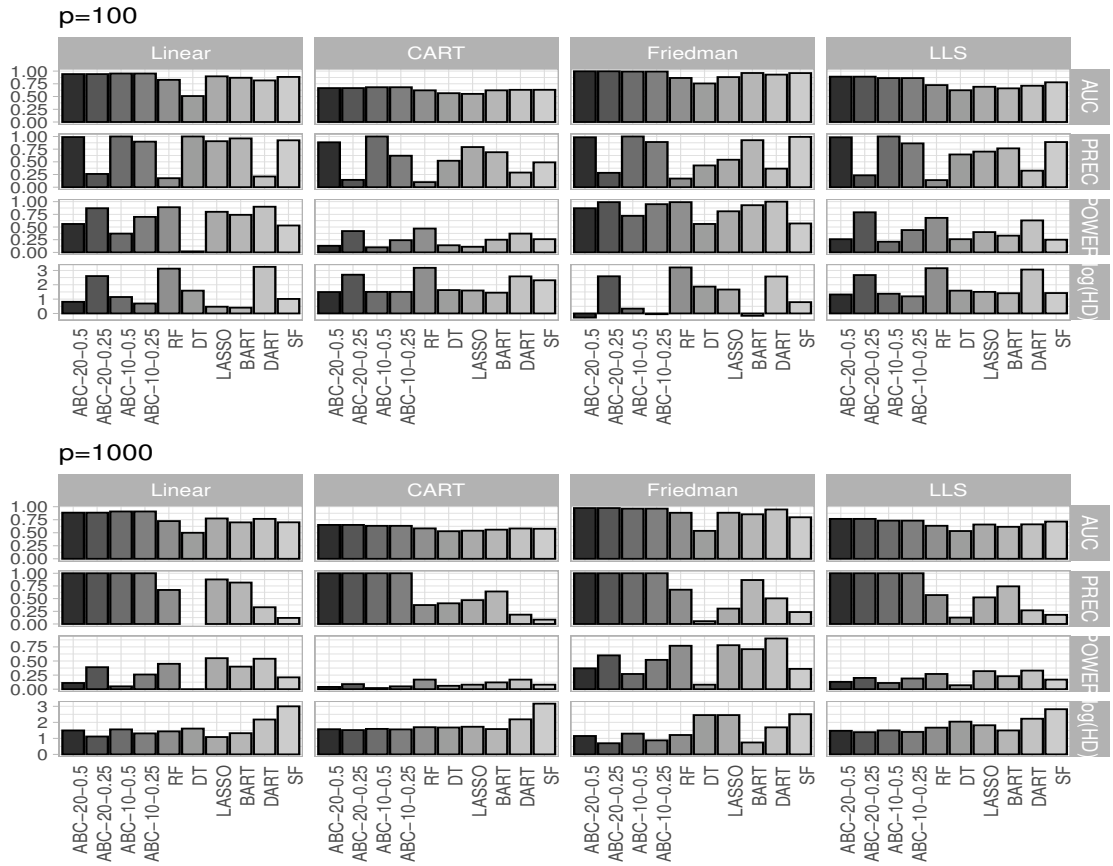


Figure 6.2: Average variable selection performance under equicorrelation $\rho_{ij} = 0.5$ over 20 simulations. Each panel corresponds to a different dimension $p \in \{100, 1000\}$. Each row reports a different statistic: AUC is the area under the ROC curve, PREC $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, POWER $= \frac{\text{TP}}{\text{TP+FN}}$, $\log(\text{HD}) = \log(\text{FP} + \text{FN})$. ABC is run for $T \in \{10, 20\}$ and cutoff $\in \{0.5, 0.25\}$.Each column indicates a different data generating process.

DART of Linero (2018), LASSO and Spike-and-Forests (the MCMC counterpart of ABC Bayesian Forests outlined in Section 6.8.2). ABC Bayesian Forests are trained with $M =$ 1 000 ABC samples, where only a fraction of ABC samples (top 10%) are kept in the reference table. The prior $\pi(\mathcal{S})$ is the usual beta-binomial prior with $\theta \sim \mathcal{B}(1, 1)$. Inside each ABC step, we sample a subset of size $s = n/2$ and draw a tree ensemble using the default Bayesian CART prior (Chipman et al., 1998) and $T \in \{10, 20\}$ trees. For each ABC sample, we draw the last BART sample after $B = 200$ burnin MCMC iterations. A sensitivity analysis to the choice s, T, B and M is reported in the Supplemental Materials (Section 4). Two versions of

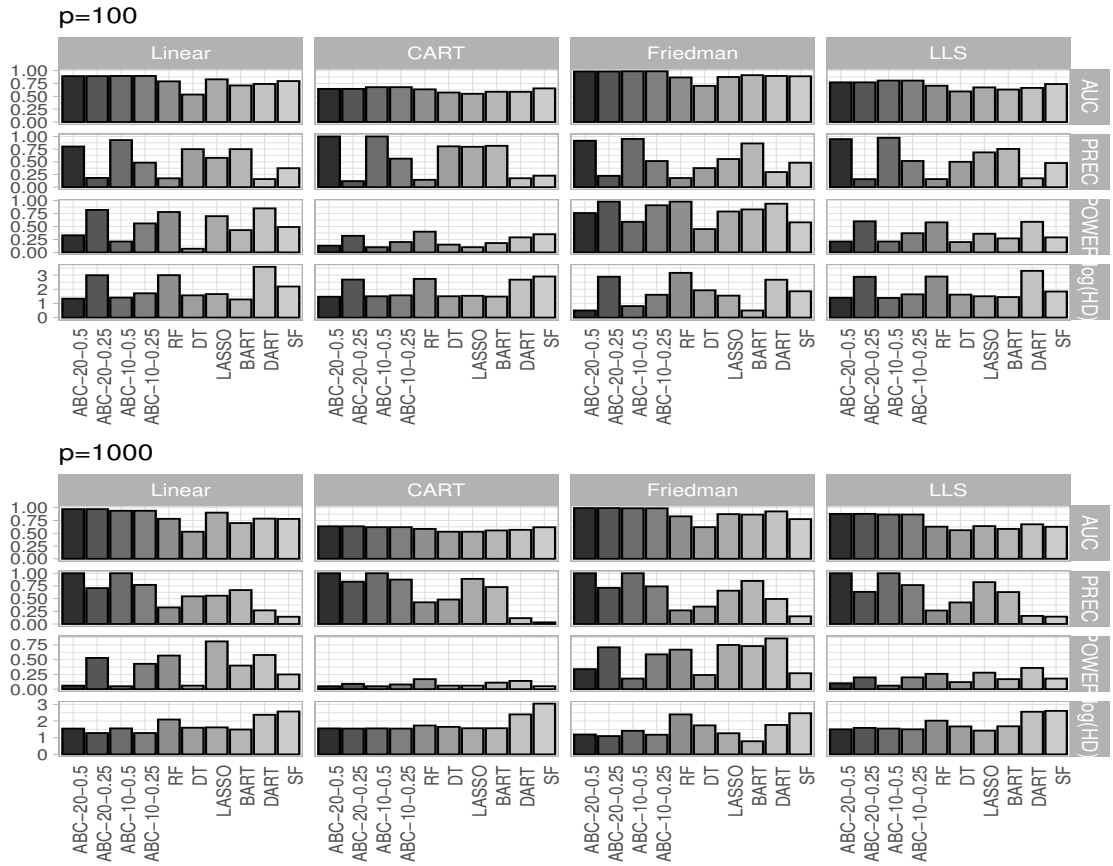


Figure 6.3: Average variable selection performance under autocorrelation $\rho_{ij} = 0.9^{|i-j|}$ over 10 simulations. Each panel corresponds to a different dimension $p \in \{100, 1000\}$. Each row reports a different statistics: AUC is the area under the ROC curve, PREC $= 1 - \text{FDP} = \frac{\text{TP}}{\text{TP+FP}}$, POWER $= \frac{\text{TP}}{\text{TP+FN}}$, $\log(\text{HD}) = \log(\text{FP} + \text{FN})$. ABC is run for $T \in \{10, 20\}$ and cutoff $\in \{0.5, 0.25\}$. Each column indicates a different data generating process.

BART (without ABC) were deployed using the `R` package `BART`: (1) the standard BART from Chipman et al. (2010) with $T = 20$ (as recommended in Bleich et al. (2014)), and (2) the sparse version DART of Linero (2018) with a Dirichlet prior (`sparse=TRUE, a=0.5, b=1`) with $T = 200$. Both versions are run with 10 000 MCMC samples after 10 000 burn-in. For LASSO, we use the `glmnet` package in `R` (Friedman et al., 2010) using the 1-se rule to select the penalty λ. For Random Forests, we deploy the `randomForest` package in `R` (Liaw and Wiener, 2002) using the default number of 500 trees where variable importance is based on

the difference in predictions (with and without each covariate) in out-of-bag samples.

To select variables with random forests, there are at least three commonly used strategies: (1) Recursive Feature Elimination (RFE) implemented in the `caret` package with 5-fold cross-validation (as suggested in Linero (2018)); (2) truncating importance at the $1 - \alpha$ quantile of a standard normal distribution (as suggested by Breiman and Cutler (2013)); (3) truncating importance at the Bonferroni-corrected $(1 - \alpha/p)$ quantile of a standard normal distribution (Bleich et al., 2014). We report the third method, which was seen to perform the best. For BART and DART, we select those variables which have been split on inside a forest at least once on average. Alternative strategies based on truncating inclusion probabilities (Linero, 2018) using data-adaptive thresholds (Bleich et al., 2014) did not perform better, in general. For ABC, we report results for two selection thresholds 0.5 and 0.25. For Spike-and-Forest (SF), we report the median probability model.

The performance comparisons for variable selection are summarized in Figure 6.2 (equi-correlation $\rho_{ij} = 0.5$) and Figure 6.3 (autocorrelation $\rho_{ij} = 0.9^{|i-j|}$). These figures show that ABC has an advantage in terms of AUC, suggesting that ABC can rank variables more efficiently. While RF tend to have a higher power, they are plagued with false discoveries (i.e. smaller precision). ABC Bayesian Forests, on the other hand, are seen to yield fewer false discoveries (i.e. higher precision) relative to the other procedures. The ABC threshold 0.5 yields higher precision whereas 0.25 yields higher power.

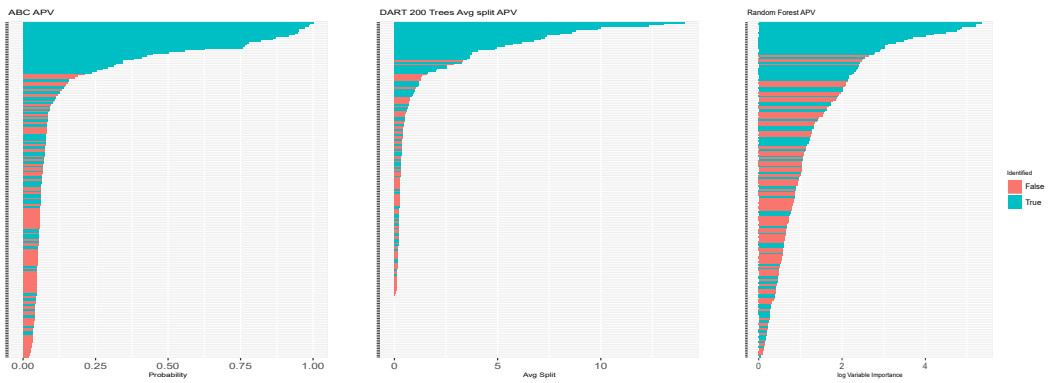
While ABC Bayesian Forests were designed to explore the posterior distribution over models, it is natural to ask whether they also yield reasonable prediction. There are various ways to perform prediction with our ABC method. One natural strategy is to save each draw $f^m_{\mathcal{S},\boldsymbol{B}}$ at the m^{th} ABC iteration when $\epsilon_m < \epsilon$ and average out individual predictions obtained from these single draws. Alternatively, one could first select variables based on ABC Bayesian Forests and then run a separate BART method (using the default number of $T = 200$ trees which is recommended for prediction) with the selected variables. Using both

	ABC2 $T=20$	**ABC1** $T=20, c=0.5$	**ABC1** $T=20, c=0.25$	**ABC2** $T=10$	**ABC1** $T=10, c=0.5$	**ABC1** $T=10, c=0.25$	**RF**	**RLT**	**DT**	**BART**	**DART**
					Equi-correlation $\rho_{ij}=0.5$ **for** $i \neq j$						
					Linear						
$p=100$	5.56	5.58	5.84	5.60	5.84	5.55	5.63	5.45	5.92	5.49	**5.40**
$p=1\,000$	5.79	6.15	5.73	5.86	6.28	5.95	5.83	5.70	6.04	5.82	**5.62**
					CART						
$p=100$	34.21	34.63	37.19	**34.00**	36.10	35.81	34.21	34.64	34.61	35.48	35.57
$p=1\,000$	32.00	34.27	35.72	**31.99**	33.93	33.17	32.30	32.40	33.08	33.77	34.04
					Friedman						
$p=100$	30.32	29.28	31.59	30.52	30.30	**29.03**	31.84	30.17	41.41	31.31	**29.03**
$p=1\,000$	33.14	35.97	31.54	33.54	38.42	32.71	34.35	32.22	45.69	32.99	**29.42**
					LLS						
$p=100$	**26.23**	27.00	28.70	26.25	26.90	27.36	26.80	26.46	28.51	27.42	27.42
$p=1\,000$	27.37	26.98	**26.94**	27.38	27.07	27.02	27.18	26.68	30.66	28.21	27.49
					Auto-correlation $\rho_{ij}=0.9^{\lvert i-j \rvert}$						
					Linear						
$p=100$	6.17	6.29	6.37	6.20	6.25	6.18	6.37	6.09	6.77	6.17	**5.91**
$p=1\,000$	6.39	6.44	**6.00**	6.47	6.21	6.13	6.55	6.20	7.06	6.53	6.42
					CART						
$p=100$	33.80	37.72	37.28	33.83	36.78	36.61	**33.57**	34.40	35.05	35.61	35.81
$p=1\,000$	31.57	33.55	37.21	**31.52**	33.52	37.43	31.63	31.88	32.22	33.11	33.43
					Friedman						
$p=100$	34.09	32.51	34.65	34.27	34.97	32.77	36.88	33.83	48.64	34.21	**30.36**
$p=1\,000$	39.09	39.57	32.58	40.58	43.05	33.46	41.80	37.38	49.51	35.96	**30.81**
					LLS						
$p=100$	28.57	**27.94**	30.71	28.45	28.03	29.12	28.88	27.87	30.69	28.83	28.81
$p=1\,000$	29.98	**28.25**	28.96	30.14	28.40	28.38	30.19	28.56	32.29	31.76	29.28

Table 6.1: Average out-of-sample mean squared prediction error over 20 independent validation datasets. ABC1 denotes predictions using ABC samples $f^m_{\mathcal{S},\boldsymbol{B}}$ and ABC2 uses ABC variable selection and runs BART ($T = 200$) on the selected subset. T designates the number of trees and c is the selection threshold. The best performing method for each row is denoted in bold.

strategies, we report average out-of-sample mean squared prediction error, where the average is taken over 20 independent validation samples generated from the same data generating process (Table 6.1). We include both ABC predictions described above and denote them as ABC1 and ABC2, respectively, for the two different thresholds ($c \in \{0.5, 0.25\}$) and for the two choices of the number of trees ($T \in \{10, 20\}$).

The best method under each simulation setting is marked in bold. When the data becomes more non-linear (CART and LLS setups) and the correlation among variables gets stronger, ABC tends to outperform the other methods. DART, on the other hand, works better for more linear datasets. Note that our default ABC implementation internally uses only a *small* number of $B = 200$ burn-in iterations and a small number of trees. For prediction, it has been recommended that BART is deployed with a larger number of trees



(a) ABC (b) DART (c) RF

Figure 6.4: A barplot of ordered importance measures (inclusion probabilities for ABC, importance measures for DART and RF) for each of the $p = 201$ mutations for the drug APV, where blue represents mutations found in Rhee et al. (2005). (a) Inclusion probabilities are computed using the top 1 000 out of $M = 10\,000$ ABC samples; (b) Average split of DART with 20 000 MCMC iterations; (c) log variable importance of Random Forest with 500 trees.

(Chipman et al., 2010). In addition, the ABC computation produces forest samples $f^m_{\mathcal{S},B}$ which are from an *approximate* posterior. These two facts may affect resulting predictions which may not necessarily outperform BART (DART) across-the-board.

6.6 HIV Data

To further illustrate the usefulness of our approach, we consider a dataset described and analyzed in Rhee et al. (2006) and Barber and Candès (2015). The data consists of genotype and resistance measurements (log-decrease in susceptibility) for three drug classes, i.e. protease inhibitors (PIs), nucleoside reverse transcriptase inhibitors (NRTIs) and non-nucleoside reverse transcriptase inhibitors (NNRTIs). The data is publicly available from the Stanford HIV Drug Resistance Database.[2]

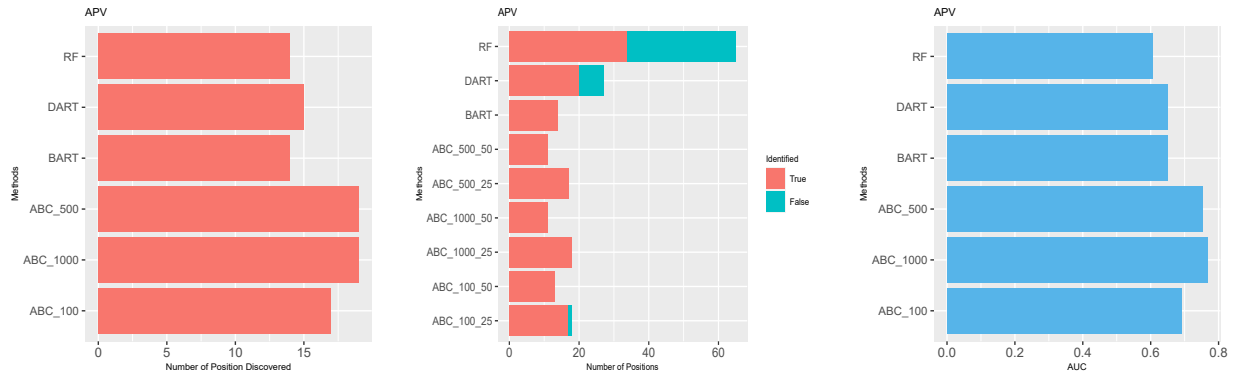
The goal of this analysis is to identify possible non-polymorphic mutation positions which result in a log-fold increase of lab-tested drug resistance. The design matrix $X = (x_{ij})_{i,j=1}^{n,p}$

2. `https://hivdb.stanford.edu/pages/published_analysis/genophenoPNAS2006/`

consists of binary indicators $x_{ij} \in \{0, 1\}$ for whether or not the j^{th} mutation occurred in the i^{th} sample. As in Barber and Candès (2015), only mutations that appear at least 3 times are taken into consideration. One appealing feature of this dataset is the availability of a proxy to the 'ground truth'. Indeed, in an independent experimental study, Rhee et al. (2005) identified mutations that are present at a significantly higher frequency in patients who have been treated with each drug. Similarly as Barber and Candès (2015), we treat this experimental data as an approximation to the truth for comparisons and for validation of our findings.

We run ABC with $M = 10\,000$ iterations, where each internal BART sample is obtained after 200 burnin iterations with 20 trees. The top 1 000 ABC samples with the smallest ϵ_m are kept and used to compute inclusion probabilities for each mutation. For illustration, we visualize results for one of the PI drugs (APV) and report the results for all the drugs in Liu et al. (2021, Section S.5). The inclusion probabilities have been ordered and plotted in Figure 6.4, where the mutations experimentally validated by Rhee et al. (2005) (a proxy for true signals) are denoted in blue and the rest is in red. For comparisons, we also included the importance measure (the average number of splits on each variable) from DART run with 20 000 MCMC iterations and $T = 200$ trees as well as the importance measure (on a log scale) from Random Forests (RF) run with 500 trees.

Figure 6.4 reveals that ABC Bayesian Forests have a strong separation power, where experimentally validated mutations generally have a higher inclusion probability. Compared to DART and RF, ABC clearly stands out as being more effective in weeding out 'noise'. We gauge the strength of the signal/noise separation using several descriptive statistics. In these comparisons, we also consider plain BART method (using $T = 20$ trees and 20 000 MCMC iterations) and ABC using the top 100 and 500 samples with the smallest tolerance level ϵ_m. Since the selection of the cut-off point is not obvious for BART and RF, we first select variables based on an adaptive cut-off point so that there are no false discoveries (i.e.



(a) Adaptive Cut-off (b) Automated Cut-off (c) AUC

Figure 6.5: (a) The number of true discoveries using an adaptive cut-off; (b) The number of true (red) and false (blue) discoveries using an automated cut-off; (c) The AUC of each method.

the cut-off is the largest importance weight of a *not* experimentally validated mutation). From the plot of the number of 'True' locations selected (displayed in Figure 6.5(a)) we can see that all three ABC implementations find more signal variables. Next, we choose the cut-off point in an automated way, where ABC importance probabilities are truncated at 0.5 and 0.25, BART and DART measures are truncated at one (i.e. the variable has been used on average at least once), and RF select variables using recursive feature elimination as explain in the previous section. Similarly to Barber and Candès (2015), we report the number of 'True' locations and 'False' locations (Figure 6.5(b)). RF selection is plagued with false discoveries and DART is not free from false identifications either. The ABC selection cutoff 0.5 results in a more conservative selection, where lowering the cutoff point to 0.25 yields more discoveries. Finally, from the plot of the AUC values for all considered methods (Figure 6.5(c)), we conclude that ABC is better at separating the experimentally validated mutations from the rest even using a very few filtered ABC samples.

6.7 Discussion

This paper makes advancements at two fronts. One is the proposal of ABC Bayesian Forests for variable selection based on a new idea of data splitting, where a fraction of data is first used for ABC proposal and the rest for ABC rejection. This new strategy increases ABC acceptance rate. We have shown that ABC Bayesian Forests are highly competitive with (and often better than) other tree-based variable selection procedures. The second development is theoretical and concerns consistency for variable and regularity selection. Continuing the theoretical investigation of BART by Ročková and van der Pas (2020), we proposed new complexity priors which jointly penalize model dimensionality and tree size. We have shown joint consistency for variable *and* regularity selection when the level of smoothness is unknown and no greater than 1. Our results are the first model selection consistency results for BART priors.

Our ABC sampling routine has the potential to be extended in various ways. Sampling from $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}_m)$ in ABC Bayesian Forests is one way of distilling $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ to propose a candidate ensemble $f^m_{\mathcal{E},\boldsymbol{B}}$. We noticed that the ABC acceptance rate can be further improved by replacing a randomly sampled tree with a fitted tree. Indeed, instead of drawing from $\pi(f_{\mathcal{E},\boldsymbol{B}}, \sigma^2 \,|\, \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S})$, one can *fit* a tree $\widehat{f}^m_{\mathcal{T},\boldsymbol{\beta}}$ to $\boldsymbol{Y}^{obs}_{\mathcal{I}_m}$ using recursive partitioning algorithms (such as the `rpart` R package of Therneau and Atkinson (2018) or with BART (by taking the posterior mean estimate $\widehat{f}^m_{\mathcal{E},\boldsymbol{B}} = \mathbb{E}[f_{\mathcal{E},\boldsymbol{B}} \,|\, \boldsymbol{Y}^{obs}_{\mathcal{I}_m}, \mathcal{S}]$). This variant, further referred to as ABC Forest Fit, is indirectly linked to other model-selection methods based on resampling.

Felsenstein (1985) proposed a "first-order bootstrap" to assess confidence of an estimated tree phylogeny. The idea was to construct a tree from each bootstrap sample and record the proportion of bootstrap trees that have a feature of interest (for us, this would be variables used for splits). Efron and Tibshirani (1998) embedded this approach within a parametric bootstrap framework, linking the bootstrap confidence level to both frequentist p-values and Bayesian a posteriori model probabilities. The authors proposed a second-order extension

by reweighting the first-order resamples according to a simple importance sampling scheme. This second-order variant performs frequentist calibration of the a-posteriori probabilities and amounts to performing Bayesian analysis with Welch-Peers uninformative priors. Efron (2012) further develops the connection between parametric Bootstrap and posterior sampling through reweighting in exponential family models. Using non-parametric bootstrap ideas, Newton and Raftery (1994) introduce the weighted likelihood bootstrap (WLB) to sample from approximate posterior distributions. The WLB samples are obtained by maximum reweighted likelihood estimation with random weights. Such posterior sampling can be beneficial when, for instance, maximization is easier than Gibbs sampling from conditionals. In a similar spirit, our ABC Forest Fit variant would perform optimization (instead of sampling) on a random subset of the dataset to obtain a candidate tree/ensemble.

It is worth pointing out that $\widehat{f}^m_{\mathcal{E},\boldsymbol{B}}$ does not necessarily have to be a tree/forest. We suggest trees because they are are easily trainable and produce stable results using traditional software packages. In principle, however, this method could be deployed in tandem with other non-parametric methods, such as deep learning, to perform variable selection.

6.8 Appendix

6.8.1 Theory

6.8.1.1 Proof of Theorem 43

We first review some notation used throughout this section and adapted from Ročková and van der Pas (2020). Recall that $\Pi_{\mathcal{S}}(\cdot)$ denotes the conditional distribution given the model $\mathcal{S}$. Next, $\mathcal{F}_{\mathcal{S}}(K)$ denotes a set of all step functions $f_{\mathcal{T},\boldsymbol{\beta}}(\cdot)$ with K steps that split on covariates $\mathcal{S}$ and $\|f_{\mathcal{T},\boldsymbol{\beta}}\|_\infty \leq B$. A tree partition is called valid when each tree splits on observed values $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and has nonempty cells. We denote with $\mathcal{V}^K_{\mathcal{S}}$ all valid trees obtained by splitting $K-1$ times along coordinates inside $\mathcal{S}$. The number of such valid trees is denoted

with $\Delta(\mathcal{V}_{\mathcal{S}}^K)$. For a valid tree partition $\mathcal{T} \in \mathcal{V}_{\mathcal{S}}^K$, we denote with $\mathcal{F}(\mathcal{T}) \subset \mathcal{F}_{\mathcal{S}}(K)$ all step functions supported on $\mathcal{T}$. We prove Theorem 43 by verifying conditions B1-B4 in Theorem 4 of Yang and Pati (2017) (further referred to as YP17). We build on tools developed in Ročková and van der Pas (2020) (further referred to as RP17).

6.8.1.1.1 Prior Concentration Condition The first condition pertains to prior concentration and consists of two parts: (a) the model prior mass condition and (b) the prior concentration condition in the parameter space under the true model. Namely, we want to show that

$$\pi(\mathcal{S}_0) \geq \mathrm{e}^{-n\,\varepsilon_{n,\mathcal{S}_0}^2} \tag{6.26}$$

and

$$\Pi_{\mathcal{S}_0}\left(f_{\mathcal{T},\boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}_0}(K) : \|f_{\mathcal{T},\boldsymbol{\beta}} - f_0\|_n \leq \varepsilon_{n,\mathcal{S}_0}\right) \geq \mathrm{e}^{-d\,n\,\varepsilon_{n,\mathcal{S}_0}^2} \tag{6.27}$$

for some $d > 2$. The prior concentration (6.26) follows directly from the definition of model weights (6.13) for $C \leq C_\varepsilon^2$ under our assumption $q_0 \log p < n^{q_0/(2\alpha+q_0)}$.

Regarding (6.27), a variant of this condition is verified in Section 8.2 of RP17 assuming that K is random with a prior. It follows from their proof, however, that (6.27) holds if we fix K at $K_{\mathcal{S}_0} = \lfloor C_K/C_\varepsilon^2\, n\,\varepsilon_{n,\mathcal{S}_0}^2/\log n \rfloor = 2^{q_0 s}$ for some $s \in \mathbb{N}$. The proof consists of (a) constructing a single approximating tree (i.e. the k-d tree with $s = (\log_2 K_{\mathcal{S}_0})/q_0$ cycles of splits on each coordinate in $\mathcal{S}_0$) and showing that it has enough prior support. This tree exists under the assumption that the design is $\mathcal{S}_0$-regular. From (8.5) of RP17, such tree approximates f_0 with an error bounded by a constant multiple of $\varepsilon_{n,\mathcal{S}_0}$. The verification of (6.27) then follows directly from RP17.

6.8.1.1.2 Entropy Condition The second condition (B4 in the notation of YP17) entails controlling the complexity of over/underfitting models. In the sequel, we focus only on models with up to q_n covariates, where $q_n = C_q\lceil n\,\varepsilon_{n,\mathcal{S}_0}^2/\log p\rceil$. This restriction is justified

by the following lemma.

Lemma 49. *Denote with* $q_n = C_q \lceil n\, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$. *Under the assumptions of Theorem 43, we have*

$$\Pi(q \geq q_n \,|\, \boldsymbol{Y}^{(n)}) \to 0 \tag{6.28}$$

in $\mathbb{P}^{(n)}_{f_0}$*-probability as* $n \to \infty$.

Proof. First, we show that $\Pi(q \geq q_n) \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \to 0$, where $d > 2$ is as in (6.27). We can write

$$\begin{aligned}
\Pi(q > q_n) \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} &\lesssim \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \sum_{k=q_n}^{p} \binom{p}{k} \mathrm{e}^{-C \times \max\{n^{k/(2\alpha+k)} \log n, k \log p\}} \\
&\leq \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0} - (C-2)\, q_n \log p} = \mathrm{e}^{-n\, \varepsilon^2_{n,\mathcal{S}_0} [(C-2) C_q - (d+2)]}.
\end{aligned}$$

The right hand side above goes to zero when $(C-2)C_q - (d+2) > 0$. This can be satisfied with $C > 2$ and C_q large enough. This fact, together with prior mass conditions (6.27) and (6.26), yields (6.28) according to Lemma 1 of Ghosal and Van Der Vaart (2007). □

Lemma 49 essentially states that the posterior will not reward models whose dimensionality is larger than (or equal to) q_n. In our following considerations, we thus condition only models with less than q_n variables.

We now verify that the complexity of overfitting models $\mathcal{S} \supset \mathcal{S}_0$ is not too large in the sense that their global metric entropy satisfies

$$\log N\left(\varepsilon_{n,\mathcal{S}}\,;\, \mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}})\,;\, \|\cdot\|_n\right) \leq n\, \varepsilon^2_{n,\mathcal{S}}. \tag{6.29}$$

First, we note that for two tree step functions $f_{\mathcal{T},\boldsymbol{\beta}_1} \in \mathcal{F}(\mathcal{T})$ and $f_{\mathcal{T},\boldsymbol{\beta}_2} \in \mathcal{F}(\mathcal{T})$ that have the same partition $\mathcal{T} \in \mathcal{V}_{\mathcal{S}}^{K_{\mathcal{S}}}$ and different step heights $\boldsymbol{\beta}_1 \in \mathbb{R}^{K_{\mathcal{S}}}$ and $\boldsymbol{\beta}_2 \in \mathbb{R}^{K_{\mathcal{S}}}$, we have $\{\|f_{\mathcal{T},\boldsymbol{\beta}_1} - f_{\mathcal{T},\boldsymbol{\beta}_2}\|_n \leq \varepsilon_{n,\mathcal{S}}\} \supset \{\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_2 \leq \varepsilon_{n,\mathcal{S}}\}$. Furthermore, noting that

$\mathcal{F}(\mathcal{T}) = \{f_{\mathcal{T},\boldsymbol{\beta}} : \|f_{\mathcal{T},\boldsymbol{\beta}}\|_\infty \leq B\} \subset \{\boldsymbol{\beta} \in \mathbb{R}^{K_\mathcal{S}} : \|\boldsymbol{\beta}\|_2 \leq B\sqrt{n}\}$ we can write

$$N\left(\varepsilon_{n,\mathcal{S}}\,;\,\mathcal{F}(\mathcal{T})\,;\,\|\cdot\|_n\right) \leq \left(\frac{3\,B\,\sqrt{n}}{\varepsilon_{n,\mathcal{S}}}\right)^{K_\mathcal{S}} \leq \left(3\,B\,n^{3/2}/C_\varepsilon\right)^{K_\mathcal{S}},$$

where we used the standard $\varepsilon_{n,\mathcal{S}}$ covering number of a $K_\mathcal{S}$-Euclidean ball of a radius $B\sqrt{n}$ and the fact that $1/\varepsilon_{n,\mathcal{S}} \leq 1/C_\varepsilon \times n^{\alpha/(2\alpha+|\mathcal{S}|)} \leq 1/C_\varepsilon \times n$. Then we can write

$$N\left(\varepsilon_{n,\mathcal{S}}\,;\,\mathcal{F}_\mathcal{S}(K_\mathcal{S})\,;\,\|\cdot\|_n\right) \leq \Delta(\mathcal{V}_\mathcal{S}^{K_\mathcal{S}})\left(3\,B\,n^{3/2}/C_\varepsilon\right)^{K_\mathcal{S}}.$$

Using Lemma 3.1 of Ročková and van der Pas (2020), we have $\Delta(\mathcal{V}_\mathcal{S}^{K_\mathcal{S}}) \leq (K_\mathcal{S}\,n\,|\mathcal{S}|)^{K_\mathcal{S}}$.

The overall log-covering number is then upper-bounded with (since $|\mathcal{S}| \leq q_n \leq n$)

$$K_\mathcal{S} \log\left(3\,B\,n^3\,n^{3/2}\right) \lesssim K_\mathcal{S} \log n \propto n\,\varepsilon_{n,\mathcal{S}}^2. \tag{6.30}$$

This verifies the model complexity condition for overfitting models. Next, we need to verify (6.29) with $\varepsilon_{n,\mathcal{S}}$ replaced by $\widetilde{\varepsilon}_n$ for "underfitting" models $\mathcal{S} \in \boldsymbol{\Gamma}_{\mathcal{S}\not\supset\mathcal{S}_0}$ where $|\mathcal{S}| \leq q_n$. This follows from the same arguments as above and the fact that $\varepsilon_{n,\mathcal{S}} \leq \widetilde{\varepsilon}_n$. Finally, the last requirement in Assumption B4 of YP17 is verifying that

$$\sum_{\mathcal{S}\not\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \mathrm{e}^{-C_2\,n\,\widetilde{\varepsilon}_n^2} + \sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \mathrm{e}^{-C_2\,n\,\varepsilon_{n,\mathcal{S}}^2} \leq 1 \tag{6.31}$$

for some large constant $C_2 > 0$. Since $\widetilde{\varepsilon}_n \geq \varepsilon_{n,\mathcal{S}} > \varepsilon_{n,\mathcal{S}_0}$ for any $\mathcal{S} \supset \mathcal{S}_0$ such that $|\mathcal{S}| \leq q_n$, we can upper-bound the left-hand side above with

$$\sum_{q=0}^{q_n} \sum_{\mathcal{S}:|\mathcal{S}|=q} \mathrm{e}^{-C_2\,n\,\varepsilon_{n,\mathcal{S}_0}^2} \leq \mathrm{e}^{-C_2\,n\,\varepsilon_{n,\mathcal{S}_0}^2} \sum_{q=0}^{q_n} \binom{p}{q} \leq \left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1} \mathrm{e}^{-C_2\,n\varepsilon_{n,\mathcal{S}_0}^2}$$

From our definition of q_n, we have $q_n \log p \asymp n\,\varepsilon_{n,\mathcal{S}_0}^2$ and (6.31) will be satisfied for a large

enough C_2.

6.8.1.1.3 Prior Anticoncentration Condition Lastly, as one of the sufficient conditions for model selection consistency, we need to verify

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \pi(\mathcal{S})\,\Pi_{\mathcal{S}}\left(f_{\mathcal{T},\boldsymbol{\beta}}\in\mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}}):\|f_0-f_{\mathcal{T},\boldsymbol{\beta}}\|_n\leq M\,\varepsilon_{n,\mathcal{S}}\right)\leq \mathrm{e}^{-H\,n\,\varepsilon^2_{n,\mathcal{S}_0}} \tag{6.32}$$

for some $H>0$. Alternatively, YP17 introduce the so-called "anti-concentration condition" $\Pi_{\mathcal{S}}\left(f_{\mathcal{T},\boldsymbol{\beta}}\in\mathcal{F}_{\mathcal{S}}(K_{\mathcal{S}}):\|f_0-f_{\mathcal{T},\boldsymbol{\beta}}\|_n\leq M\,\varepsilon_{n,\mathcal{S}}\right)\leq \mathrm{e}^{-H\,n\,\varepsilon^2_{n,\mathcal{S}_0}}$ for overfitting models $\mathcal{S}\supset\mathcal{S}_0$ where $\varepsilon_{n,\mathcal{S}}\geq\varepsilon_{n,\mathcal{S}_0}$. This condition is needed to show that the posterior probability of more complex models that contain the truth goes to zero.

It turns out that this condition can be avoided with our choice of model weights $\pi(\mathcal{S})$ (Ghosal et al., 2008). We can verify (6.32) directly (without the anticoncentration condition) by upper-bounding the left hand side of (6.32) with

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n}\pi(\mathcal{S})\leq\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n}\mathrm{e}^{-C\,n\,\varepsilon^2_{n,\mathcal{S}}}\leq\mathrm{e}^{-C\,n\,\varepsilon^2_{n,\mathcal{S}_0}}\left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1}. \tag{6.33}$$

Since $q_n\log p\asymp n\,\varepsilon^2_{n,\mathcal{S}_0}$, (6.32) holds for $H<C-1$.

6.8.1.1.4 Identifiability Under the identifiability and irrepresentability assumptions (41) and (42), it turns out that we cannot approximate f_0 well enough with models that miss at least one covariate. This property is summarized in the following Lemma, which is a variant of Proposition 1 of YP17.

Lemma 50. *For $f_0\in\mathcal{H}^{\alpha}_p\cap\mathcal{C}(\mathcal{S}_0)$, assume that $\mathcal{S}_0$ is (f_0,ε)-identifiable and that ε-irrepresentability holds. Then*

$$\inf_{\mathcal{S}\not\supset\mathcal{S}_0}\inf_{f_{\mathcal{T},\boldsymbol{\beta}}\in\mathcal{F}_{\mathcal{S}}}\|f_0-f_{\mathcal{T},\boldsymbol{\beta}}\|_n>M\,\varepsilon.$$

Proof. We decompose $\mathcal{S} \not\supset \mathcal{S}_0$ into true positives and false positives, i.e. $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$, where $\mathcal{S}_1 \subset \mathcal{S}_0$ and $\mathcal{S}_2 \cap \mathcal{S}_0 = \emptyset$. We denote with $\widehat{f}^{\mathcal{S}}$ the projection of f_0 onto $\mathcal{F}_{\mathcal{S}}$, omitting the subscripts $\widehat{\mathcal{T}}$ and $\widehat{\boldsymbol{\beta}}$. With a slight abuse of notation we denote $\mathbb{E}(f, g) = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)g(\boldsymbol{x}_i)$. Then we can write

$$\|f_0 - \widehat{f}^{\mathcal{S}}\|_n^2 = \|f_0 - \widehat{f}^{\mathcal{S}_1} + \widehat{f}^{\mathcal{S}_1} - \widehat{f}^{\mathcal{S}}\|_n^2 > \|f_0 - \widehat{f}^{\mathcal{S}_1}\|_n^2 - 2|\mathbb{E}[(f_0 - \widehat{f}^{\mathcal{S}_1})(\widehat{f}^{\mathcal{S}} - \widehat{f}^{\mathcal{S}_1})]|,$$

where $\mathbb{E}[(f_0 - \widehat{f}^{\mathcal{S}_1})(\widehat{f}^{\mathcal{S}} - \widehat{f}^{\mathcal{S}_1})]$ equals $\rho_n^{\mathcal{S}}$ defined in (6.18). We note that $\delta_n^{\mathcal{S}_1}$ is monotone increasing in the number of false non-discoveries $|\mathcal{S}_0 \backslash \mathcal{S}_1|$. The statement of the Lemma then follows from the fact that $\|f_0 - \widehat{f}^{\mathcal{S}}\|_n^2 > \inf_{\mathcal{S}_1 \subset \mathcal{S}_0} \delta_n^{S_1} - 2 \sup_{\mathcal{S} \not\supset \mathcal{S}_0} \rho_n^{S} > \inf_{i \in \mathcal{S}_0} \delta_n^{S_0 \backslash i} - M\,\varepsilon > M\varepsilon$. □

6.8.1.2 Proof of Theorem 47

We introduce some more notation. We denote with $\mathcal{F}_{\mathcal{S}} = \bigcup_{K=1}^{n} \mathcal{F}_{\mathcal{S}}(K)$ all valid trees that split on directions inside $\mathcal{S}$ and we write $\Pi_{K,\mathcal{S}}(\cdot)$ for the conditional prior, given K and $\mathcal{S}$.

Similarly as in Section 6.8.1.1, we verify the three conditions (Prior Concentration, Entropy, Prior Anti-concentration). The prior model concentration condition is again satisfied automatically from the definition of model weights in (6.20) and $K_{\mathcal{S}_0} = \lfloor C_K / C_\varepsilon \, n \, \varepsilon_{n,\mathcal{S}_0}^2 / \log n \rfloor$. Namely,

$$\pi(K_{\mathcal{S}_0}, \mathcal{S}_0) \propto \mathrm{e}^{-C \, \max\{C_K / C_\varepsilon n \, \varepsilon_{n,\mathcal{S}_0}^2, q_0 \log p\}} \geq \mathrm{e}^{-n \, \varepsilon_{n,\mathcal{S}_0}^2}, \tag{6.34}$$

for $C_K < C_\varepsilon / C$, where we used the assumption $q_0 \log p \leq n^{q_0/(2\alpha + q_0)}$. Next, the prior concentration in the parameter space associated with the true model

$$\Pi_{K_{\mathcal{S}_0}, \mathcal{S}_0} \left(f_{\mathcal{T}, \boldsymbol{\beta}} \in \mathcal{F}_{\mathcal{S}_0}(K_{\mathcal{S}_0}) : \|f_{\mathcal{T}, \boldsymbol{\beta}} - f_0\|_n \leq \varepsilon_{n, \mathcal{S}_0} \right) \geq \mathrm{e}^{-d \, n \, \varepsilon_{n, \mathcal{S}_0}^2}$$

follows again from Section 8.2 of RP17.

For the entropy considerations, we focus only on models with up to q_n covariates and

up to K_n splits, where $q_n = \lceil C_q n\, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$ and $K_n = \lceil \bar{C} n\, \varepsilon^2_{n,\mathcal{S}_0} / \log n \rceil$ were defined in Theorem 47. This restriction is justified by the following Lemma.

Lemma 51. *Denote with* $q_n = \lceil C_q n\, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$ *and* $K_n = \lceil \bar{C}\, n\, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$. *Under the assumptions of Theorem 43, we have*

$$\Pi(q \geq q_n \,|\, \boldsymbol{Y}^{(n)}) \to 0 \quad \textit{and} \quad \Pi(K \geq K_n \,|\, \boldsymbol{Y}^{(n)}) \to 0 \tag{6.35}$$

in $\mathbb{P}^{(n)}_{f_0}$*-probability as* $n \to \infty$.

Proof. It suffices to show that $\Pi(q > q_n) \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \to 0$ and $\Pi(K \geq K_n) \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \to 0$ for $d > 2$ from (6.27). We have $q_0 \leq q_n$ for n large enough, since $q_0 = \mathcal{O}(1)$ as $n \to \infty$, and thereby

$$\begin{aligned}
&\Pi(q \geq q_n) \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \lesssim \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=q_n}^{p} \binom{p}{q} \sum_{K=1}^{n} \mathrm{e}^{-C \max\{K \log n, q \log p\}} \\
&\leq \mathrm{e}^{(d+2) n \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=q_n}^{p} \mathrm{e}^{\log n + q \log(p\,\mathrm{e}/q) - C\, q \log p} \leq \mathrm{e}^{\log p + \log n - (C-1)\, q_n \log p + (d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \\
&\leq \mathrm{e}^{-(C-3)\, q_n \log p + (d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}},
\end{aligned}$$

where we used the fact that for $q_0 \geq 2$ and $\alpha \in (0, 1]$, we have $\log n \leq n^{q_0/(2\alpha+q_0)}$. Since $q_n \log p \geq C_q n \varepsilon^2_{n,\mathcal{S}_0}$, the right hand side above goes to zero when $(C-3) C_q > d + 2$. This will be guaranteed with $C > 3$ and C_q large enough. Similarly, we have

$$\begin{aligned}
&\Pi(K \geq K_n) \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \lesssim \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=0}^{p} \binom{p}{q} \sum_{K=K_n}^{n} \mathrm{e}^{-C \max\{K \log n, q \log p\}} \\
&\leq \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=0}^{p} \sum_{K=K_n}^{n} \mathrm{e}^{-(C-1) \max\{K \log n, q \log p\}} \\
&\leq \mathrm{e}^{\log(p+1) + \log n - (C-1)\, K_n \log n + (d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \leq \mathrm{e}^{-(C-2)\, K_n \log n + (d+3) n\, \varepsilon^2_{n,\mathcal{S}_0}},
\end{aligned}$$

where we used our assumption $\log p \leq n^{q_0/(2\alpha+q_0)}$. Since $K_n \geq \bar{C} n\, \varepsilon^2_{n,\mathcal{S}_0}$, the right hand side above goes to zero when $(C-2)\bar{C} > d+3$. Together with the prior mass conditions (6.27) and (6.34), (6.35) follows from Lemma 1 of Ghosal and Van Der Vaart (2007). □

This Lemma essentially says that the posterior does not overfit in terms of both q and K, where the mass concentrates on models with $K < K_n$ splits. Note that K_n is of the same order as the optimal regularity $K_{\mathcal{S}_0}$. Now, we denote with $\mathbf{\Gamma}_n \subset \mathbf{\Gamma}$ a sieve consisting of all models with less than q_n variables and K_n splits. For the entropy bounds of overfitting and underfitting models (inside the sieve $\mathbf{\Gamma}_n$), we can use the same arguments as in Section 6.8.1.1. Assume a model $(K, \mathcal{S}) \in \mathbf{\Gamma}_n$. Then it follows from (6.30) that

$$\log N\left(\varepsilon_{n,\mathcal{S}}\,;\, \mathcal{F}_{\mathcal{S}}(K)\,;\, \|\cdot\|_n\right) \leq K \log(3\,B\,n^3 n^{3/2}) \lesssim K_n \log n \lesssim n\,\varepsilon^2_{n,\mathcal{S}_0}.$$

For over-fitting models, this can be further upper-bounded with a multiple of $n\,\varepsilon^2_{n,\mathcal{S}}$, thus satisfying (6.29). The last requirement for the entropy condition is verifying the following variant of (6.31)

$$\sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\, \mathcal{S}\not\supset\mathcal{S}_0\cup K<K_{\mathcal{S}_0}} \mathrm{e}^{-C_2\,M^2\,n\,\varepsilon^2_{n,\mathcal{S}_0}} + \sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\, \mathcal{S}\supset\mathcal{S}_0\cap K\geq K_{\mathcal{S}_0}} \mathrm{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}}} \leq 1 \qquad (6.36)$$

for some suitable $C_2 > 0$. Since $n\,\varepsilon^2_{n,\mathcal{S}_0} \leq n\,\varepsilon^2_{n,\mathcal{S}}$ for $\mathcal{S} \supset \mathcal{S}_0$, we can upper-bound the left hand side with

$$\sum_{\mathcal{S}:|\mathcal{S}|<q_n} \sum_{K=1}^{K_n} \mathrm{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}} \leq \mathrm{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}} \left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1} \mathrm{e}^{\log K_n} \leq \mathrm{e}^{-C_2 n\,\varepsilon^2_{n,\mathcal{S}_0}+(q_n+1)\log p+\log K_n}. \qquad (6.37)$$

Since $q_n \log p \asymp n\,\varepsilon^2_{n,\mathcal{S}_0}$ and $\log K_n \lesssim n^{q_0/(2\alpha+q_0)} \lesssim n\varepsilon^2_{n,\mathcal{S}_0}$, the right-hand side of (6.37) converges to zero for some suitably large C_2 as $n \to \infty$, thus satisfying (6.36).

In place of the anti-concentration condition (similarly as in (6.33)), we need to verify that

the prior probability of larger models (that contain the truth) is small in the sense that, for some $H > 0$,

$$\sum_{(K,\mathcal{S})\in\mathbf{\Gamma}_n:\{\mathcal{S}\supset\mathcal{S}_0\}\cap\{K\geq K_{\mathcal{S}_0}\}} \pi(\mathcal{S},K) \leq \mathrm{e}^{-H\,n\,\varepsilon^2_{n,\mathcal{S}_0}}. \tag{6.38}$$

We can write

$$\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|<q_n} \sum_{K=K_{\mathcal{S}_0}}^{K_n} \pi(\mathcal{S},K) \leq \sum_{q=0}^{q_n} \binom{p}{q} \sum_{K=K_{\mathcal{S}_0}}^{K_n} \mathrm{e}^{-C\,K_{\mathcal{S}_0}\log n} \tag{6.39}$$

$$\leq \left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1} \mathrm{e}^{\log K_n} \mathrm{e}^{-C\,K_{\mathcal{S}_0}\log n}. \tag{6.40}$$

Because $q_n \log p \asymp n\,\varepsilon^2_{n,\mathcal{S}_0}$ and $\log K_n \lesssim n^{q_0/(2\alpha+q_0)} \lesssim n\varepsilon^2_{n,\mathcal{S}_0}$ the condition (6.38) is satisfied for some $H > 0$ when C and C_K are large enough.

6.8.1.3 Proof of Theorem 48

We modify the notation a bit. We adopt the definition of δ-valid ensembles from RP17 (Definition 5.3). With $\mathcal{F}_{\mathcal{S}}(\boldsymbol{K})$ we denote all δ-valid tree ensembles $f_{\mathcal{E},\boldsymbol{B}}$ that (a) are uniformly bounded (i.e. $\|f_{\mathcal{E},\boldsymbol{B}}\|_\infty \leq B$ for some $B > 0$), (b) consist of T trees with $\boldsymbol{K} = (K^1,\ldots,K^T)' \in \mathbb{N}^T$ leaves and (c) that split along directions $\mathcal{S}$.

We start by showing that the prior model concentration condition is satisfied. From our assumption $q_0 \log p \leq n^{q_0/(2\alpha+q_0)}$ and definition $K_{\mathcal{S}_0} < C_K/C_\varepsilon^2\, n\,\varepsilon^2_{n,\mathcal{S}_0}/\log n$ and using (6.23) and (6.24), we obtain

$$\pi(\mathcal{S}_0, E(K_{\mathcal{S}_0})) \propto \sum_{T=1}^{K_{\mathcal{S}_0}} \mathrm{e}^{-C_T\,T} \sum_{\boldsymbol{K}\in\mathbb{N}^T:\sum_{t=1}^T K^t = K_{\mathcal{S}_0}} \mathrm{e}^{-C\,n^{q_0/(2\alpha+q_0)}\log n} \geq \mathrm{e}^{-(C_T C_K/(C_\varepsilon\,\log n)+C/C_\varepsilon^2)\,n\,\varepsilon^2_{n,\mathcal{S}_0}}.$$

The right-hand side can be further lower-bounded with $\mathrm{e}^{-n\,\varepsilon^2_{n,\mathcal{S}_0}}$ for a large enough C_ε and n. Next, we need to show prior concentration in the parameter space under the true model

equivalence class $(\mathcal{S}_0, E(K_{\mathcal{S}_0}))$. All that is needed is finding a single well-approximating forest supported on one partition ensemble characterized by $(T, \boldsymbol{K})$ from the equivalence class $E(K_{\mathcal{S}_0})$. Such an ensemble can be obtained by considering $T = 1$ and a single k-d tree with $K_{\mathcal{S}_0}$ leaves from Lemma 3.2 of RP17. The prior concentration condition then boils down to (6.27), which has already been verified in RP17.

Next, we show that for $K_n = \lceil \bar{C}\, n\, \varepsilon^2_{n,\mathcal{S}_0} / \log n \rceil$ we have

$$\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^{T} K^t \geq K_n \,\middle|\, \boldsymbol{Y}^{(n)} \right) \to 0.$$

We can write

$$\begin{aligned}\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^{T} K^t \geq K_n \right) &\lesssim \sum_{T=1}^{n} \mathrm{e}^{-C_T\, T} \sum_{q=0}^{p} \binom{p}{q} \sum_{Z=K_n}^{n} \sum_{\boldsymbol{K}:\sum_{t=1}^{T} K^t = Z} \mathrm{e}^{-C \max\{Z \log n, q \log p\}} \\ &\lesssim \mathrm{e}^{-(C-1)K_n \log n + \log p + 2\log n + \log p(n) - C_T},\end{aligned}$$

where $p(n)$ is the partitioning number. According to Andrews (1976), we have

$$\log p(n) \sim \pi \sqrt{\frac{2\, n}{3}} \quad \text{as} \quad n \to \infty. \tag{6.41}$$

Under our assumptions $q_0 > 2$ and $\alpha \in (0, 1]$, we have $\sqrt{n} \leq n^{q_0/(2\alpha + q_0)}$ and $\log n \leq n^{q_0/(2\alpha+q_0)}$. From $\log p \leq n^{q_0/(2\alpha+q_0)}$ and using the fact that $K_n \geq \bar{C}\, n\, \varepsilon^2_{n,\mathcal{S}_0} / \log n$, we can then write

$$\Pi\left((T, \boldsymbol{K}) : \sum_{t=1}^{T} K^t \geq K_n \right) \mathrm{e}^{(d+2) n\, \varepsilon^2_{n,\mathcal{S}_0}} \lesssim \mathrm{e}^{-[(C-1)\bar{C} - D\,\pi\sqrt{2/3} - d - 5] n\, \varepsilon^2_{n,\mathcal{S}_0}}$$

for some $D > 0$. The right hand side goes to zero for $C > 1$ and $\bar{C}$ large enough. Similarly, we can show that $\Pi(q \geq q_n \,|\, \boldsymbol{Y}^{(n)}) \to 0$ as $n \to \infty$ for $q_n = \lceil C_q\, n\, \varepsilon^2_{n,\mathcal{S}_0} / \log p \rceil$ by proceeding

as in Lemma 51 in Section 6.8.1.2.

Based on the previous paragraph, we narrow down attention to a subset of model indices $\mathbf{\Gamma}_n \subset \mathbf{\Gamma}$, consisting of models $(\mathcal{S}, E(Z))$ such that $|\mathcal{S}| < q_n$ and $Z < K_n$. We now define a sieve $\mathcal{F}_n$ as follows

$$\mathcal{F}_n = \bigcup_{q=0}^{q_n} \bigcup_{T=1}^{K_n} \bigcup_{\sum_{t=1}^{T} K^t \leq K_n} \bigcup_{\mathcal{S}:|\mathcal{S}|=q} \mathcal{F}_{\mathcal{S}}(\boldsymbol{K}).$$

It follows from the previous paragraph that $\Pi(\mathcal{F}_n^c \mid \boldsymbol{Y}^{(n)}) \to 0$ as $n \to \infty$. For the entropy calculation we thus focus on the sieve $\mathcal{F}_n$.

We first note that the metric entropy $\log N\left(\varepsilon_{n,\mathcal{S}}; \mathcal{F}(\mathcal{E}); \|\cdot\|_n\right)$, where $\mathcal{F}(\mathcal{E})$ are all uniformly bounded forests supported on a δ-valid partition ensemble $\mathcal{E}$, can be upper-bounded with $\left(\sum_{t=1}^{T} K^t\right) \log(B/\varepsilon_{n,\mathcal{S}} C_1 \kappa(\mathcal{E})\sqrt{n})$ (follows from equation (9.3) of RP17), where $\kappa(\mathcal{E})$ is the condition number of a valid ensemble (defined in Section 9.1. of RP17). Next, we find an upper bound for the covering number of the tree ensembles that are attached to a model $(\mathcal{S}, E(Z))$, where $E(Z)$ is the equivalence class of $(T, \boldsymbol{K})$ defined in (6.22). From Section 9.1 of RP17, and using the fact that $\Delta(E(Z)) \leq Z! p(Z)$, it follows that

$$\begin{aligned}
&\log N\left(\varepsilon_{n,\mathcal{S}}; \bigcup_{(T,\boldsymbol{K})\in E(Z)} \mathcal{F}_{\mathcal{S}}(\boldsymbol{K}) \cap \mathcal{F}_n; \|\cdot\|_n\right) \\
&\quad \leq \log \Delta(E(Z)) + \log \Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}) + Z \log(B/\varepsilon_{n,\mathcal{S}} C_1 \kappa(\mathcal{E})\sqrt{n}) \\
&\quad \lesssim Z \log Z + \sqrt{Z} + Z \log(|\mathcal{S}| n^2) + Z \log\left(n^{2+\delta/2}\sqrt{Z}\right)
\end{aligned}$$

for some $C_1 > 0$, where $\Delta(\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}})$ is the cardinality of δ-valid ensembles $\mathcal{VE}_{\mathcal{S}}^{\boldsymbol{K}}$. Inside the sieve, we have $|\mathcal{S}| < q_n \leq n$ and $Z < K_n \asymp n\,\varepsilon_{n,\mathcal{S}_0}^2 / \log n$ and thereby we can upper bound the log entropy with a constant multiple of $n\,\varepsilon_{n,\mathcal{S}_0}^2$. For an overfitting model $(\mathcal{S}, E(Z))$ such that $Z \geq K(\mathcal{S}_0)$ and $\mathcal{S} \supset \mathcal{S}_0$, the log-covering number is further upper-bounded with

$n\,\varepsilon^2_{n,\mathcal{S}} \geq n\,\varepsilon^2_{n,\mathcal{S}_0}$. Next, we verify the following variant of condition (6.31)

$$\sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \not\supset \mathcal{S}_0\} \cup \{Z < K_{\mathcal{S}_0}\}}} \mathrm{e}^{-C_2\, M^2\, n\, \varepsilon^2_{n,\mathcal{S}_0}} + \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \mathrm{e}^{-C_2 n\, \varepsilon^2_{n,\mathcal{S}}} \leq 1 \tag{6.42}$$

for some $C_2 > 0$. Since $n\,\varepsilon^2_{n,\mathcal{S}} > n\,\varepsilon^2_{n,\mathcal{S}_0}$ for $\mathcal{S} \supset \mathcal{S}_0$ and $M > 1$, we can upper-bound the left-hand-side with

$$\mathrm{e}^{-C_2\, n\, \varepsilon^2_{n,\mathcal{S}_0}} \sum_{q=0}^{q_n} \binom{p}{q} \sum_{Z=1}^{K_n} \Delta(E(Z)) \lesssim \left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1} \mathrm{e}^{-C_2\, n\, \varepsilon^2_{n,\mathcal{S}_0} + \log q_n + \log K_n + K_n \log K_n + \pi\sqrt{2K_n/3}},$$

where we used the fact $\Delta(E(Z)) \leq Z! p(Z)$ and (6.41). Since $K_n \log K_n \lesssim n\,\varepsilon^2_{n,\mathcal{S}_0}$ and $q_n \log p \asymp n\varepsilon^2_{n,\mathcal{S}_0}$, the right hand side goes to zero for a large enough constant $C_2 > 0$.

Lastly, the anti-concentration condition is replaced with

$$\sum_{T=K_n}^{n} \pi(T) \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \sum_{\boldsymbol{K} \in \mathbb{N}^T : \sum_{t=1}^T K^t = Z} \pi(\mathcal{S}, \boldsymbol{K} \mid T) \leq \mathrm{e}^{-H\, n\, \varepsilon^2_{n,\mathcal{S}_0}}$$

for some $H > 0$. Using the fact $\pi(\mathcal{S}, \boldsymbol{K} \mid T) \gtrsim \mathrm{e}^{-C \sum K^t \log n}$, we can upper-bound the left hand side above with

$$\begin{aligned} &\sum_{T=1}^{K_n} \pi(T) \mathrm{e}^{-C\, K_{\mathcal{S}_0} \log n} \sum_{\boldsymbol{\Gamma}_n \cap \boldsymbol{\Gamma}_{\{\mathcal{S} \supset \mathcal{S}_0\} \cap \{Z \geq K_{\mathcal{S}_0}\}}} \Delta(E(Z)) \\ &\qquad \lesssim \mathrm{e}^{-C\, K_{\mathcal{S}_0} \log n} \left(\frac{2\,\mathrm{e}\,p}{q_n}\right)^{q_n+1} \mathrm{e}^{2\log K_n + K_n \log K_n + \pi\sqrt{2K_n/3} - C_T} \end{aligned}$$

Using similar arguments as before, and because $K_{\mathcal{S}_0} \log n \geq C_K / C_\varepsilon\, n\, \varepsilon^2_{n,\mathcal{S}_0}$, the condition will be satisfied for large enough $C > 0$ and $C_K > 0$.

6.8.1.4 Theory for ABC

First, we show the following ABC posterior concentration result.

Theorem 52. *Under the assumptions of Theorem 4.1 and assuming $\sigma^2 = 1/n$ in (1), the naive ABC posterior satisfies with $\mathbb{P}_{f_0}^{(n)}$ tending to one*

$$\Pi\left[\|f - f_0\|_n > \lambda_n \,|\, \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \lesssim 1/M$$

for $\epsilon_n^T = \sqrt{2\log n/n}$, $\lambda_n = 4\epsilon_n^T/3 + 1/\sqrt{n}$ and for any $M > 0$ large enough.

Proof. We will be working conditionally on the event $\mathcal{A} = \{\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)' : \max_{1\leq i\leq n} |\varepsilon_i| \leq \sqrt{2\log n/n}\}$ whose complement has a small probability, i.e. $\mathbb{P}_{f_0}^{(n)}[\mathcal{A}^c] \leq c_0/\sqrt{2\log n}$ for some $c_0 > 0$ when $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 1/n)$. On the event $\mathcal{A}$, we have

$$\|\boldsymbol{Y} - f_0\|_n = \sqrt{\frac{1}{n}\sum_{i=1}^n \varepsilon_i^2} \leq \sqrt{2\log n/n} \equiv \epsilon_n^T.$$

We now define a joint event

$$\mathcal{A}(\epsilon_n^T, \lambda_n) \equiv \{(\boldsymbol{Y}^\star, f) : \|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T \quad \text{and} \quad \|f - f_0\|_n > \lambda_n\}.$$

For all $(\boldsymbol{Y}^\star, f) \in \mathcal{A}(\epsilon_n^T, \lambda_n)$ we have

$$\|f - f_0\|_n \leq \|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n + \|f - \boldsymbol{Y}^\star\|_n + \|f_0 - \boldsymbol{Y}\|_n \leq \frac{4}{3}\epsilon_n^T + \|f - \boldsymbol{Y}^\star\|_n.$$

This means that $(\boldsymbol{Y}^\star, f) \in \mathcal{A}(\epsilon_n^T, \lambda_n)$ implies $\|f - \boldsymbol{Y}^\star\|_n > \lambda_n - \frac{4}{3}\epsilon_n^T$ and choosing $\lambda_n \geq \frac{4}{3}\epsilon_n^T + t_\varepsilon$ leads to

$$\Pi[\mathcal{A}(\epsilon_n^T, \lambda_n)] \leq \int \mathbb{P}_f[\|f - \boldsymbol{Y}^\star\|_n > t_\varepsilon] d\Pi(f)$$

and

$$\Pi\left[\|f-f_0\|_n > \frac{4}{3}\epsilon_n^T + t_\varepsilon \,\middle|\, \|\boldsymbol{Y}-\boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \leq \frac{\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n > t_\varepsilon] d\Pi(f)}{\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T] d\Pi(f)}. \quad (6.43)$$

Now, we have for a random variable χ_n^2 with a chi-square distribution with n degrees of freedom

$$\mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n > u] = \mathbb{P}_f\left[\frac{\chi_n^2}{n^2} > u^2\right] = \mathbb{P}_f\left[\mathrm{e}^{\chi_n^2/4} > \mathrm{e}^{u^2\, n^2/4}\right] \leq \frac{2^{n/2}}{\mathrm{e}^{u^2\, n^2/4}}.$$

Next, for n large enough we can write

$$\begin{aligned}
\int \mathbb{P}_f[\|\boldsymbol{Y}^\star - \boldsymbol{Y}\|_n \leq \epsilon_n^T] d\Pi(f) &\geq \int_{\|f-f_0\|_n \leq \epsilon_n^T/3} \mathbb{P}_f[\|\boldsymbol{Y}^\star - f\|_n \leq \epsilon_n^T/3] d\Pi(f) && (6.44)\\
&\geq \Pi[\|f-f_0\|_n \leq \epsilon_n^T/3] - \mathrm{e}^{n/2\log 2 - n\,\log n/18} && (6.45)\\
&\geq \Pi[\|f-f_0\|_n \leq \epsilon_n^T/3]/2. && (6.46)
\end{aligned}$$

Next (under the assumption $q_0 \log p < n^{q_0/(2\alpha+q_0)}$, we have $\pi(\mathcal{S}_0) \geq \mathrm{e}^{-n\varepsilon^2_{n,\mathcal{S}_0}}$ and (assuming $K = K_{\mathcal{S}_0} \asymp n\varepsilon^2_{n,\mathcal{S}_0}/\log n$ and denoting $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^K$ the steps of the $\|\cdot\|_n$ projection of f_0 onto trees with K leafs) for some $c > 0$

$$\begin{aligned}
\Pi[\|f-f_0\|_n \leq \epsilon_n^T/3] &> \mathrm{e}^{-n\,\varepsilon^2_{n,\mathcal{S}_0}} \Pi(\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|_2 \leq \epsilon_n^T/6) && (6.47)\\
&> \mathrm{e}^{-n\,\varepsilon^2_{n,\mathcal{S}_0}} \frac{\mathrm{e}^{-K\log 2 - \|\widehat{\boldsymbol{\beta}}\|_2^2 - (\varepsilon_n^T)^2/72 + K/2\log[(\varepsilon_n^T)^2/36]}}{\Gamma(K/2)K/2} > \mathrm{e}^{-c\,n\,\varepsilon^2_{n,\mathcal{S}_0}}. && (6.48)
\end{aligned}$$

We can now upper-bound (6.43) with $2^{n/2}\mathrm{e}^{-t_\varepsilon^2\, n^2/4 + cn\,\varepsilon^2_{n,\mathcal{S}_0}}$ which is smaller than an arbitrary constant $M > 0$ for n large enough if we choose $t_\varepsilon = 1/\sqrt{n}$. □

Given this consistency result, we can immediately conclude (using the inequality in (21)

in the paper) that the ABC posterior will not reward underfitting model as long as our identifiability and irrepresentability conditions are satisfied with $\varepsilon = \lambda_n$. In other words, under the assumptions of Theorem 52 and assuming that $\mathcal{S}_0$ is (f_0, λ_n)-identifiable and that λ_n-irrepresentability holds we have, with $\mathbb{P}_{f_0}$ tending to one and for any $M > 0$,

$$\Pi\left[\mathcal{S} \not\supset \mathcal{S}_0 \,|\, \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] \lesssim 1/M.$$

Regarding over-fitting models, we first show the following ABC analogue of Lemma 8.1. We can write, on the event $\mathcal{A}$, and for $q_n = C_q \lceil n\, \varepsilon_{n,\mathcal{S}_0}^2 / \log p \rceil$ (as in Lemma 8.1)

$$\Pi_1 \equiv \Pi\left[q \geq q_n \,|\, \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] = \sum_{\mathcal{S}:|\mathcal{S}|\geq q_n} \pi(\mathcal{S}) \frac{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f \,|\, \mathcal{S})}{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f)}.$$

It turns out from the proof of Theorem 52 that

$$\Pi_1 \leq \frac{\sum_{q\geq q_n} \sum_{\mathcal{S}:|\mathcal{S}|=q} \pi(\mathcal{S})}{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f)} \lesssim \mathrm{e}^{c\, n\varepsilon_{n,\mathcal{S}_0}^2} \Pi(q \geq q_n).$$

In the proof of Lemma 1.1 we have already showed (under the assumptions of Theorem 4.1) that $\Pi(q \geq q_n) \lesssim \mathrm{e}^{-n\, \varepsilon_{n,\mathcal{S}_0}^2 C}$ for some $C > 0$. Choosing C_q large enough, one concludes that $\Pi_1 \to 0$ as $n \to \infty$. This shows that the ABC posterior concentrates on the sieve of models $\mathcal{F}_n$ with up to q_n covariates. Using this result, we can focus on models of size up to q_n and show that the posterior probability of over-fitting models goes to zero. Indeed, on the event $\mathcal{A}$ and on $\mathcal{F}_n$ we have (using an inequalities (4) and (6))

$$\begin{aligned}\Pi\left[\{\mathcal{S} \supset \mathcal{S}_0\} \cap \mathcal{F}_n \,|\, \|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T\right] &\leq \frac{\sum_{\mathcal{S}\supset\mathcal{S}_0:|\mathcal{S}|\leq q_n} \pi(\mathcal{S})}{\int \mathbb{P}_f[\|\boldsymbol{Y} - \boldsymbol{Y}^\star\|_n \leq \epsilon_n^T] d\Pi(f)} \\ &\lesssim \mathrm{e}^{(c-C)\, n\varepsilon_{n,\mathcal{S}_0}^2} \left(\frac{2\mathrm{e}p}{q_n}\right)^{q_n+1} \lesssim \mathrm{e}^{-H\, n\varepsilon_{n,\mathcal{S}_0}^2}\end{aligned}$$

for some $H > 0$ with $C > 0$ is large enough. This concludes that the ABC posterior will

lead to consistent variable selection as well.

We now discuss how the theory can be extended when data-splitting is deployed in ABC. First, we discuss the case when the split is done only once before applying ABC (not internally at each iteration). Denote with n_1 the training sample size and with n_2 the validation sample size. In order for the consistency result in Theorem 52 to hold, we need to make sure that prior concentration holds in the sense that $\Pi[\|f-f_0\|_{n_2} \lesssim \epsilon^T_{n_2}] \geq \mathrm{e}^{-c\, n_2 \epsilon^2_{n_2,\mathcal{S}_0}}$ for some $c>0$. Leaving n_1 data-points for training the prior, we know (from results in RP17 under fixed σ^2) that the posterior concentrates at the optimal rate (up to a log factor), i.e.

$$\Pi[\|f-f_0\|_{n_1} \lesssim \epsilon_{n_1,\mathcal{S}_0} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}}, \mathcal{S}_0] \to 1 \quad \text{as } n_1 \to \infty.$$

Choosing n_1 and n_2 in such a way so that $\epsilon_{n_1,\mathcal{S}_0} \lesssim \epsilon^T_{n_2} \equiv \sqrt{2(\log n_2)/n_2}$ (and assuming that observed fixed covariates in the training and testing sets are close), the prior concentration condition will be satisfied and the ABC will be consistent and concentrate at the rate λ_{n_2}. This implies variable selection consistency of our ABC method under identifiability and irrepresentabilty conditions which depend on λ_{n_2}. A similar conclusion is obtained for the expected posterior prior (6.9) where

$$\Pi[\|f-f_0\|_{n_1} \lesssim \epsilon_{n_1,\mathcal{S}_0}] \geq \pi(\mathcal{S}_0)\frac{1}{L}\sum_l \Pi[\|f-f_0\|_{n_1} \lesssim \epsilon_{n_1,\mathcal{S}_0} \mid \boldsymbol{Y}^{(n)}_{\mathcal{I}_l}, \mathcal{S}_0] \gtrsim \pi(\mathcal{S}_0).$$

A rigorous proof of ABC consistency for the expected posterior priors would require more care and will be left for future investigation.

6.8.2 Spike-and-Forests: MCMC Variant

As a precursor to ABC Bayesian Forests, we first implemented an MCMC algorithm for joint sampling from a posterior $\Pi(\mathcal{S}, \mathcal{E} \mid \boldsymbol{Y}^{(n)})$ over the space of models and tree ensemble partitions. We refer to this algorithm as Spike-and-Forests. The sampling follows a

Metropolis-Hasting scheme, exploiting the additive structure of forests by sampling each tree individually from conditionals in a Gibbs manner within each Metropolis step (Bayesian backfitting by Chipman et al. (2010)). The key is assigning a joint proposal distribution $pr(\mathcal{S}, \mathcal{E} \mid \mathcal{S}_m, \mathcal{E}_m) = pr(\mathcal{S} \mid \mathcal{S}_m) pr(\mathcal{E} \mid \mathcal{S}, \mathcal{E}_m)$ over variable subsets $\mathcal{S}$ and partition ensembles $\mathcal{E}$, where $\mathcal{S}_m$ and $\mathcal{E}_m$ are current MCMC states.

We explain the proposal mechanism using a single tree and write $\mathcal{T}$ instead of $\mathcal{E}$. First, a model proposal $\mathcal{S}^\star$ is sampled from $pr(\mathcal{S} \mid \mathcal{S}_m)$ which consists of the following three options: `add`, `delete` and `stay` for adding/deleting one (or none) of the variables. These three steps are chosen with probabilities $0.4, 0.4$ and 0.2, respectively. Candidate variables for deletion/addition are chosen from a uniform distribution. Given the newly suggested model $\mathcal{S}^\star$, the proposal distribution $pr(\mathcal{T} \mid \mathcal{S}^\star, \mathcal{T}_m)$ consists of various moves, described below, depending on the status of $\mathcal{S}^\star$.

If $\mathcal{S}^\star$ was obtained from $\mathcal{S}_m$ by `adding` a variable, the proposal $pr(\mathcal{T} \mid \mathcal{S}^\star = \texttt{add}, \mathcal{T}_m)$ consists of two steps: `birth` and `replace`. In the `birth` step, a bottom node is added to $\mathcal{T}_m$ and in the `replace` step one of the variables that occurs more than once inside $\mathcal{T}_m$ is replaced with the new variable. The birth step increases the size of the tree, while the replace step does not. The two steps are chosen with probabilities

$$\pi_{\text{birth,add}} = 0.7 \min\left\{\frac{\pi(K+1)}{\pi(K)}, 1\right\}, \pi_{\text{birth, replace}} = 1 - \pi_{\text{birth,add}},$$

where K is the number of bottom nodes in $\mathcal{T}_m$ and $\pi(K)$ is a prior on the number of bottom nodes. If no variable appears more than once in the tree, then `replace` is invalid and $\pi_{\text{birth, replace}}$ is set to 0.

If $\mathcal{S}^\star$ is obtained from $\mathcal{S}_m$ by `deleting` a variable, the proposal $pr(\mathcal{T} \mid \mathcal{S}^\star = \texttt{delete}, \mathcal{T}_m)$ consists of two steps: `death` and `replace`. If the variable chosen for deletion occurs in a bottom node, it can be removed from a tree $\mathcal{T}_m$ with a `delete` step that erases the bottom node. If the variable occurs inside the tree, it can be deleted by replacing it with other

variables in the `replace` step. If both of these moves are eligible, we pick one of them with probabilities

$$\pi_{\text{death,delete}} = 0.7 \min\left\{\frac{\pi(K-1)}{\pi(K)}, 1\right\}, \pi_{\text{death,replace}} = 1 - \pi_{\text{death,delete}}.$$

If the variable suggested for deletion is not in a bottom node, then $\pi_{\text{death,delete}} = 0$.

If the pool of variables stays the same, i.e. $\mathcal{S}^\star = \mathcal{S}_m$, the proposal $pr(\mathcal{T} \,|\, \mathcal{S}^\star = \texttt{stay}, \mathcal{T}_m)$ consists of 4 moves: `add, delete, replace` and `rule`. All proposal moves, and their probabilities, are adopted from Bayesian CART of Denison et al. (1998a). These steps only modify the tree configuration without adding/deleting variables.

Regarding the prior distributions for our MCMC implementation, we assume the beta-binomial prior on the variable subsets. Namely, for binary indicators $\gamma_j \in \{0, 1\}$, for whether or not x_j is active, we assume $\mathbb{P}(\gamma_j = 1 \,|\, \theta) = \theta$ and $\theta \sim \mathcal{B}(a, b)$. The prior distribution on trees consists of (a) the truncated Poisson distribution on the number of bottom leaves, (b) uniform prior over trees with the same number of leaves and (c) standard Gaussian prior on the step sizes. This is the Bayesian CART prior proposed by Denison et al. (1998a) and analyzed theoretically by Ročková and van der Pas (2020). In the computation of MH acceptance ratios, we leverage the fact that the bottom leave parameters can be integrated out to obtain a conditional marginal likelihood, given each partition.

The MCMC sampling routine can be extended to spike-and-forests, altering each tree inside the forests one by one through Bayesian backfitting (Chipman et al., 2010). One big advantage of the Bayesian forest representation is that it accelerates mixing since most trees are shallow and thereby more easily modified throughout MCMC (see Pratola (2016)).

REFERENCES

Sungjin Ahn, Anoop Korattikara Balan, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1591–1598, 2012.

Mattias Akesson, Prashant Singh, Fredrik Wrede, and Andreas Hellander. Convolutional neural networks as summary statistics for approximate Bayesian computation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

Carlo Albert, Hans R Künsch, and Andreas Scheidegger. A simulated annealing approach to approximate Bayes computations. *Statistics and computing*, 25(6):1217–1232, 2015.

Johan Alenlov, Arnaud Doucet, and Fredrik Lindsten. Pseudo-marginal hamiltonian monte carlo. *Journal of Machine Learning Research*, 22(141), 2021.

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

Luca Ambrogioni, Umut Güçlü, Marcel AJ van Gerven, and Eric Maris. The kernel mixture network: A nonparametric method for conditional density estimation of continuous random variables. *arXiv preprint arXiv:1705.07111*, 2017.

Luca Ambrogioni, Umut Güçlü, Yağmur Güçlütürk, Max Hinne, Marcel A van Gerven, and Eric Maris. Wasserstein variational inference. *Advances in Neural Information Processing Systems*, 31, 2018.

Ziwen An, David J Nott, and Christopher Drovandi. Robust Bayesian synthetic likelihood via a semi-parametric approach. *Statistics and Computing*, 30(3):543–557, 2020.

George E Andrews. *The Theory of Partitions*. Cambridge University Press, 1976.

Christophe Andrieu and Gareth O Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*, volume 9. Cambridge University Press, 1999.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Kellie J. Archer and Ryan V. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260, 2008.

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

Artin Armagan, Merlise Clyde, and David B Dunson. Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems*, pages 523–531, 2011.

Artin Armagan, David B Dunson, and Jaeyong Lee. Generalized double pareto shrinkage. *Statistica Sinica*, 23(1):119, 2013.

Susan Athey and Stefan Wager. Efficient policy learning. Technical report, 2017.

Susan Athey, Guido W Imbens, Jonas Metzger, and Evan Munro. Using Wasserstein generative adversarial networks for the design of Monte Carlo simulations. *Journal of Econometrics*, 2021.

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2019.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

Randall Balestriero and Richard G. Baraniuk. A spline theory of deep learning. In *International Conference on Machine Learning*, pages 374–383, 2018.

Rina Foygel Barber and Emmanuel J Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.

Stuart Barber, Jochen Voss, and Mark Webster. The rate of convergence for approximate Bayesian computation. *Electronic Journal of Statistics*, 9(1):80–105, 2015.

Maria M Barbieri, James O Berger, Edward I George, and Veronika Ročková. The median probability model and correlated variables. *Bayesian Analysis*, 16(4):1085–1112, 2021.

Maria Maddalena Barbieri and James O Berger. Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897, 2004.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Andrew R Barron and Jason M Klusowski. Approximation and estimation for high-dimensional deep learning networks. *arXiv*, 2018.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing systems*, 30, 2017.

Benedikt Bauer and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47(4):2261–2285, 2019.

Mark A Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

James O Berger and Luis R Pericchi. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

James O Berger and Luis R Pericchi. Training samples in objective Bayesian model selection. *The Annals of Statistics*, 32(3):841–869, 2004.

Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. Approximate Bayesian computation with the Wasserstein distance. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(2):235–269, 2019.

Karine Bertin and Guillaume Lecu\'e. Selection of variables and dimension reduction in high-dimensional non-parametric regression. *Electronic Journal of Statistics*, 2:1224–1241, 2008.

Anindya Bhadra, Jyotishka Datta, Nick Polson, Vadim Sokolov, and Jianeng Xu. Merging two cultures: deep and statistical learning. *arXiv preprint arXiv:2110.11561*, 2021.

Anirban Bhattacharya, Debdeep Pati, Natesh S Pillai, and David B Dunson. Dirichlet–Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110 (512):1479–1490, 2015.

Anirban Bhattacharya, Antik Chakraborty, and Bani K Mallick. Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika*, page asw042, 2016.

Gérard Biau, Erwan Scornet, and Johannes Welbl. Neural random forests. *Sankhya A*, pages 1–40, 2016.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, pages 81–88, 2007.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006.

CM Bishop. Mixture density networks. *Technical Report*, 1994.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 78(5):1103, 2016.

Justin Bleich, Adam Kapelner, Edward I George, and Shane T Jensen. Variable selection for BART: An application to gene regulation. *The Annals of Applied Statistics*, pages 1750–1781, 2014.

Michael GB Blum. Regression approaches for ABC. In *Handbook of Approximate Bayesian Computation*, pages 71–85. Chapman and Hall/CRC, 2018.

Michael GB Blum and Olivier François. Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

Michael GB Blum, Maria Antonieta Nunes, Dennis Prangle, and Scott A Sisson. A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, 28(2):189–208, 2013.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman and Adele Cutler. Online manual for random forests. 2013. URL `www.stat.berkeley.edu/~breiman/RandomForests/cc_home.html`.

Leo Breiman, Jerome Friedman, RA Olshen, and Charles J Stone. *Classification and regression trees*. New York: Chapman and Hall, 1984.

Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems*. SIAM, 2009.

Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems: revised reprint*. SIAM, 2012.

Cristina Butucea. Goodness-of-fit testing and quadratic functional estimation from indirect observations. *The Annals of Statistics*, 35(5):1907–1930, 2007.

T Tony Cai and Mark G Low. On adaptive estimation of linear functionals. *The Annals of Statistics*, 33(5):2311–2343, 2005.

T Tony Cai and Mark G Low. Adaptive confidence balls. *The Annals of Statistics*, 34(1): 202–228, 2006a.

T Tony Cai and Mark G Low. Optimal adaptive estimation of a quadratic functional. *The Annals of Statistics*, 34(5):2298–2325, 2006b.

Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold:'model-x' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

P. Carbonetto and M. Stephens. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Analysis*, 7(1):73–108, 2012.

Peter Carr and Liuren Wu. The finite moment log stable process and option pricing. *The Journal of Finance*, 58(2):753–777, 2003.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

Ismaël Castillo. A semiparametric Bernstein-von Mises theorem for Gaussian process priors. *Probability Theory and Related Fields*, 152(1-2):53–99, 2012a.

Ismaël Castillo. Semiparametric Bernstein-von Mises theorem and bias, illustrated with Gaussian process priors. *Sankhya A*, 74(2):194–221, 2012b.

Ismaël Castillo and Romain Mismer. Empirical Bayes analysis of spike and slab posterior distributions. *Electronic Journal of Statistics*, 12:3953–4001, 2018.

Ismaël Castillo and Richard Nickl. Nonparametric Bernstein–von Mises theorems in Gaussian white noise. *The Annals of Statistics*, 41(4):1999–2028, 2013.

Ismaël Castillo and Richard Nickl. On the Bernstein-von Mises phenomenon for nonparametric Bayes procedures. *The Annals of Statistics*, 42(5):1941–1969, 2014.

Ismael Castillo and Veronika Ročková. Uncertainty quantification for bayesian cart. *The Annals of Statistics*, 49(6):3482–3509, 2021.

Ismaël Castillo and Judith Rousseau. A Bernstein–von Mises theorem for smooth functionals in semiparametric models. *The Annals of Statistics*, 43(6):2353–2383, 2015.

Ismaël Castillo and Aad van der Vaart. Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *The Annals of Statistics*, 40(4):2069–2101, 2012.

Ismaël Castillo, Johannes Schmidt-Hieber, and Aad van der Vaart. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018, 2015.

John M Chambers, Colin L Mallows, and BW4159820341 Stuck. A method for simulating stable random variables. *Journal of the American Statistical Association*, 71(354):340–344, 1976.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Xiaohong Chen and Halbert White. Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2): 682–691, 1999.

Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Gaussian approximation of suprema of empirical processes. *Annals of Statistics*, 42(4):1564–1597, 2014.

Siddhartha Chib. Bayes inference in the Tobit censored regression model. *Journal of Econometrics*, 51(1-2):79–99, 1992.

Siddhartha Chib and Edward Greenberg. Markov chain Monte Carlo simulation methods in econometrics. *Econometric Theory*, 12(3):409–431, 1996.

Hugh Chipman, Edward I George, and Robert E McCulloch. The practical implementation of Bayesian model selection. In *Model Selection*, pages 65–116. Institute of Mathematical Statistics, 2001.

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Merlise A Clyde and Edward I George. Empirical Bayes estimation in wavelet nonparametric regression. In *Bayesian inference in wavelet-based models*, pages 309–322. Springer, 1999.

Olivier Collier, Laëtitia Comminges, and Alexandre B Tsybakov. Minimax estimation of linear and quadratic functionals on sparsity classes. *The Annals of Statistics*, 45(3):923–958, 2017.

Laëtitia Comminges and Arnak S Dalalyan. Tight conditions for consistency of variable selection in the context of high dimensionality. *The Annals of Statistics*, 40(5):2667–2696, 2012.

Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of Information Theory*, 2(1):12–13, 1991.

Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2015.

Katalin Csillery, Michael GB Blum, Oscar E Gaggiotti, and Olivier Francois. Approximate Bayesian computation (ABC) in practice. *Trends in ecology \& evolution*, 25(7):410–418, 2010.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

Andrew Czarn, Cara MacNish, Kaipillil Vijayan, Berwin Turlach, and Ritu Gupta. Statistical exploratory analysis of genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):405–421, 2004.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 207–215, 2013.

A Philip Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

George Deligiannidis, Arnaud Doucet, and Michael K Pitt. The correlated pseudomarginal method. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80 (5):839–870, 2018.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Frank Den Hollander. *Large deviations*, volume 14. American Mathematical Soc., 2008.

Wei Deng, Xiao Zhang, Faming Liang, and Guang Lin. An adaptive empirical Bayesian method for sparse deep learning. In *Advances in Neural Information Processing Systems*, pages 5564–5574, 2019.

David GT Denison, Bani K Mallick, and Adrian FM Smith. A Bayesian CART algorithm. *Biometrika*, 85(2):363–377, 1998a.

DGT Denison, BK Mallick, and AFM Smith. Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):333–350, 1998b.

Xavier Didelot, Richard G Everitt, Adam M Johansen, and Daniel J Lawson. Likelihood-free estimation of model evidence. *Bayesian Analysis*, 6(1):49–76, 2011.

Peter J Diggle and Richard J Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2): 193–212, 1984.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

Sophie Donnet, Vincent Rivoirard, Judith Rousseau, and Catia Scricciolo. Posterior concentration rates for empirical Bayes procedures with applications to Dirichlet process mixtures. *Bernoulli*, 24(1):231–256, 2018.

David Donoho. 50 years of data science. *Journal of Computational and Graphical Statistics*, 26(4):745–766, 2017.

Arnaud Doucet, Michael K Pitt, George Deligiannidis, and Robert Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313, 2015.

Christopher Drovandi and David T Frazier. A comparison of likelihood-free methods with and without summary statistics. *Statistics and Computing*, 32(3):1–23, 2022.

Christopher C Drovandi, Anthony N Pettitt, and Malcolm J Faddy. Approximate Bayesian computation using indirect inference. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60(3):317–337, 2011.

Leo L Duan, James E Johndrow, and David B Dunson. Scaling up data augmentation MCMC via calibration. *Journal of Machine Learning Research*, 19(1):2575–2608, 2018.

Lutz Dümbgen. New goodness-of-fit tests and their application to nonparametric confidence sets. *The Annals of Statistics*, 26(1):288–314, 1998.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017.

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, pages 2771–2781. PMLR, 2020.

Vincent Dutordoir, Hugh Salimbeni, James Hensman, and Marc Deisenroth. Gaussian process conditional density estimation. *Advances in Neural Information Processing systems*, 31, 2018.

David Eric Edmunds and Hans Triebel. *Function spaces, entropy numbers, differential operators*, volume 120. Cambridge University Press, 2008.

Sam Efromovich and Mark G Low. On optimal adaptive estimation of a quadratic functional. *The Annals of Statistics*, 24(3):1106–1125, 1996.

Bradley Efron. Bayesian inference and the parametric bootstrap. *The Annals of Applied Statistics*, 6(4):1971, 2012.

Bradley Efron and Robert Tibshirani. The problem of regions. *The Annals of Statistics*, pages 1687–1718, 1998.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan, Cong Ma, and Yiqiao Zhong. A selective overview of deep learning. *Statistical Science*, 36(2):264–290, 2021.

Yanan Fan, David J Nott, and Scott A Sisson. Approximate Bayesian computation via regression density estimation. *Stat*, 2(1):34–48, 2013.

Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

Matteo Fasiolo, Simon N Wood, Florian Hartig, and Mark V Bravington. An extended empirical saddlepoint approximation for intractable likelihoods. *Electronic Journal of Statistics*, 12(1):1544–1578, 2018.

Matteo Fasiolo, Simon N Wood, Margaux Zaffran, Raphaël Nedellec, and Yannig Goude. Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, 116(535):1402–1412, 2021.

Paul Fearnhead and Dennis Prangle. Constructing ABC summary statistics: semi-automatic ABC. *Nature Precedings*, pages 1–1, 2011.

Joseph Felsenstein. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39(4):783–791, 1985.

Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between'uncertainty in Bayesian neural networks. In *Uncertainty in Deep Learning Workshop, ICML*, 2019.

David T Frazier. Robust and efficient approximate bayesian computation: A minimum distance approach. *arXiv preprint arXiv:2006.14126*, 2020.

David T Frazier, Gael M Martin, Christian P Robert, and Judith Rousseau. Asymptotic properties of approximate Bayesian computation. *Biometrika*, 105(3):593–607, 2018.

David T Frazier, Christian Robert, and Judith Rousseau. Model misspecification in ABC: Consequences and diagnostics. *Journal of the Royal Statistical Society: Series B*, 2019.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. New York: Springer Series in Statistics, 2001.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19 (1):1–67, 1991.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of statistics*, pages 1189–1232, 2001.

Yun-Xin Fu and Wen-Hsiung Li. Estimating the age of the common ancestor of a sample of DNA sequences. *Molecular Biology and Evolution*, 14(2):195–199, 1997.

Masahiro Fujisawa, Takeshi Teshima, Issei Sato, and Masashi Sugiyama. γ-abc: Outlier-robust approximate Bayesian computation based on a robust divergence estimator. In *International Conference on Artificial Intelligence and Statistics*, pages 1783–1791. PMLR, 2021.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

Zhe Gan, Ricardo Henao, David Carlson, and Lawrence Carin. Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, pages 268–276, 2015.

Chao Gao and Harrison H Zhou. Bernstein-von Mises theorems for functionals of the covariance matrix. *Electronic Journal of Statistics*, 10(2):1751–1806, 2016.

Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class project for Stanford CS231N: convolutional neural networks for visual recognition, Winter semester*, 2014(5):2, 2014.

Kishore Gawande. Comparing theories of endogenous protection: Bayesian comparison of Tobit models using Gibbs sampling output. *Review of Economics and Statistics*, 80(1): 128–140, 1998.

Ghislaine Gayraud and Yuri Ingster. Detection of sparse additive functions. *Electronic Journal of Statistics*, 6:1409–1448, 2012.

Donald Geman and George Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):367–383, 1992.

Stuart Geman and Chii-Ruey Hwang. Nonparametric maximum likelihood estimation by the method of sieves. *The Annals of Statistics*, pages 401–414, 1982.

Stuart Geman and Chii-Ruey Hwang. Diffusions for global optimization. *SIAM Journal on Control and Optimization*, 24(5):1031–1043, 1986.

Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Edward I George and Robert E McCulloch. Approaches for Bayesian variable selection. *Statistica sinica*, pages 339–373, 1997.

Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. PAC-Bayesian theory meets Bayesian inference. *Advances in Neural Information Processing Systems*, 29, 2016.

Charles J Geyer. Estimation and optimization of functions. In *Markov chain Monte Carlo in practice*, pages 241–258. Chapman and Hall, 1996.

Sandesh Ghimire, Aria Masoomi, and Jennifer Dy. Reliable estimation of KL divergence using a discriminator in reproducing kernel Hilbert space. *Advances in Neural Information Processing Systems*, 34:10221–10233, 2021.

Subhashis Ghosal and Aad Van Der Vaart. Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, 35(1):192–223, 2007.

Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

Subhashis Ghosal, Jayanta K Ghosh, and Aad W Van Der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, pages 500–531, 2000.

Subhashis Ghosal, Jüri Lember, and Aad van der Vaart. Nonparametric Bayesian model selection and averaging. *Electronic Journal of Statistics*, 2:63–89, 2008.

Jayanta K Ghosh and Tapas Samanta. Nonsubjective bayes testing–an overview. *Journal of statistical planning and inference*, 103(1-2):205–223, 2002.

Daniel T Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

Chris Glynn, Surya T Tokdar, Brian Howard, and David L Banks. Bayesian analysis of dynamic linear topic models. *Bayesian Analysis*, 14(1):53–80, 2019.

Michael Goldstein. Subjective Bayesian analysis: principles and practice. *Bayesian Analysis*, 1(3):403–420, 2006.

Isidore Jacob Good. Probability and the weighing of evidence. 1950.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

R. Gramacy and H. Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103:1119–11303, 2008a.

Robert B Gramacy and Herbert K H Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103 (483):1119–1130, 2008b.

Robert B Gramacy and Nicholas G Polson. Simulation-based regularized logistic regression. *Bayesian Analysis*, 7(3):567–590, 2012.

Peter J Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170, 1984.

Peter J Green. Reversible jump markov chain monte carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Peter J Green and David I Hastie. Reversible jump MCMC. *Genetics*, 155(3):1391–1403, 2009.

Aude Grelaud, Christian P Robert, Jean-Michel Marin, François Rodolphe, and Jean-François Taly. Abc likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis*, 4(2):317–335, 2009.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773, 2012.

Y. Guan and M. Stephens. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *The Annals of Applied Statistics*, 5:1780–1815, 2011.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

Maya Gupta, Erez Louidor, Oleksandr Mangylov, Nobu Morioka, Taman Narayan, and Sen Zhao. Multidimensional shape constraints. In *International Conference on Machine Learning*, pages 3918–3928. PMLR, 2020.

Michael Gutmann and Aapo Hyvärinen. Learning features by contrasting natural images with noise. In *International Conference on Artificial Neural Networks*, pages 623–632. Springer, 2009.

Michael U Gutmann and Jukka Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 2016.

Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2), 2012.

Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pages 2596–2604, 2019.

Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.

Chris Hans, Adrian Dobra, and Mike West. Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, 102(478):507–516, 2007.

Wolfgang H\"ardle. *Applied nonparametric regression*. Number 19. Cambridge university press, 1990.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. 2019a.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Training dynamics of deep networks using stochastic gradient descent via neural tangent kernel. *arXiv*, 2019b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

JB Heaton, NG Polson, and Jan Hendrik Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1):3–12, 2017.

Pashupati Hegde, Markus Heinonen, Harri Lähdesmäki, and Samuel Kaski. Deep learning with differential gaussian process flows. In *International Conference on Artificial Intelligence and Statistics*, pages 1812–1821, 2019.

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

J. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20:217–240, 2011.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012a.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012b.

Guang-Bin Huang. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3):376–390, 2014.

Guang-Bin Huang. What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle. *Cognitive Computation*, 7(3):263–278, 2015.

Guang-Bin Huang, Lei Chen, and Chee Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.

Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

Ildar Abdullovich Ibragimov and Rafail Zalmanovich Khasminskii. On nonparametric estimation of the value of a linear functional in Gaussian white noise. *Theory of Probability & Its Applications*, 29(1):18–32, 1985.

Hemant Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

V Ismailov. Approximation by sums of ridge functions with fixed directions. *St. Petersburg Mathematical Journal*, 28(6):741–772, 2017.

Eric Jacquier, Michael Johannes, and Nicholas Polson. MCMC maximum likelihood for latent state models. *Journal of Econometrics*, 137(2):615–640, 2007.

Marko Järvenpää, Michael U Gutmann, Arijus Pleska, Aki Vehtari, and Pekka Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *Bayesian Analysis*, 14(2):595–622, 2019.

Marko Jarvenpaa, Aki Vehtari, and Pekka Marttinen. Batch simulations and uncertainty quantification in Gaussian process surrogate approximate Bayesian computation. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, volume 124, pages 779–788. PMLR, 2020.

Vinay Jethava and Devdatt Dubhashi. Easy high-dimensional likelihood-free inference. *arXiv preprint arXiv:1711.11139*, 2017.

B. Jiang, T. Wu, C. Zheng, and W. Wong. Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, 27:1595–1618, 2017a.

Bai Jiang, Tung-yu Wu, Charles Zheng, and Wing H Wong. Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618, 2017b.

Bai Jiang, Tung-Yu Wu, and Wing Hung Wong. Approximate Bayesian computation with Kullback-Leibler divergence as data discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1711–1721. PMLR, 2018.

Valen E Johnson and David Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Paul Joyce and Paul Marjoram. Approximately sufficient statistics and Bayesian computation. *Statistical applications in Genetics and Molecular Biology*, 7(1), 2008.

Tetsuya Kaji and Veronika Ročková. Metropolis-Hastings via classification. *Journal of the American Statistical Association*, pages 1–33, 2022.

Tetsuya Kaji, Elena Manresa, and Guillaume Pouliot. An adversarial approach to structural estimation. *arXiv preprint arXiv:2007.06169*, 2020.

Olav Kallenberg. *Foundations of Modern Probability*, volume 2. Springer, 2002.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv*, 2017.

Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In *Advances in Neural Information Processing Systems*, pages 425–434, 2017.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30:5574–5584, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Bas JK Kleijn and Aad W van der Vaart. Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics*, 34(2):837–877, 2006.

Bas JK Kleijn and Aad W van der Vaart. The Bernstein-von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6:354–381, 2012.

Jason M Klusowski and Andrew R Barron. Risk bounds for high-dimensional ridge function combinations including neural networks. *arXiv*, 2016.

Jason M Klusowski and Andrew R Barron. Minimax lower bounds for ridge combinations including neural nets. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1376–1380. IEEE, 2017.

Andrei Nikolaevich Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*, volume 114, pages 953–956. Russian Academy of Sciences, 1957.

Vladimir Koltchinskii and Ming Yuan. Sparsity in multiple kernel learning. *The Annals of Statistics*, pages 3660–3695, 2010.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing systems*, pages 1097–1105, 2012.

Chung-Ming Kuan and Halbert White. Artificial neural networks: an econometric perspective. *Econometric Reviews*, 13(1):1–91, 1994.

Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1958.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Suprateek Kundu, Bani K Mallick, and Veera Baladandayuthapani. Efficient Bayesian regularization for graphical model selection. *Bayesian Analysis*, 2018.

John Lafferty and Larry Wasserman. Iterative Markov Chain Monte Carlo computation of reference priors and minimax risk. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 293–300. Morgan Kaufmann Publishers Inc., 2001.

John Lafferty and Larry Wasserman. RODEO: sparse, greedy nonparametric regression. *The Annals of Statistics*, pages 28–63, 2008.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Kenneth Lange. The MM algorithm. In *Optimization*, pages 185–219. Springer, 2013a.

Kenneth Lange. *Optimization*, volume 95. Springer Science & Business Media, 2013b.

Kenneth Lange, David R Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

Béatrice Laurent, Carenne Ludena, and Clémentine Prieur. Adaptive estimation of linear functionals by model selection. *Electronic journal of statistics*, 2:993–1020, 2008.

Holden Lee, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. On the ability of neural nets to express distributions. In *Conference on Learning Theory*, pages 1271–1296, 2017.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Jüri Lember and Aad van der Vaart. On universal Bayesian adaptation. *Statistics & Decisions*, 25(2/2007):127–152, 2007.

Fred B Lempers. Posterior probabilities of alternative linear models. 1971.

Wentao Li and Paul Fearnhead. Convergence of regression-adjusted approximate Bayesian computation. *Biometrika*, 105(2):301–318, 2018.

Yingzhen Li and Yarin Gal. Dropout inference in Bayesian neural networks with alpha-divergences. In *International Conference on Machine Learning*, volume 70, pages 2052–2061. JMLR. org, 2017.

Yinpu Li, Antonio R Linero, and Jared Murray. Adaptive conditional distribution estimation with Bayesian decision tree ensembles. *Journal of the American Statistical Association*, pages 1–14, 2022.

Faming Liang, Qizhai Li, and Lei Zhou. Bayesian neural networks for selection of drug sensitive genes. *Journal of the American Statistical Association*, 113(523):955–972, 2018.

Shiyu Liang and R Srikant. Why deep neural networks for function approximation? *International Conference on Learning Representations*, 2017.

Tengyuan Liang. How well generative adversarial networks learn distributions. *Journal of Machine Learning Research*, 22:1–41, 2021.

Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R news*, 2 (3):18–22, 2002.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7:13276, 2016.

Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

Antonio R Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, pages 1–11, 2018.

Antonio Ricardo Linero and Yun Yang. Bayesian regression tree ensembles that adapt to smoothness and sparsity. *arXiv preprint arXiv:1707.09461*, 2017.

Jarno Lintusaari, Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, 66(1):e66–e82, 2017.

Jeremiah Zhe Liu. Variable selection with rigorous uncertainty quantification using deep Bayesian neural networks: Posterior concentration and Bernstein-von Mises phenomenon. pages 3124–3132, 2021.

Yi Liu, Veronika Ročková, and Yuexi Wang. Variable selection with abc bayesian forests. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3):453–481, 2021.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in Neural Information Processing Systems*, 30, 2017.

Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H Macke. Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. PMLR, 2019.

Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42): 20881–20885, 2019.

David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

Malik Magdon-Ismail and Amir F Atiya. A maximum likelihood approach to volatility estimation for a Brownian motion using high, low and close price data. *Quantitative Finance*, 3(5):376, 2003.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Bani K Mallick, Debashis Ghosh, and Malay Ghosh. Bayesian classification of tumours by using gene expression data. *Journal of the Royal Statistical Society: Series B*, 67(2): 219–234, 2005.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(1):4873–4907, 2017.

Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.

Gael M Martin, Brendan PM McCabe, David T Frazier, Worapree Maneesoonthorn, and Christian P Robert. Auxiliary likelihood-based approximate Bayesian computation in state space models. *Journal of Computational and Graphical Statistics*, 28(3):508–522, 2019.

James S Martin, Ajay Jasra, Sumeetpal S Singh, Nick Whiteley, Pierre Del Moral, and Emma McCoy. Approximate Bayesian computation for smoothing. *Stochastic Analysis and Applications*, 32(3):397–420, 2014.

Robert McCulloch, Rodney Sparapani, Robert Gramacy, Charles Spanbauer, and Matthew Pratola. *BART: Bayesian Additive Regression Trees*, 2018. URL `https://CRAN.R-project.org/package=BART`. R package version 1.6.

Patrick L McDermott and Christopher K Wikle. Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*, 30(3):e2553, 2019.

Edward Meeds and Max Welling. Gps-abc: Gaussian process surrogate approximate Bayesian computation. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 593–602, 2014.

Kerrie L Mengersen, Pierre Pudlo, and Christian P Robert. Bayesian computation via empirical likelihood. *Proceedings of the National Academy of Sciences*, 110(4):1321–1326, 2013.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*, pages 2391–2400. PMLR, 2017.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *AAAI*, pages 2343–2349, 2017.

Jeffrey W Miller and David B Dunson. Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, 114(527):1113–1125, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Hadrien Montanelli and Qiang Du. New error bounds for deep ReLU networks using sparse grids. *SIAM Journal on Mathematics of Data Science*, 1(1):78–92, 2019.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2924–2932, 2014.

Elías Moreno, Javier Girón, and George Casella. Posterior model consistency in variable selection as the model dimension grows. *Statistical Science*, 30(2):228–241, 2015.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, pages 429–443, 1997.

Iain Murray and Matthew Graham. Pseudo-marginal slice sampling. In *Artificial Intelligence and Statistics*, pages 911–919, 2016.

Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, pages 541–548, 2010.

Naveen Naidu Narisetty and Xuming He. Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817, 2014.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in Neural Information Processing Systems*, pages 475–482, 1993.

Radford M Neal. Slice sampling. *The Annals of Statistics*, pages 705–741, 2003.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *International Conference on Learning Representations*, 2017.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o $(1/k^2)$. In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

Michael A Newton and Adrian E Raftery. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994.

Michael A Newton, Nicholas G Polson, and Jianeng Xu. Weighted bayesian bootstrap for scalable posterior distributions. *Canadian Journal of Statistics*, 49(2):421–437, 2021.

Hien Duy Nguyen, Julyan Arbel, Hongliang Lü, and Florence Forbes. Approximate Bayesian computation via the energy statistic. *IEEE Access*, 8:131683–131698, 2020.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 1089–1096, 2007.

David J Nott, Victor M-H Ong, Y Fan, and SA Sisson. High-dimensional ABC. In *Handbook of Approximate Bayesian Computation*, pages 211–241. Chapman and Hall/CRC, 2018.

Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992.

Elina Numminen, Lu Cheng, Mats Gyllenberg, and Jukka Corander. Estimating the transmission dynamics of Streptococcus pneumoniae from strain prevalence data. *Biometrics*, 69(3):748–757, 2013.

Matthew A Nunes and David J Balding. On optimal selection of summary statistics for approximate Bayesian computation. *Statistical applications in Genetics and Molecular Biology*, 9(1), 2010.

Anthony O'Hagan. Fractional bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):99–118, 1995.

Philip D O'Neill, David J Balding, Niels G Becker, Mervi Eerola, and Denis Mollison. Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(4): 517–542, 2000.

Vittorio Orlandi, Jared Murray, Antonio Linero, and Alexander Volfovsky. Density regression with Bayesian additive regression trees. *arXiv preprint arXiv:2112.12259*, 2021.

Colleen O'Ryan, Eric H Harley, Michael W Bruford, Mark Beaumont, Robert K Wayne, and Michael I Cherry. Microsatellite analysis of genetic diversity in fragmented South African buffalo populations. *Animal Conservation*, 1(2):85–94, 1998.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pages 1028–1036, 2016.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Mijung Park, Wittawat Jitkrittum, and Dino Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. In *Artificial Intelligence and Statistics*, pages 398–407. PMLR, 2016.

José M Pérez and James O Berger. Expected-posterior prior distributions for model selection. *Biometrika*, 89(3):491–512, 2002.

Fernando Pérez-Cruz. Kullback-Leibler divergence estimation of continuous distributions. In *2008 IEEE international Symposium on Information Theory*, pages 1666–1670. IEEE, 2008.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

David B Phillips and Adrian FM Smith. Bayesian model comparison via jump diffusions. *Markov Chain Monte Carlo in Practice*, 215:239, 1996.

Martin Pincus. A closed form solution of certain programming problems. *Operations Research*, 16(3):690–694, 1968.

Martin Pincus. A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research*, 18(6):1225–1228, 1970.

Vincent Plagnol and Simon Tavaré. Approximate Bayesian Computation and MCMC. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 99–113. Springer, 2004.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

Nicholas Polson and Vadim Sokolov. Deep learning: Computational aspects. *Wiley Interdisciplinary Reviews: Computational Statistics*, 12(5):e1500, 2020.

Nicholas Polson, Vadim Sokolov, and Jianeng Xu. Deep learning partial least squares. *arXiv preprint arXiv:2106.14085*, 2021.

Nicholas G Polson and Veronika Ročková. Posterior concentration for sparse deep learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 938–949, 2018.

Nicholas G Polson and James G Scott. Local shrinkage rules, Lévy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74 (2):287–311, 2012.

Nicholas G Polson and James G Scott. Data augmentation for Non-Gaussian regression models using variance-mean mixtures. *Biometrika*, 100(2):459–471, 2013.

Nicholas G Polson and Steven L Scott. Data augmentation for Support Vector Machines. *Bayesian Analysis*, 6(1):1–23, 2011.

Nicholas G Polson and Vadim Sokolov. Deep learning: a Bayesian perspective. *Bayesian Analysis*, 12(4):1275–1304, 2017.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108 (504):1339–1349, 2013.

Nicholas G Polson, James G Scott, and Brandon T Willard. Proximal algorithms in statistics and machine learning. *Statistical Science*, 30(4):559–581, 2015a.

Nicholas G Polson, Brandon T Willard, and Massoud Heidari. A statistical theory of deep learning via proximal splitting. *arXiv:1509.06061*, 2015b.

Matthew T Pratola. Efficient Metropolis–Hastings proposal mechanisms for Bayesian regression tree models. *Bayesian Analysis*, 11(3):885–911, 2016.

Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

Pierre Pudlo, Jean-Michel Marin, Arnaud Estoup, Jean-Marie Cornuet, Mathieu Gautier, and Christian P Robert. Reliable ABC model choice via random forests. *Bioinformatics*, 32(6):859–866, 2015.

Peter Radchenko and Gareth M James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105 (492):1541–1553, 2010.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008a.

Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE, 2008b.

Poornima Ramesh, Jan-Matthis Lueckmann, Jan Boelts, Álvaro Tejero-Cantero, David S Greenberg, Pedro J Gonçalves, and Jakob H Macke. GATSBI: Generative adversarial training for simulation-based inference. In *International Conference on Learning Representations*, 2022.

Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. *Advances in Neural Information Processing Systems*, 29, 2016.

Varun Ranganathan and S Natarajan. A new backpropagation algorithm without gradient descent. *arXiv:1802.00027*, 2018.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13(Feb):389–427, 2012.

Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

Kolyan Ray and Aad van der Vaart. Semiparametric Bayesian causal inference. *The Annals of Statistics*, 48(5):2999 – 3020, 2020.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

Soo-Yon Rhee, W Jeffrey Fessel, Andrew R Zolopa, Leo Hurley, Tommy Liu, Jonathan Taylor, Dong Phuong Nguyen, Sally Slome, Daniel Klein, and Michael Horberg. Hiv-1 protease and reverse-transcriptase mutations: correlations with antiretroviral therapy in subtype b isolates and implications for drug-resistance surveillance. *The Journal of infectious diseases*, 192(3):456–465, 2005.

Soo-Yon Rhee, Jonathan Taylor, Gauhar Wadhera, Asa Ben-Hur, Douglas L Brutlag, and Robert W Shafer. Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences*, 103(46):17355–17360, 2006.

William Edwin Ricker. Stock and recruitment. *Journal of the Fisheries Board of Canada*, 11(5):559–623, 1954.

Vincent Rivoirard and Judith Rousseau. Bernstein-von Mises theorem for linear functionals of the density. *The Annals of Statistics*, 40(3):1489–1523, 2012.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Christian P Robert, Jean-Marie Cornuet, Jean-Michel Marin, and Natesh S Pillai. Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences*, 108(37):15112–15117, 2011.

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

Veronika Ročková and Edward I George. The Spike-and-Slab LASSO. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

Veronika Ročková and Stéphanie van der Pas. Posterior concentration for Bayesian regression trees and forests. *Annals of Statistics*, 48(4):2108–2131, 2020.

Leonard CG Rogers and Fanyin Zhou. Estimating correlation from high, low, opening and closing prices. *The Annals of Applied Probability*, 18(2):813–823, 2008.

Yaniv Romano, Matteo Sesia, and Emmanuel J Candès. Deep knockoffs. *Journal of the American Statistical Association*, 115(532):1861–1872, 2020.

Judith Rousseau and Botond Szabo. Asymptotic behaviour of the empirical Bayes posteriors associated to maximum marginal likelihood estimator. *The Annals of Statistics*, 45(2): 833–865, 2017.

V. Ročková. Particle EM for variable selection. *Journal of the American Statistical Association*, 113(524):1684–1697, 2018.

V. Ročková and E. Saha. On theory for BART. In *Artificial Intelligence and Statistics*, pages 2839–2848, 2019.

Veronika Ročková. On semi-parametric inference for BART. In *International Conference on Machine Learning*, pages 8137–8146. PMLR, 2020.

Veronika Ročková and Edward I George. EMVS: The em approach to Bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846, 2014.

Veronika Ročková and Edward I George. The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable selection for nonparametric Gaussian process priors: Models and computational strategies. *Statistical Science*, 26(1): 130, 2011.

Fabian Scheipl. spikeSlabGAM: Bayesian variable selection, model choice and regularization for generalized additive mixed models in R. *Journal of Statistical Software*, (14):1–24, 2011.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

Sebastian M Schmon, Patrick W Cannon, and Jeremias Knoblauch. Generalized posteriors in approximate Bayesian computation. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Lorraine Schwartz. On Bayes procedures. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 4(1):10–26, 1965.

James G Scott and James O Berger. Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, pages 2587–2619, 2010.

Bodhisattva Sen. A gentle introduction to empirical process theory and applications. *Lecture Notes, Columbia University*, 2018.

Robert W Shafer, Soo-Yon Rhee, Deenan Pillay, Veronica Miller, Paul Sandstrom, Jonathan M Schapiro, Daniel R Kuritzkes, and Diane Bennett. Hiv-1 protease and reverse transcriptase mutations for drug resistance surveillance. *AIDS (London, England)*, 21(2): 215, 2007.

Chris Sherlock, Alexandre H Thiery, Gareth O Roberts, and Jeffrey S Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics*, 43(1):238–275, 2015.

Jorge Silva and Shrikanth Narayanan. Universal consistency of data-driven partitions for divergence estimation. In *2007 IEEE International Symposium on Information Theory*, pages 2021–2025. IEEE, 2007.

Jorge Silva and Shrikanth S Narayanan. Information divergence estimation based on data-dependent partitions. *Journal of Statistical Planning and Inference*, 140(11):3180–3198, 2010.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, and Marc Lanctot. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

Scott A Sisson, Yanan Fan, and Mark M Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

Scott A Sisson, Yanan Fan, and Mark A Beaumont. Overview of ABC. In *Handbook of Approximate Bayesian Computation*, pages 3–54. ChapmanBaye and Hall/CRC, 2018.

Helle Sørensen. Parametric inference for diffusion processes observed at discrete points in time: a survey. *International Statistical Review*, 72(3):337–354, 2004.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems*, pages 4134–4142, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Jan Stanczuk, Christian Etmann, Lisa Maria Kreusser, and Carola-Bibiane Schönlieb. Wasserstein GANs work because they fail (to approximate the Wasserstein distance). *arXiv preprint arXiv:2103.01678*, 2021.

Xiaochuan Sun, Tao Li, Qun Li, Yue Huang, and Yingqi Li. Deep belief echo-state network and its application to time series prediction. *Knowledge-Based Systems*, 130:17–29, 2017.

Yan Sun, Qifan Song, and Faming Liang. Consistent sparse deep learning: theory and computation. *Journal of the American Statistical Association*, pages 1–15, 2021.

Yitong Sun, Anna Gilbert, and Ambuj Tewari. On the approximation capabilities of ReLU neural networks and random ReLU features. *arXiv*, 2019.

Mikael Sunnaaker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate bayesian computation. *PLoS computational biology*, 9(1):e1002803, 2013.

M. Taddy, R. Gramacy, and N. Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106:409–123, 2011a.

Matthew A Taddy, Robert B Gramacy, and Nicholas G Polson. Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493):109–123, 2011b.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint arXiv:1804.06788*, 2018.

Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data. *Genetics*, 145(2):505–518, 1997.

Matus Telgarsky. A primal-dual convergence analysis of boosting. *Journal of Machine Learning Research*, 13(Mar):561–606, 2012.

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539, 2016.

Matus Telgarsky. Neural networks and rational functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3387–3393. JMLR. org, 2017.

George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2018. URL `https://CRAN.R-project.org/package=rpart`. R package version 4.1-13.

Terry M Therneau, Beth Atkinson, and Maintainer Brian Ripley. The rpart package, 2010.

Owen Thomas and Jukka Corander. Diagnosing model misspecification and performing generalized Bayes' updates via probabilistic classifiers. *arXiv preprint arXiv:1912.05810*, 2019.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2020a.

Owen Thomas, Henri Pesonen, and Jukka Corander. Generalised Bayes updates with f-divergences through probabilistic classifiers. *arXiv preprint arXiv:2007.04358*, 2020b.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31, 2022.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Networks Mach. Learn*, 2012.

Michalis K Titsias and Francisco Ruiz. Unbiased implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 167–176. PMLR, 2019.

Surya T Tokdar, Yu M Zhu, and Jayanta K Ghosh. Bayesian density regression with logistic Gaussian process and subspace projection. *Bayesian Analysis*, 5(2):319–344, 2010.

Dustin Tran, Rajesh Ranganath, and David Blei. Hierarchical implicit models and likelihood-free variational inference. *Advances in Neural Information Processing Systems*, 30, 2017.

Nghia Tran, M-N sand Nguyen, David Nott, and Robert Kohn. Bayesian deep net GLM and GLMM. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

Berwin A Turlach. Discussion on least angle regression. *The Annals of Statistics*, 32(2): 481–490, 2004.

Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-Hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR, 2019.

Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. In *International Conference on Learning Representation*, 2017.

Sara A van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

Sara A Van De Geer and Peter Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

Aad W van der Vaart and Jon A Wellner. *Weak convergence and empirical processes: with applications to statistics*. Springer, 1996.

Ramon Van Handel. Probability in high dimension. Technical report, PRINCETON UNIV NJ, 2014.

CJ van Rijsbergen. *Information retrieval*. Butterworth, 1979.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Hrishikesh D Vinod. A survey of ridge regression and related techniques for improvements over ordinary least squares. *The Review of Economics and Statistics*, pages 121–131, 1978.

Anatoliy Georgievich Vitushkin. A proof of the existence of analytic functions of several variables not representable by linear superpositions of continuously differentiable functions of fewer variables. In *Doklady Akademii Nauk*, volume 156, pages 1258–1261. Russian Academy of Sciences, 1964.

Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Hao Wang and Dit-Yan Yeung. Towards Bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3395–3408, 2016.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory*, 51(9):3064–3074, 2005.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, 2009.

Yuexi Wang and Veronika Ročková. Adversarial bayesian simulation. *arXiv preprint arXiv:2208.12113*, 2022.

Yuexi Wang and Veronika Ročková. Uncertainty quantification for sparse deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 298–308. PMLR, 2020.

Yuexi Wang, Tetsuya Kaji, and Veronika Ročková. Approximate Bayesian computation via classification. *Journal of Machine Learning Research*, 23(350):1–49, 2022a.

Yuexi Wang, Nicholas Polson, and Vadim O Sokolov. Data augmentation for Bayesian deep learning. *Bayesian Analysis*, 1(1):1–29, 2022b.

Gunter Weiss and Arndt von Haeseler. Inference of population history using a likelihood approach. *Genetics*, 149(3):1539–1546, 1998.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688, 2011.

Darren J Wilkinson. *Stochastic modelling for systems biology*. Chapman and Hall/CRC, 2018.

Richard David Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in Genetics and Molecular Biology*, 12(2):129–141, 2013.

Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.

Pei-Shien Wu and Ryan Martin. A comparison of learning rate selection methods in generalized Bayesian inference. *Bayesian Analysis*, 18(1):105–132, 2023.

Yun Yang and Debdeep Pati. Bayesian model selection consistency and oracle inequality with intractable marginal likelihood. *arXiv preprint arXiv:1701.00311*, 2017.

Yun Yang and Surya T Tokdar. Minimax-optimal nonparametric regression in high dimensions. *The Annals of Statistics*, 43(2):652–674, 2015.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *arXiv*, 2019.

Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. In *International Conference on Machine Learning*, pages 5660–5669. PMLR, 2018.

Peng Zhao and Bin Yu. On model selection consistency of LASSO. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

Puning Zhao and Lifeng Lai. Analysis of k nearest neighbor KL divergence estimation for continuous distributions. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2562–2567. IEEE, 2020.

Mingyuan Zhou, Lauren Hannah, David Dunson, and Lawrence Carin. Beta-negative binomial process and Poisson factor analysis. In *Artificial Intelligence and Statistics*, pages 1462–1471, 2012.

Xingyu Zhou, Yuling Jiao, Jin Liu, and Jian Huang. A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, pages 1–12, 2022.

R. Zhu. *Reinforcement Learning Trees*, 2018. URL `https://cran.r-project.org/package=RLT`. R package version 3.2.2.

Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.