THE UNIVERSITY OF CHICAGO

A TRIPTYCH IN COMPUTATION:
DEEP LEARNING FOR MOLECULAR MASS SPECTRA, SUM-OF-SQUARES OPTIMIZATION, AND DIFFUSION GENERATIVE PROCESSES

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

COMMITTEE ON COMPUTATIONAL AND APPLIED MATHEMATICS

BY
RICHARD LICHENG ZHU

CHICAGO, ILLINOIS
DECEMBER 2024

This thesis is dedicated to my family.

What a misfortune, although you are made
for fine and great works
this unjust fate of yours always
denies you encouragement and success;
that base customs should block you;
and pettiness and indifference.
And how terrible the day when you yield
(the day when you give up and yield),
and you leave on foot for Susa,
and you go to the monarch Artaxerxes
who favorably places you in his court,
and offers you satrapies and the like.
And you accept them with despair
these things that you do not want.
Your soul seeks other things, weeps for other things;
the praise of the public and the Sophists,
the hard-won and inestimable Well Done;
the Agora, the Theater, and the Laurels.
How can Artaxerxes give you these,
where will you find these in a satrapy;
and what life can you live without these.

—*Constantine P. Cavafy*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This work would not have been possible without the continuous support of many people.

I want to express my deepest gratitude to my advisors, Yuehaw Khoo and Eric Jonas, for their invaluable guidance and mentorship. Yuehaw, you are a fountain of ideas and a patient and kind mentor. You embody the spirit of shokunin, an ethos all too infrequently found. Eric, I am grateful for the time we spent together and for the many insights you shared with me. Thank you both for helping me dream bigger.

To my other committee members Mathias Oster, Mary Silber, and Lek-Heng Lim – and to the CAM faculty and staff, thank you for being a part of my story. Mathias, I am thankful for many fruitful conversations. I appreciate your patience with me, especially during our last-minute revisions and rebuttals. Mary, thank you for always being in my corner. Things may have gone very differently had it not been for your presence and advice. Jonathan Rodriguez and Zellencia Harris: You both have been instrumental in making CAM such a wonderful place to be, thank you.

Paul Kim, Sheila Edstrom, Jim Kollar, Gil Refael, Dave Stevenson, Rob Phillips, Nick Rubin, and Emmett McQuinn: you opened my eyes and taught me to see. I hope you see the lessons you taught me reflected in this work.

And finally, to my friends and family. Words cannot express how blessed I have been to have you by my side. Phillip, Hwanwoo, and the other CAM students: The community has flourished in recent years, thanks in no small part to your collective efforts to build it. Only those who have gone through a PhD can fully understand the experience, so we shall always share that special bond. Andrew, Anthony, Eric, and the rest of the Dcrew: To think that a chance meeting on a ranch would have led to all this. Our conversations and gatherings have been welcome nourishment in the darkest of times. Your collective erudition and *areté* are inspirations, and you really do make the world a better place. Nishaad, Zack, Kat, and Tom: thank you for putting up with me. Like Hamming, I am reminded that keeping one's door

open can be the difference between a productive but dull life and a less productive but much richer one.

Sean: I am inspired both by the depth of your thinking and your ability to down beers like a champ. Let there be peace on earth and good will to men; may your code always run on the hot path. Tina: These last few years have been some of the best of my life thanks to you. I'm confident the best is yet to come. Let this work be a testament to never giving up. Grace: When I started this journey you were still in high school, and now you're about to graduate. Your hard work and vision will take you far, and I look forward to seeing all that you may bring to this world. To my parents: it's been a long journey hasn't it? To think that when we arrived in Chicago all those years ago I would end up back here. So much of my ability to pursue this work is thanks to your unwavering support. Consider this a small tribute to the sacrifices you've made.

People pass in and out of one's life like the shuttle weaving threads in a tapestry. Some stay with you for a brief moment, others still for longer, each thread a meaningful part of the whole. I am glad to have shared this road with all of you. This degree is as much yours as it is mine.

ABSTRACT

Mathematics, as Eugene Wigner once noted, has no inherent reason to be as effective in the natural sciences as it is. Yet, those who seek to model the world have long used it to formulate powerful physical theories – from explaining the motions of planets to the dynamics of electricity and even the quantum-mechanical behavior of particles too small to observe. While many of the big questions have been answered, countless others remain. Why are some problems so easily solved, while others remain stubbornly intractable? What governs the dynamics of complex systems? How do we distinguish real regularities in data from phantom fluctuations? Today's solutions look very different from yesterday's as our reliance on data and predictive power continues to grow. In such a world, efficiency and simplicity matter more than ever.

This thesis presents three seemingly unrelated ideas. First, we present a data-driven approach to structured prediction of mass spectra. Mass spectrometry is commonly used in analytical chemistry as a means to characterize compounds, as it counts and weighs the fragments from the high-energy breakdown of molecules. By combining supervision from the substructures generated in the fragmentation of small molecules with graph neural networks, we achieve state-of-the-art performance in the prediction of electron-ionization mass spectra.

Next, we explore a semidefinite programming perspective on parametric polynomial optimization. We demonstrate how parameterized polynomial optimization can be lower bounded by the solution to an infinite-dimensional sum-of-squares optimization problem and we show how semidefinite programming can be used to approximate a solution. We prove the convergence of the resulting hierarchy (a variant of the Lasserre SOS hierarchy) and present some practical applications.

Finally, we do a deep dive into generative diffusion processes. We discuss the connections between generative diffusion processes and physics, closely examine the structure of score-matching, and illustrate ways to uncover the structure of a problem from sparsity priors.

Each of these topics illuminates a distinct facet of computational science – from the efficient use of structured data in deep learning to the power and challenges of semidefinite programming, and finally, returning to the continued inspiration that physics offers for modern modeling approaches.

May their synthesis be an ode to modern computational thinking and a tribute to the simple yet powerful ideas of the past, present, and future.

CHAPTER 1

INTRODUCTION

This thesis is a triptych in computation, exploring three ideas: deep learning for molecular mass spectra, sum-of-squares optimization, and the structure of diffusion models. The thread tying them all together is the effective use of modern computing power and well-designed and prosaic mathematical ideas. We are not interested in mathematics for the sake of mathematics, but rather in the service of understanding the world around us.

The 21st-century may be considered the century of data. We are drowning in data, collected from various sources, whether it be physical measurements and observations, Internet content, and the vast array of detritus generated automatically via our devices. Our machines emit $\sim$ 300 exabytes of data a day (300 billion gigabytes) while the new disk storage space is produced at a rate of $\sim$ 3 exabytes a day. Of course, not all the data generated is useful, which is why we only store $\sim 1\%$ of it. The useful data predominantly comes from measurement and metrology devices that observe something about the world, such as the high-quality cameras we now carry in our pockets or the mass spectrometers used in analytical chemistry, found in hospitals, laboratories, airports, and everywhere that detection of chemicals is required.

It is a wonder that mass spectrometry works so well. The basic idea resembles that of Japanese kintsugi pottery: a molecule is energized by some physical process, and in the relaxation process it fragments into pieces. By measuring the masses of these fragments and their relative abundances, we can "put back together" (infer) the structure of the original molecule.

In gas chromatography electron-ionization mass spectrometry (GC/EI-MS), the setting we analyze, an electron beam is accelerated by a 70 volt electric field. It transfers anywhere from 15-30 eV to molecules in the gaseous phase. This energy must dissipate in some form, and typically it acts to fragment the molecule into many pieces. A typical chemical bond has energy on the order of 3 eV, so quite extensive fragmentation often occurs. This is quite a bit

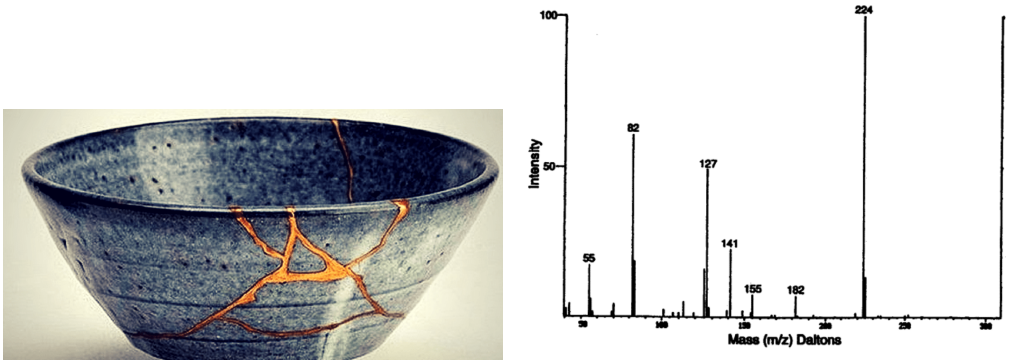

Figure 1.1: An example of Japanese kintsugi (left). An example of an observed mass spectrum (right, from [McLafferty and Turecek, 1994]).

of energy: for context, if we convert this to temperature via the equation $E = k_B T$ (with k_B the Boltzmann constant), the equivalent temperature is $T \sim 35000$ K! If the process is so energetically violent, one might ask whether the process is at all repeatable. Individual realizations of the process for a single molecule may not be, but when applied to a large number of molecules, the law of large numbers takes hold. Mass spectra tend to be very consistent from run-to-run, with over 98% dot product similarity [NIST, Zhu and Jonas, 2023]. Such high similarity suggests that there is fruitful structure to be modeled here!

Other types of mass spectrometry differ in the technical details of how molecules are accelerated and energized, but the ultimate outcome is a spectrum of masses with relative abundances. We can think of them as histograms of the masses of the fragments. The spectrometer actually observes the m/z ratio, the mass of the fragment divided by the charge of the fragment. For our purposes we can assume $z = 1$ and that the spectrometer is just measuring mass.

The NIST database presently contains $\sim 3 \cdot 10^5$ mass spectra that have been collected, and is growing only at the rate of $\sim 10\%$ a year [NIST]. New compounds are not measured very often, as most compounds of interest have already been measured. However, consider that just a tabulation of known natural products contains $\sim 2 \cdot 10^5$ molecular structures [Buckingham,

2023]. This is much smaller than the total possible of compounds that we are aware of and have tabulated in the PubChem database ($\sim 10^8$), which is itself much smaller than the total possible number of compounds that could exist, $\sim 10^{60}$. This disparity is generally true for most biological and chemical databases: it is fairly straightforward to enumerate possible chemical structures but much harder to characterize them via biological assays or analytical chemistry techniques. Choosing to focus only on the molecules that have been characterized and limiting ourselves to expanding the ones that have been characterized by our limited capacity is untenable. "Completing" the remaining measurements with synthetic predictions using accurate models is the only way to get a handle on the vast space of possible molecules.

This requires us to be more intelligent with how we use the data. In chemistry, this means taking advantage of the structural and physical chemistry knowledge involved in the process. We cannot just build a black-box model and hope that it works in the mass spectrometry context. It may perform well in-sample, but we can have no guarantees on how well it will generalize on larger molecules. Our approach involves learning a model to explicitly assign probabilities to the fragments of a molecule, with a complete enumeration of the fragments assumed. Previous work either used deep learning as an end-to-end model to predict the mass spectrum directly, or used a more structured chemical model that had limited capacity and training data. We shall see our approach in more detail in Chapter 2.

Next, we consider optimization. In general optimization problems, we are given some function $f(x)$ and we seek to find its minimum over some domain $x \in \mathcal{X}$. In particular, when the domain $\mathcal{X}$ and the function $f(x)$ have some particular structure, we call them convex problems and can solve them quite efficiently. Convex optimization problems are those where the function $f(x)$ is convex, i.e. for any pair $x_1, x_2 \in \mathcal{X}$ and any $\lambda \in [0, 1]$, the following holds:

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

and the domain $\mathcal{X}$ is a convex set. They are particularly special because *any locally optimal*

solution is also globally optimal. Because of this, they are very well understood, with efficient solvers readily available. Modern research has turned to non-convex optimization problems, where we instead pursue iterative methods that can converge quickly to a local minimum with few guarantees about the optimality of a solution. We are often satisfied with these solutions, particularly in deep learning settings, where the optimization landscape is so complex that finding a global minimum is often infeasible.

Still, there is a rich body of theory to mine in old ideas. Our next topic focuses on a case of parametric polynomial optimization, where we seek to minimize a polynomial $f(x, \omega)$ over the variables x, with parameters ω that can be varied. This is a harder problem than standard polynomial optimization, where we now seek to characterize the entire trajectory of minimizers $x^*(\omega) = \text{argmin}_x f(x, \omega)$.

For general polynomials this is a hard problem, but the theory of sum-of-squares polynomials suggests a possible solution. A polynomial is sum-of-squares if it can be written as a sum of squares of other polynomials. In particular, if it is sum-of-squares, we can guarantee its non-negativity. This was used to great effect in the Lasserre sum-of-squares hierarchy, which converts polynomial optimization $\min f(x)$ into the related non-negativity problem

$$\max_{c \in \mathbb{R}} c \quad \text{s.t.} \quad f(x) - c \geq 0$$

and then taking the sum-of-squares relaxation

$$\max_{c \in \mathbb{R}} c \quad \text{s.t.} \quad f(x) - c \text{ is sum-of-squares}$$

This last problem is tractable when we search for sum-of-squares polynomials of a fixed degree, and admits a semidefinite programming solution! Most importantly, the solutions we find can be proven to converge to the true minimum of the original polynomial as the degree of the sum-of-squares relaxation increases, and in practice we can often find the true

minimum with a low degree relaxation (finite convergence). In Chapter 3, we shall see how to generalize this approach to the parametric case.

Finally, we turn to the structure of diffusion models. Diffusion models are a class of generative models that have seen a resurgence in popularity since they were demonstrated to achieve state-of-the-art performance in image modeling. The theory of diffusion processes dates back to the time of Einstein, when he characterized the motion of colloids in suspension as a random walk driven by unobserved perturbations, now known as Brownian motion.

In diffusion generative modeling, one seeks to add noise of successively larger noise scales to progressively transition a data point $x \in \mathbb{R}^d$ to a sample $z \in \mathbb{R}^d$ that is indistinguishable from noise, typically chosen to be a standard Gaussian, e.g. through a time-varying stochastic process like

$$dx_t = -x_t dt + \sqrt{2} dW_t$$

where W_t is the standard Wiener process. The trajectory has an exact solution

$$x(t) = e^{-t} x(0) + \sqrt{1 - e^{-2t}} z$$

where $x(0)$ is the sample at time $t = 0$ and z is a sample from $N(0, I_d)$. Reversing this process would allow us to generate samples from the data distribution by only drawing samples z and "pushing" them back through the reverse process. It turns out that the reverse process is well-specified and requires the estimation of a Stein score function of the time-varying density of samples.

Diffusion generative modeling thus has deep roots in the theory of stochastic differential equations, Markov chain Monte Carlo, and numerical differential equations, among other fields. In Chapter 4 we shall take a closer look at some of the core assumptions of diffusion modeling, examining the structure of score functions in more detail.

CHAPTER 2

RAPID APPROXIMATE STRUCTURED PREDICTION OF ELECTRON-IONIZATION MASS SPECTRA

2.1 Introduction

Mass spectrometry is extremely useful in analytical chemistry as it provides valuable information on the makeup and structure of various chemicals. In this chapter, we discuss the application of data-driven deep learning methods to the prediction of the resulting mass spectra, which are an extremely noisy and difficult physical process to model from first principles. The data-driven deep learning approach detailed here adapts conventional deep learning methods to the structured setting of molecules by considering them as a graph. Several other innovations, including the fusion of standard deep learning with knowledge of the fragmentation process and generated fragments in high-energy electron-ionization mass spectrometry, are developed and detailed here. State-of-the-art performance is achieved on several large datasets of (molecule, spectrum) pairs. This chapter is adapted from the publication [Zhu and Jonas, 2023]. Additional background and supporting material external to the core ideas outlined in this chapter can be found in Chapter 5.

2.2 Background

Gas-chromatography electron-impact mass spectrometry (GC/EI-MS) ionizes a volatile substance via high-energy electron bombardment. The subsequent relaxation of the ionized substance from the high-energy state induces fragmentation, generating a shower of charged and neutral fragments. The charge-to-mass (m/z) ratio of the fragments are then measured in a spectrometer. It is reasonable to assume that fragments are singly-charged [Wei et al., 2019, Allen et al., 2016], so the measured m/z values can be interpreted directly as fragment

masses. Due to the cost-effectiveness and experimental reproducibility of GC/EI-MS, it is a mainstay of modern analytical chemistry workflows. The spectrum of a given compound is commonly used as a "fingerprint" used for matching against known database spectra. Additionally, it is often used as one of the first steps in structural characterization.

Currently, the NIST Mass Spectral Library[NIST] is the largest publicly-available database of EI-MS spectra, containing over 300,000 spectra for molecules containing ≤ 128 atoms. However, the space of possible molecules is incredibly large, and even annotated databases such as PubChem[Kim et al., 2020] have over 100 million known chemical structures. Less than 0.30% of the PubChem compounds have measured spectra. Clearly, experimental characterization at such scale is prohibitive. This is exacerbated by the fact that cheap products are easily attainable and measured many times, while many of the structures in PubChem come from the long-tail of rare, non-natural, or difficult to procure set of compounds. Such limitations require computational and statistical approaches to predicting mass spectra.

Computational approaches to the mass spectral prediction problem fall into two categories: first-principles physically-based simulation and data-driven statistical methods.

2.2.1 First-principles physical simulation.

Purely statistical theories. Ab initio approaches to EI-MS prediction leverage quasi-equilibrium theory (QET) or Rice-Ramsperger-Kassel-Marcus (RRKM) theories [McLafferty and Turecek, 1994] which explicitly model the redistribution of the energy over the internal degrees of freedom. By keeping only the relevant vibrational modes (with a harmonic oscillator approximation), the density of states (core to the estimation of the rate constants) may be approximated. Such theories and its expansions have been used to study the relative abundances of fragment ions in well-known spectra[Bauer and Grimme, 2016]. The need to enumerate the possible reaction pathways limits the successful application of such theories to very small molecules.

Born-Oppenheimer Molecular dynamics. Methods such as QCEIMS and its derivatives combine quantum-mechanical Born-Oppenheimer molecular dynamics (MD) with fragmentation pathways to compute fragment ions within picosecond reaction times and femtosecond intervals for the MD trajectories. Statistical sampling of these trajectories then provides a distribution of observed fragments, generating a spectrum. However, even with the approximations made to reduce runtime, the runtime complexity is prohibitive for scaling, on the order of $O(100$ hours) for small molecules less than 100 Da in mass [Koopman and Grimme, 2019]. While these methods can often qualitatively identify plausible fragmentation pathways, their accuracy is not yet high enough for compound identification [Wang et al., 2020].

2.2.2 Data-driven statistical methods.

Computational systems for predicting mass spectra fragmentation was a topic of interest for early AI researchers, leading to projects like DENDRAL [Lindsay et al., 1980] in the 1960s, which applied rules-based heuristics programming to the structural elucidation in organic chemistry. The heuristics used in the project have been improved upon over the last few decades, as chemists continue to add to a library of known fragmentation processes [McLafferty and Turecek, 1994], by which chemical bonds and atoms are broken and rearranged. These heuristics are used by chemists to manually identify and explain the occurrence of particular peaks in small molecule EI-MS spectra [De Vijlder et al., 2018].

Early approaches were rules-based approaches, iteratively applying thousands of known rules to combinatorially enumerate possible fragments. Such methods have very high recall, providing a possible explanation for every peak in a spectrum. Recent work fuses the high recall of the combinatorial approach with learned models to improve precision. In particular, the series of CFM-ID papers [Allen et al., 2014, 2015, 2016, Djoumbou-Feunang et al., 2019] achieved state-of-the-art results in using a general rules-based fragmentation scheme to generate a large fragmentation tree for each molecule, and then learns the parameters for a

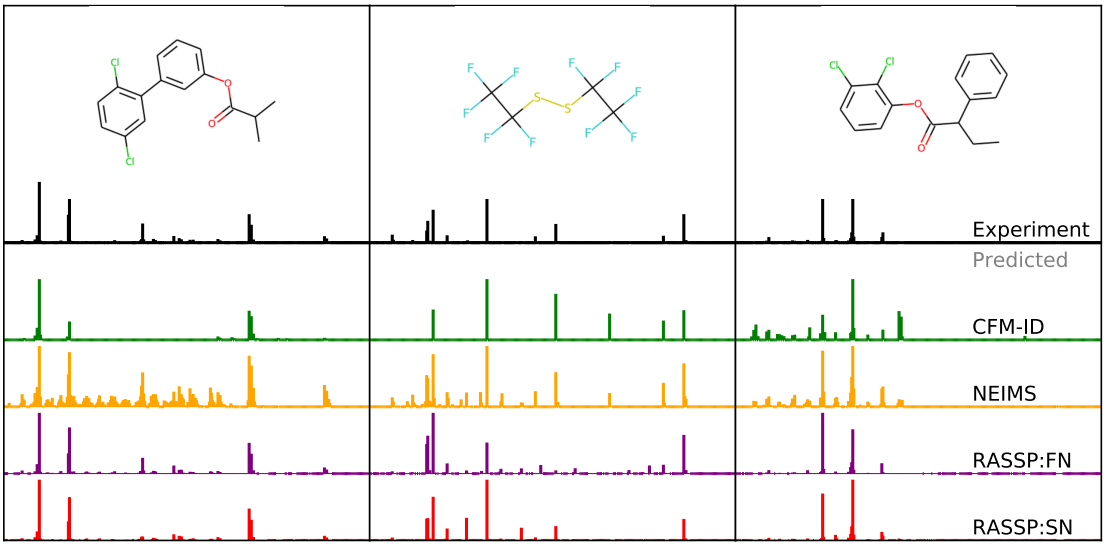

Figure 2.1: Example predictions on the held-out NIST2017 test set from the models we assess in this work: FormulaNet (RASSP:FN), SubsetNet (RASSP:SN), NEIMS[Wei et al., 2019] , CFM-ID[Allen et al., 2016] , as well as experimentally-measured spectra [NIST] .

model that parametrizes a Markov transition process over the tree.

The advent of machine learning and graph neural networks has renewed the interest in this problem. Recent work [Wei et al., 2019, Zhu et al., 2020] innovates in this area by using deep neural networks that directly predict spectra from molecular fingerprints or molecular graphs. These systems have been shown to do quite well on learning the regularities present in EI-MS data, achieving performance surpassing that of simpler linear or neural network models.

2.3 Methods

The complete calculation of the full fragmentation tree for a given molecule undergoing EI-MS would contain all necessary information to accurately predict the observed spectrum: simply compute the isotopic m/z distribution for each observed fragment, and sum these over all fragments weighting by the fragment probability. However, the physical complexities and possible fragmentation paths make this a very challenging, and perhaps impossible,

computational task. Approaches like CFM-ID [Allen et al., 2014] attempt to model this process, but the exponential growth in possible fragmentations naturally limits the types of fragmentation events and fragmentation tree depth, impacting spectral prediction accuracy.

We instead reason backwards from our observable: the spectrum. While one could attempt to directly predict the spectrum given an input molecular structure or molecular fingerprint (like NEIMS), this discards effectively all physical intuition about the problem. As we state later, we are interested in developing methods that will naturally extend to higher-resolution spectra, and contemporary machine learning methods can struggle with extremely high-dimensional output spaces. Fig. 2.2 illustrates the possible representation levels at which one can reason about the problem, starting from the input molecule structure (viewed as a graph) and ending with the mass peak distribution as viewed in the spectrometer.

Note that for any fragment child ion of the original molecule, both the chemical subformula and the vertex (atom) subsets allow us to exactly determine the observed peak m/z distribution of the fragment. However, there are far fewer formulae than atom subsets, and far fewer atom subsets than possible subgraphs. For example, $C_6H_{12}O_6$ has a total of 18 bonds. If we consider complete bond breakages out to depth d, we can generate $18!/(18-d)!$ unique bond breaks and up to the same amount of possible subgraphs but only $7 \times 13 \times 7 = 637$ possible subformulae. For $d \geq 3$, the number of possible subgraphs is already larger than the number of possible subformulae. Thus, we focus only on chemical formulae and atom subsets. Motivated by the need to generalize to higher-resolution spectra, we adopt two different physically-informed substructure enumeration methods, one that produces possible fragment formulae (used in RASSP:FN), and another that produces possible fragment vertex subsets (used in RASSP:SN).

Generating subformulae. Generating subformulae for a given molecule is straightforward. For a given molecule, we can iteratively generate all subformulae by recursively taking the set-wise Cartesian product of the possible subformulae for a single element of the molecule with

the subformulae over the rest of the molecule. For example, getSubformulae($C_6H_{12}O_6$) = getSubformulae(C_6) $\otimes$ getSubformulae($H_{12}O_6$) . The base case is a single element X occurring N times, where the possible subformulae are simply the possible occurrences of X: getSubformulae(X_N) = $[X_0, X_1, \ldots, X_N]$. However, only considering the chemical formula (which elements are present and how many) discards vital structural information such as bond connectivity. In doing so, we ignore all information about which formulae might appear more often in the final spectrum than others.

We thus explore an additional, richer representation of fragments: vertex (atom) subsets. We use atom subsets and vertex subsets interchangeably to refer a subset of the atoms present in a molecule. Atom subsets are preferred to complete fragment subgraphs because considering bond connectivity explodes the number of subgraph objects we must consider. Note that two fragment subgraphs with different bonds may still implicate the same subset of atoms from the original molecule.

Unfortunately, for most interesting molecules it is quite infeasible to enumerate all possible subsets, as a molecule with N atoms can have 2^N possible atomic subsets. Conveniently for us, this space of atomic subsets is highly redundant, with many atomic subsets having similar mass peaks in a spectrum. Thus, we cannot proceed like we did with the chemical subformulae earlier, where we could simply enumerate all possible subformulae. For atom subsets, we need to devise a scheme that can generate sufficiently plausible subsets. It should have enough generality to output all peaks in a spectrum, but not so many as to be computationally intractable to fit a model to later on.

Generating subsets. In order to select plausible subsets from this much larger space of possible subsets, we adopt a heuristic bond-breaking approach where we begin with an initial molecule and recursively break all possible bonds out to a particular depth. In this work, we consider all fragments generated by breaking bonds out to $d = 3$. Discussion on why $d = 3$ is selected is presented later in Section: Evaluating the impact of the subset

enumeration. In order to improve the recall of this process, we also perform exhaustive hydrogen rearrangements, a well-studied transition in mass spectrometric fragmentations [McLafferty and Turecek, 1994]. Since our fragment generation process features bond removal and addition, it is possible to generate subgraphs that are not subisomorphic to the original molecule graph. However, our process notably misses important fragmentation processes. Consider the fragmentation process for toluene. Toluene (C_7H_8) starts with a 6-carbon ring ion and one of the possible pathways leads to an intermediate 7-carbon ring ion. Such a graph structure is not isomorphic to the original graph, and must be formed by the bonds breaking and rearranging to form a new ring ion [McLafferty and Turecek, 1994]. However, this fragment is not explicitly generated by our subset enumeration process, though the chemical formulae may still be output by our exhaustive formulae enumeration.

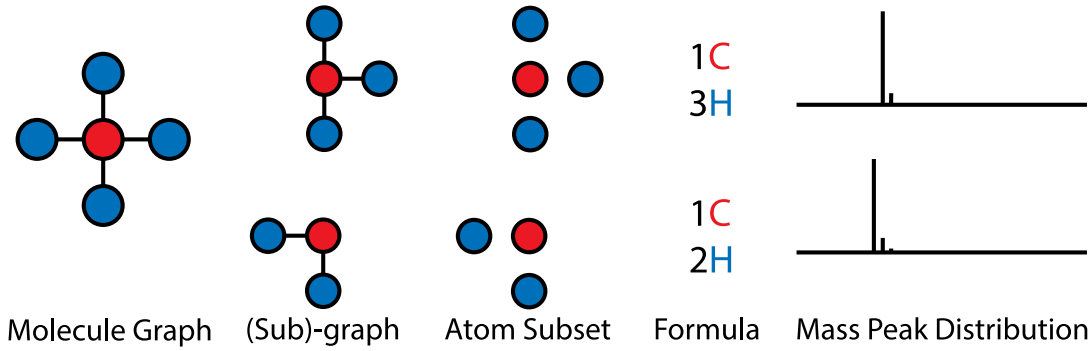

Figure 2.2: Different representation levels for the mass spectrometric forward problem. Each molecule is represented as a graph where nodes are atoms and edges are bonds. Subgraphs are connected components of the original graph, where both atom/bond presence in the subgraph are considered. Atom subsets are another level of abstraction where only atom presence in the set is considered. Formulae are yet another level, where only the counts of unique elements are considered. Finally, each unique formulae corresponds to a known mass peak distribution.

2.3.1 Graph neural networks for predicting a probability distribution over atom subsets and chemical subformulae

Our different enumeration approaches map to potential fragments, represented as either atom subsets or chemical formulae. For basic molecule identification, this often suffices – molecules of radically different structures will have fragments with non-overlapping peak distributions. However, as molecules get larger and more complex, significant overlap between their spectra can occur, even for molecules without significant structural similarities. Since more information about structure is captured in relative peak intensities, we would like to increase the precision of our barcode spectra and identify likely fragments. To do so, we employ graph neural networks (GNNs) as function approximators to learn a feature embedding for every atom in a molecule [Gilmer et al., 2017, Zhou et al., 2019, Waikhom and Patgiri, 2021, Guan et al., 2021, Zhu et al., 2020, Sanchez-Lengeling et al., 2021]. Rather than use GNNs directly to learn a molecule embedding or fingerprint that we map to a spectrum, we instead use it indirectly to produce per-atom features.

The feature embedding stored at each atom represents local about the atom's neighborhood and global information about the molecule. The chemical subformulae contains information about which elements are present in a fragment, and how many. Similarly, an atom subset contains more specific information about the atoms that are present in a fragment. The core idea is to learn to combine these two sets of information, to produce a probability distribution over chemical formulae (*FormulaNet*) or atom subsets (*SubsetNet*). Once we have a probability distribution over atom subsets (chemical subformulae), we can directly evaluate what the predicted spectrum would be.

Per-atom feature embedding via graph neural network.

For a molecule graph $M = (V, E)$ with N_A atoms, we derive F_0 features for each atom (see supplementary materials for an exact description of features and network architecture),

giving a feature matrix X_0 of shape $N_A \times F_0$. The bonds between atoms are represented as a symmetric adjacency matrix $A \in \{0, 1, 1.5, 2, 3\}^{N_A \times N_A}$, where different bond orders are represented by different values. Together, we feed the per-atom feature matrix X_0 and adjacency matrix A into a multi-layer message-passing graph neural network (GNN) that outputs a per-atom embedding $X_d \in \mathbb{R}^{N_A \times F_d}$ (Fig. 2.3).

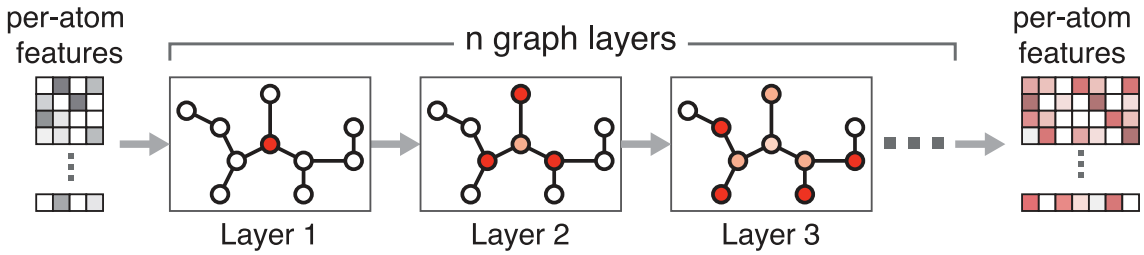

Figure 2.3: Message-passing graph neural network (GNN). We start off with a vector of features for each atom as our input features for the graph. Each successive layer of the GNN performs an update of each atom's embedding based on a nonlinear transform of the embeddings of the atoms adjacent to it (hence "message-passing"). After n iterations, we generate a new set of embeddings for each atom.

FormulaNet.

The per-atom features X_d can be combined with the atom subset/subformula information in a few ways. The first model we discuss uses only the set of all chemical formulae that arise from a molecule's fragmentation. Note that the chemical formula enumeration process is simple yet fully exhaustive, combinatorially capturing all possible formula that could arise, even the ones inaccessible via a physically-based fragmentation process.

Our universe of elements is $E = \{H, C, O, N, S, P, Cl\}$. These elements were chosen to ensure nearly full coverage of molecules from PubChem and NIST. Each chemical formula is then an array of non-zero integers $F \in \mathbb{Z}_+^{|E|}$. If a molecule generates $f(M)$ total chemical subformula, then the count-encoded representation our model takes is of form $F_c \in \mathbb{Z}_+^{f(M) \times |E|}$. Within the model, the count-encoded representation is converted into a run-length one-hot encoding of form $F_r \in \mathbb{Z}_+^{f(M) \times \text{maxelem}(E)}$, where maxelem is sufficiently large to contain all

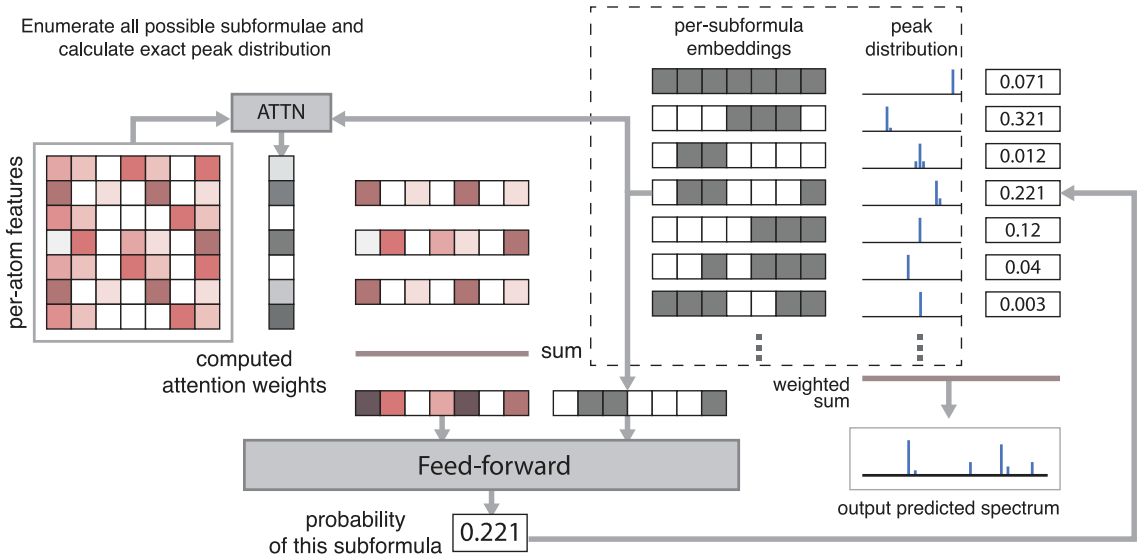

Figure 2.4: FormulaNet. We compute per-atom feature embeddings using a graph neural network (GNN). We then compute an attention weight for each atom's embeddings using the attention mechanism described in the text, and use that to perform a weighted sum of those features to produce a subformula-dependent graph embedding. We combine this with the representation of the subformula and (after several feedforward layers) derive a probability that that subformula contributes to the final spectrum.

chemical formula within the dataset. As an example, the formula CH_3 may be encoded as $[1, 1, 1, 0, 0, 1, 0, 0, 0, 0]$ where the first 5 entries correspond to 5 maximum possible H atoms and the last 5 entries correspond to 5 maximum possible C atoms. The Supplement contains exact details on how this is done.

We then compute an attention operation using the formula embeddings F_c as key and the per-atom features X_d as query and value. We then concatenate the result with the formula embeddings: $[\text{attention}(F_c, X_d, X_d), F_c]$ and pass this through a MLP to get unnormalized scores S for each formula. The unnormalized scores are converted to formula probabilities p using a softmax and scaled against weights computed via a linear layer from the per-atom features X_d (Fig. 2.4).

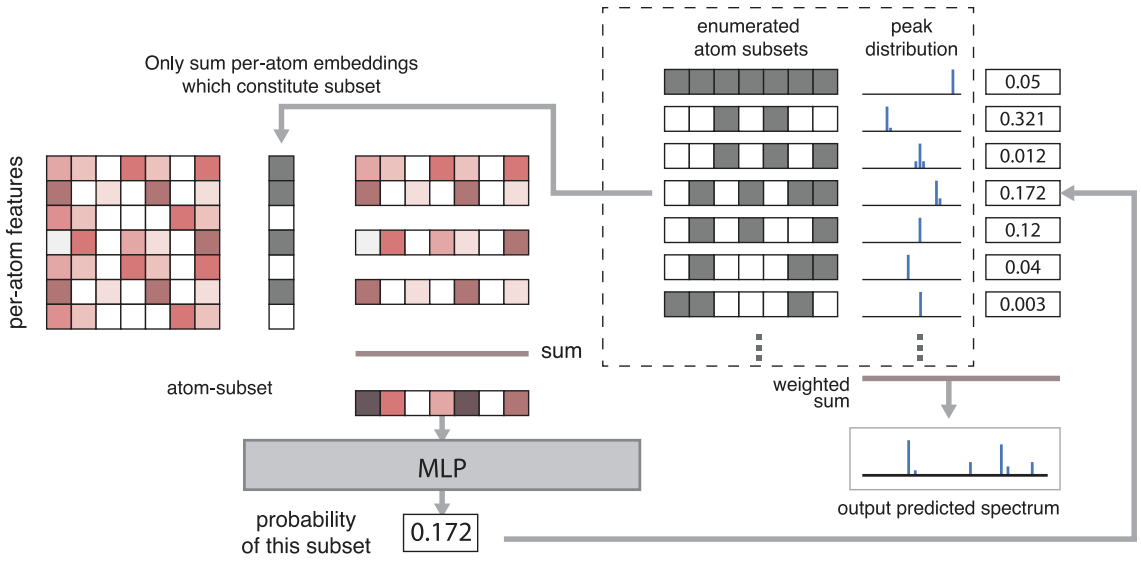

Figure 2.5: SubsetNet. Like FormulaNet, we use the GNN to generate per-atom feature embeddings. Separately, we generate candidate atom subsets via direct substructure enumeration (bond breaking and rearranging). The per-atom feature embeddings are combined using the atom subsets as "masks" to sum only the embeddings for the atoms present in each subset, generating an embedding for each atom subset. These subset embeddings are then fed into a MLP to generate probabilities for each subset.

SubsetNet.

The direct fragmentation process generates a set of atom subsets. For a molecule $M = (V, E)$ with N_A atoms and N_S unique atom subsets, the subset indicator matrix is a binary matrix of $\{0, 1\}^{N_S \times N_A}$. We generate an embedding for each subset by taking the mean of the per-atom embeddings X_d for only the atoms present in each subset. The subset embeddings X_{d+1} and the run-length N-hot encoding of the formula for each subset F_r are combined and then fed into a MLP to generate probabilities for each subset (Fig. 2.5).

2.3.2 Observation model

Both RASSP:FN and RASSP:SN generate probability distributions, the first over unique chemical formulae and the second over atom subsets of the original molecule. Given a formula we can exactly calculate the observed spectrum, taking into account isotopic variability

at natural abundance and mass defect. At integer-Dalton resolution, summing the atomic masses and rounding is sufficient, but using this exact spectral distribution will prove useful for later high-resolution experiments.

We then weight each formula/subset's mass spectrum according to the model's output probability, and sum all the observed mass spectra together to obtain one final mass spectrum prediction for the entire molecule.

2.3.3 Learning model parameters from data

Note that for both models, the input consists of the molecule graph and either (1) a set of possible chemical subformula of the molecule or (2) a set of possible atom subsets of the molecule. The output is a probability distribution over the subformulae or atom subsets. Because the exact mass peak distribution is known for each subformulae and subset (Section 2.3.2), we then exactly compute the mass spectrum at arbitrary resolution. We fit each model using stochastic gradient descent against minibatches of experimentally-observed (molecule, spectra) pairs to minimize the L2 error between scaled spectra, where the spectral intensities are scaled by a power. Powers < 1 reduce the importance of outlier peaks, whereas powers > 1 emphasize the importance of outlier peaks.

Metrics.

Each spectrum is represented as a set of charge-to-mass ratios, intensity tuples (m_k, I_k). We assume that all measured ions have charge one, and as such the charge-to-mass ratios may be interpreted directly as masses. Nearly all EI-MS data is obtained at integer-Dalton resolution, i.e. $(1.0, I_1), (2.0, I_2), \ldots$. For peaks that do not conform to this specification, such as output peaks from CFM-ID [Allen et al., 2016] that specify the exact fragment mass, we transform spectra from a set of discrete peaks to a histogram by binning at integer-Dalton resolution, with bins centered on integer values with unit widths and summing all the intensities for

peaks falling within the same bin. After binning the spectrum, we normalize it to have unit L2 norm.

The key metric for forward model performance is the weighted dot product (Eq. 2.1). The weighted dot product scales each mass by a mass power and each intensity by an intensity power. Note that due to the normalization factors on the bottom, this metric is actually weighted cosine similarity and not a proper dot product. Due to the normalization, the values of weighted dot product (for any a, b) fall in the range $[0, 1]$.

$$\mathrm{DP}_{a,b}(S_p, S_r) = \frac{\sum_k m_k^a I_{pk}^b \cdot m_k^a I_{rk}^b}{|| \sum_k (m_k^a I_{pk}^b)^2 || || \sum_k (m_k^a I_{rk}^b)^2 ||} \tag{2.1}$$

Some common values include $(a, b) = (1, 0.5)$ (regular dot product, DP) and $(a, b) = (3, 0.6)$ (Stein dot product, SDP) [Stein and Scott, 1994]. $a \geq 1$ increases the weight placed on errors at large masses, and $b < 1$ reduces the impact of outlier intensity values. SDP is commonly used in the literature to search and match spectra against spectral databases [Stein and Scott, 1994].

Beyond dot product (DP) and Stein dot product (SDP), we also track intensity-weighted barcode precision (WP) and intensity-weighted false positive rate (WFPR). These additional metrics respectively represent how much of the predicted spectral intensity was in bins also seen in the true spectrum and how much of the predicted spectral intensity was in bins *not seen* in the true spectrum. For barcode precision, a bin was considered only if the L1-normalized intensity surpassed some cutoff $i_{\min}$. In this work, we use $i_{\min} = 0.0001$. Top-K precision is also a relevant metric (how many of the top-K peaks in the predicted spectrum are also in the true spectrum). This and further metrics may be found in the Supplement.

Datasets.

The primary dataset used for training both SubsetNet and FormulaNet models was the NIST 2017 Main Library [NIST]. After filtering the dataset down to molecules containing only HCONFSPCl atoms, with total atoms ≤ 48, number of unique fragment formulae ≤ 4096 we obtained a dataset of 125643 molecules. Each molecule was divided into 10 mutually-exclusive dataset folds according to the last digit of the CRC32 checksum of the hashed Morgan fingerprint for the molecule. This procedure groups identical molecules in the same dataset fold, acting as an automatic check against repeated rows or molecules in the dataset. We used the first 8 folds for training (2 through 9, 100438 molecules, `nist-train`) and the last 2 folds for validation (0 and 1, 25205 molecules, `nist-test`).

To compare effectively with CFM-ID [Allen et al., 2016] which provides spectra for evaluation on a small subset of the NIST 2014 Spectral Library, we generate the `smallmols-orig` dataset from their provided molecule list [Allen et al., 2016]. In addition, we pulled molecules from the PubChem Substance Database [Kim et al., 2020]. `smallmols-orig` was filtered in the same way as the `nist17-mainlib` (HCONFSPCl atoms, ≤ 48 atoms, ≤ 4096 unique fragment formula), and used for evaluation against publicly-available parameters for the CFM-ID model [Allen et al., 2016] and the NEIMS model [Wei et al., 2019]. More information on datasets used is available in the Supplementary Information.

The final model with highest SDP and recall at 10 was a FormulaNet (see supplemental information for exact model parameters). The trained model generalizes to molecules of arbitrary size and fragments, so we evaluated it against the 73.2M PubChem molecules with HCONFSPCl atoms, ≤ 64 atoms, ≤ 32768 max unique fragment formula, and ≤ 49152 max vertex subsets. All the molecules and spectra are indexed and publicly available at our website `spectroscopy.ai`.

The NIST Replicate dataset consists of 63741 total "replicate" experimental measurements of 23200 unique molecules. None of these molecules appear in the NIST Main Library. Each

molecule was replicated a minimum of 2 times, with a mean of 2.7 replicates, a median of 2, and a maximum of 24 replicates. This dataset allows us to measure the variability of the experimental process due to stochasticity and inconsistent apparatuses. We use the replicate dataset to estimate the run-to-run variability between measured spectra contributed by varying apparatuses and protocols around the world. This experimental noise provides an upper bound on forward model performance.

2.4 Results

2.4.1 EI-MS forward prediction

Example spectral predictions are presented in Fig. 2.1, and forward prediction metrics are presented in Fig. 2.6. SubsetNet (RASSP:SN) and FormulaNet (RASSP:FN) were trained for 40 full epochs against a subset of the NIST 2017 EI-MS Spectral Library after selecting for molecules with ≤ 48 atoms, ≤ 4096 max unique subformulae, and ≤ 12288 subsets (100,438 molecules from `nist17-train`). A subset of molecules was held-out and used as a validation set for tuning hyperparameters and model architectures (`nist17-test`). Where relevant, RASSP:SN and RASSP:FN refer to the models of each architecture with best performance on this validation set. Where available, performance was also compared against the CFM-ID and NEIMS forward models [Allen et al., 2016, Wei et al., 2019]. NEIMS was trained from scratch for 100 epochs on `nist17-train`. CFM-ID spectra for the `smallmols` subset was derived from the supplementary data provided by the authors. Full model details, the training process, as well as code is available in the Supplementary Information.

As we can see in Fig. 2.6(a), our models show significant improvement in performance over previous physics-based models (CFM-ID), achieving a 95% SDP (out of 100%, actual values are bounded in $[0, 1]$) on `smallmols` compared to the CFM-ID 68%. FN and SN outperform NEIMS significantly on both the `smallmols` dataset and the `nist17` datasets.

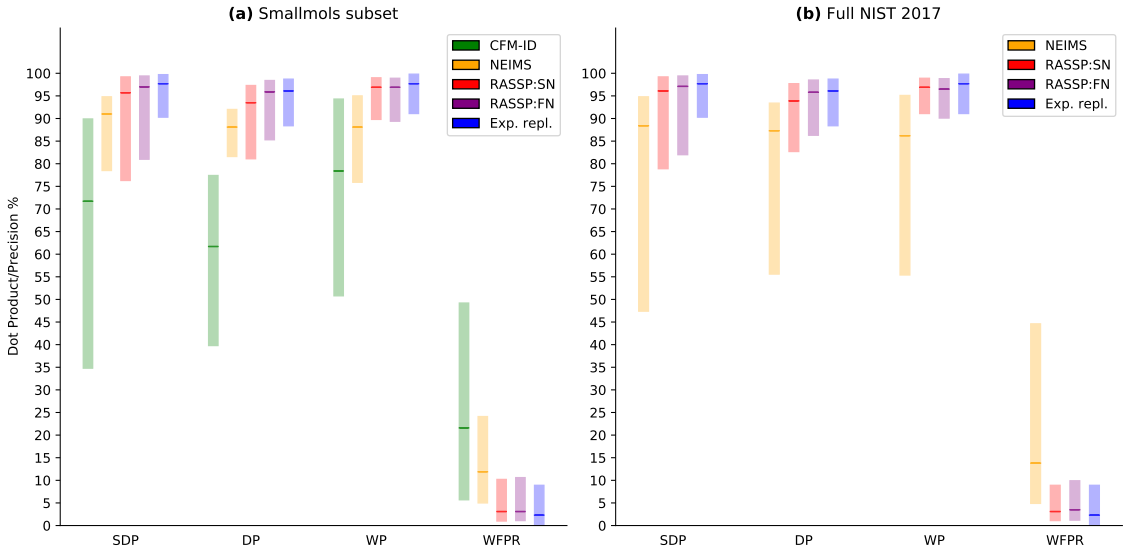

Figure 2.6: EI-MS prediction performance – The bottom and top of the bars represent the 10th and the 90th percentiles, with the middle bold tick representing the median (all percentiles evaluated over the dataset specified). (a) Performance of CFM-ID, NEIMS, SubsetNet, and FormulaNet models on molecules from `smallmols-orig` (a subset of NIST EI-MS data selected in a previous paper[Allen et al., 2016]). (b) Performance of NEIMS, SubsetNet, and FormulaNet models on `nist17-mainlib`. Metrics are: Stein dot product (SDP, weighted dot product with $(a, b) = (3, 0.6)$), regular dot product (DP, $(1, 0.5)$), intensity-weighted precision (WP), and intensity-weighted false positive rate (WFPR). "Exp. repl." refers to experimental replicate variability, estimated by taking the mean metrics over all replicate experiments in `nist17-replib`, and are shown in both (a) and (b) for comparison purposes. They can be viewed as a proxy for experimental variability and as such an "upper limit" to the forward prediction accuracy.

We leverage the `nist17` replicate experiments to compute the best possible intra-experimental performance (labeled “Exp. repl”) Note that our prediction performance approaches this experimental accuracy, as depicted in Fig. 2.6(b).This gives us a sense of the run-to-run and apparatus-to-apparatus variability in the EI-MS process, providing an upper-bound on forward model performance.

The actual distribution of DP values is depicted in Fig. 2.7. As we can see, the distributions for both SN and FN skew much closer to that of experimental variability than NEIMS. There remains some room for improvement, especially with SN. This indicates how much headroom there might be left to improve upon by improving forward model predictive performance.

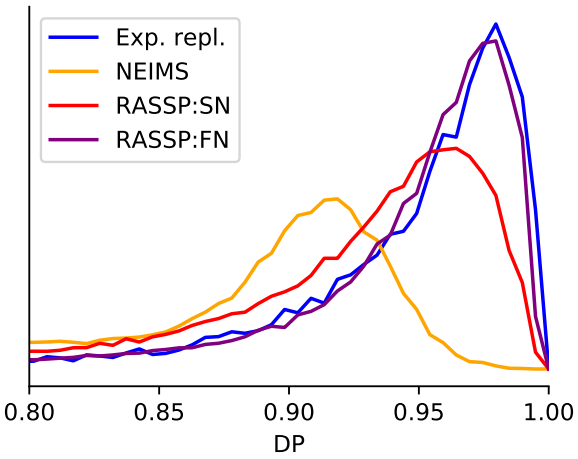

Figure 2.7: Histogram (probability density function) of prediction dot products $DP_{1,0.5}$. Here we show the distribution of dot products for all predictions on the NIST Mainlib from the 3 models NEIMS, SubsetNet, and FormulaNet as compared to the distribution of dot products for replicate experiments from NIST Replib (labeled "Exp. repl."). As forward models improve their accuracy, the distribution should shift to the right. The NIST Replib distribution represents the current limit of prediction performance, accounting for intrinsic experimental variability as well as differences in experimental setups.

2.4.2 Library matching

Another validation of the accuracy of our predicted spectra is to use them in a database lookup (library matching) task resembling the common comparison of experimental spectra against spectral databases to identify unknown compounds. We follow the procedure detailed in the NEIMS paper [Wei et al., 2019]: we evaluate the performance of an EI-MS forward model by using model-inferred spectra to replace a set of molecule, spectra pairs in a spectral database, and then comparing known experimental "replicate" (molecule, spectra) pairs to the database to see whether the true molecule is ranked highly.

We use the NIST 2017 Main and Replicate Libraries (`nist17-mainlib` and `nist17-replib` respectively) for this task. The Replicate library consists of replicated experimental measurements, and has no overlap with the Main library. To evaluate a given model's library matching performance, we evaluate it against all molecules in the Replicate library. These spectra are then added to the Main library to form an augmented library that consists of

mainlib experimental spectra and replicate model-inferred spectra. We use the replicate library as a query library, randomly selecting a replicate experimental spectrum for each molecule. Each mol, spectrum row in the query library is then tested against the augmented library. The max peak in the query spectrum is used to filter the augmented library molecules to ±5Da, and then the rows from the augmented library are sorted by decreasing SDP vs the query spectrum. The rank of the matching spectrum is recorded. Some examples of the library matching task are illustrated in Fig. 2.9.

As seen in Fig. 2.8, both SN and FN outperform NEIMS in the library matching (database lookup) task they originally detailed [Wei et al., 2019]. The error rate at 1 for NIST, at 16.9%, indicates that doing a simple database lookup and taking the top matching molecule gets the wrong match 1 out of every 6 spectra. We improve the error rate at 1 from 1 in 2 spectra (47.2%, NEIMS) to 1 in 4.6 spectra (21.4%, FN). The numbers improve rapidly as the window increases, with the error rate at 10 declining to 1 in 83.3 molecules (1.2%, NIST Ref). FN improves on NEIMS by nearly 3× in this library matching task. Moreover, we note that SN and FN were trained to maximize forward metric performance (SDP), not recall at 10.

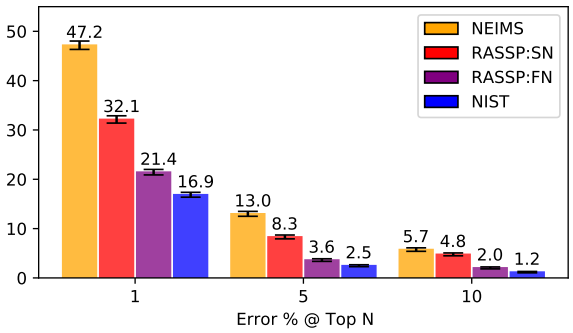

Figure 2.8: Library matching performance. Comparison of error rate on the library matching task in [Wei et al., 2019] over the top 1, 5, and 10 ranked spectra achieved by different model architectures. All graphics display the performance of using NIST replicate spectra as query spectra, indicating the lower bound of error rate given present EI-MS experimental accuracy. Error bars correspond to 1-σ variation when estimating the error rate using bootstraps, drawing 20% of the query library randomly without replacement.

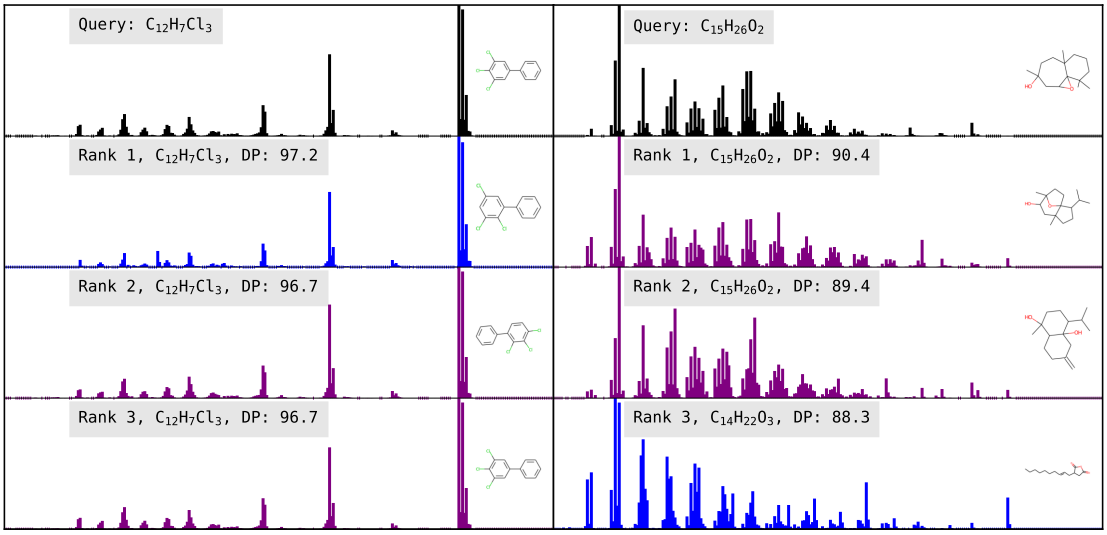

Figure 2.9: Library matching task. The left and right panels demonstrate two examples of the library matching task. The query spectrum (experimental spectrum from the NIST Replib) is displayed at top in black, and the top 3 ranked spectra from the augmented database (comprised of NIST Mainlib experimental spectra and model-predicted spectra on the NIST Replib) are shown, along with their chemical formulae and the similarity metric (dot product with $(1, 0.5)$.) Blue spectra are experimental spectra from NIST Mainlib and purple spectra are the predicted spectra from the model used in the task. In this figure, predicted spectra are output from the best FormulaNet (FN) model. On the left, we see that the correct match is the spectrum at rank 3. Two molecules with exact formula matches but slightly different structures (hydrogen placements) are ranked higher. On the right, the correct match is ranked outside the top 3, but we can see that two molecules with matching formulae but slightly different structures are ranked at the top.

2.4.3 Higher-resolution data

Nearly all computational prediction and database lookups utilize EI-MS spectra measured at integer-Dalton resolution. Our results detailed here are similar. To test whether either of these models generalize to higher-resolution data, we trained both SN and FN against a high-resolution synthetic dataset generated by using the CFM-ID [Allen et al., 2016] provided weights to predict spectra (and their exact peaks) for molecules from PubChem. Rather than binning at the 1Da resolution, we binned at 0.10 Da resolution. We randomly selected 1000, 10000, and 100000 molecules to use as training, and held out 10000 molecules to use as test. The generalization performance of SN and FN is depicted in Fig. 2.10. We see that the performance of SN and FN converge as the dataset size (and molecular diversity) increases, but SN generalizes much better at low-dataset size. Due to the limited availability and expense of collecting high-resolution EI-MS data, this indicates that SN may generalize far better in the low-dataset regime than FN, indicating that the atom subset representation generated by substructure enumeration may be a more natural representation of the mass spectral problem than simply enumerating the formulae. For full details about the generation of the high-resolution synthetic dataset, see the Supplementary Information.

2.4.4 Dependence on molecular similarity

Ultimately we are interested in our model's performance on new, unseen structures. Machine-learning methods learn to recognize patterns in their training data, and thus care is taken to separate out train and test datasets. Fitting of our model is performed exclusively on molecules in our identified training set, with test molecules reserved solely for metrics evaluation. In computational spectral prediction, training and evaluating a spectral prediction model on molecules of a particular class or structural motif can lead to erroneous evaluation of its performance.

To further investigate how our model may generalize to previously-unseen structures,

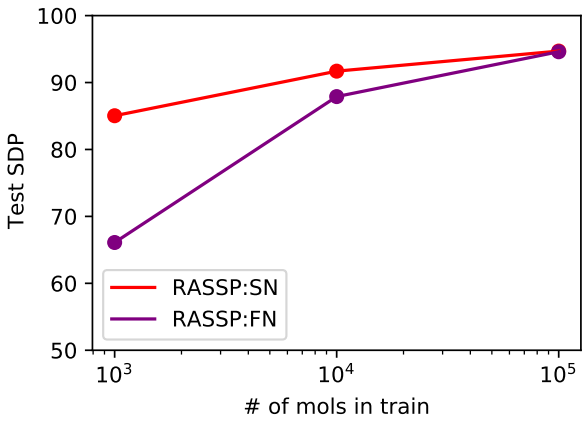

Figure 2.10: Performance of SubsetNet and FormulaNet with scaling dataset size. As we increase the size of the high-resolution training dataset (synthesized using CFM-ID [Allen et al., 2014, 2015, 2016] for molecules from PubChem), we see that SN and FN both converge to similar performance. However, their performance diverges dramatically when the dataset is small.

we examine how our prediction on molecules in the test set changes depending on how structurally those test molecules were to molecules in our training set. Such analysis is key in determining whether a model truly generalizes to structures it has never seen before, and may provide further confidence in using its spectral predictions on molecules that have no observed spectrum.

Forward spectral prediction performance

In Fig. 2.11, we present the SDP vs similarity to the closest molecule in the training set for all the molecules in our test set. We see a clear dependence on similarity – the higher the similarity to the training set, the better the performance. This effect is most pronounced at low similarity levels, where the SDP for the 10% similarity quantile falls to below 20%. Note that 90% of test set molecules have a similarity to the training set over 69.0% (vertical red line).

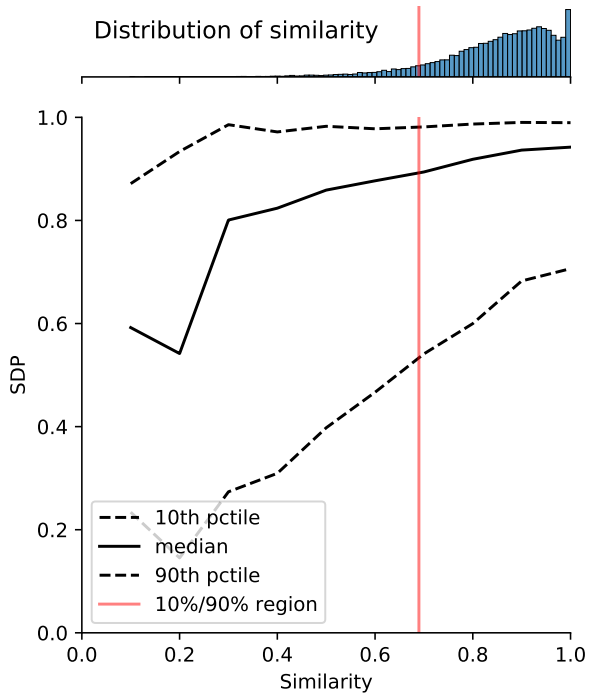

Figure 2.11: Stein dot product (SDP) vs Tanimoto similarity of our test molecules ($n = 25205$) to the closest molecule in the training dataset ($n = 100438$). Results are binned to the nearest decile and the 10%-50%(median)-90% percentiles within each bin are plotted. Additionally, the histogram of the similarities is shown inset above the plot. The vertical red line is the 10th percentile of similarity, plotted at Similarity $\approx 69.0\%$. 10% of test set molecules fall below this similarity value, and 90% of test set molecules fall above.

Library matching performance

In the library matching task, the NIST Replicate Library we use as the query set features molecules that are not seen in the Main Library. Thus, for each molecule in the Replicate Library, we compute its similarity to the Main Library as the similarity to the closest molecule in the Main Library. We bin the molecules into "low similarity" molecules ($n = 29339$) and "high similarity" molecules ($n = 18771$). The cutoff is 90%, below which a molecule is classified as "low similarity", otherwise "high similarity". Low similarity molecules have a mean log10(rank) of 0.11, whereas high similarity molecules have a mean log10(rank) of 0.14. This intuitively makes sense – Replicate Library molecules with high structural similarity to Main Library molecules are likely to have similar spectra in the database, and similar spectra

can often be hard to distinguish from each other, causing the lookup rank to be higher (worse identification) than molecules with lower similarity. More detailed statistics can be found in the Supplement.

2.4.5 Evaluating the impact of the subset enumeration

The way we enumerate substructures (here, atom subsets and chemical subformulae) is critical. Chemical subformulae can be completely enumerated without knowledge of the molecule structure, but atom subsets requires bond-breaking and hydrogen-rearrangements. As we increase the depth to which we break bonds, we generate more fragments and should expect monotonically-increasing recall and coverage of spectra. In Fig. 2.12 we study the final performance of trained SubsetNets as all parameters are held constant except the bond-breaking depth used to generate atom subsets for training is varied. Each model is trained for 1000 epochs or until the validation SDP no longer increases. The highest-performing checkpoint as measured by validation SDP is selected for final metrics. As we increase the depth to which we break bonds from $d = 1$ to 3, we see increases in forward similarity (SDP and DP), but a decrease at $d = 4$. The decrease may be due to the way we randomly select a subset of the atom subsets in order to fit the entire atom subset indicator matrix on GPU. Randomly subsampling the generated atom subsets may throw-out important fragments that we no longer consider for weighting and observation later in the pipeline. In this work, we only focus on subset achievable by bond breaking out to depth 3. Notice that if we add hydrogen rearrangements ("d=3 B&R"), we continue to see improvement in performance. This indicates that further improvements in the recall and physical-plausibility of the generated subsets is likely to boost performance, in addition to increasing the number of atom subsets considered for observation.

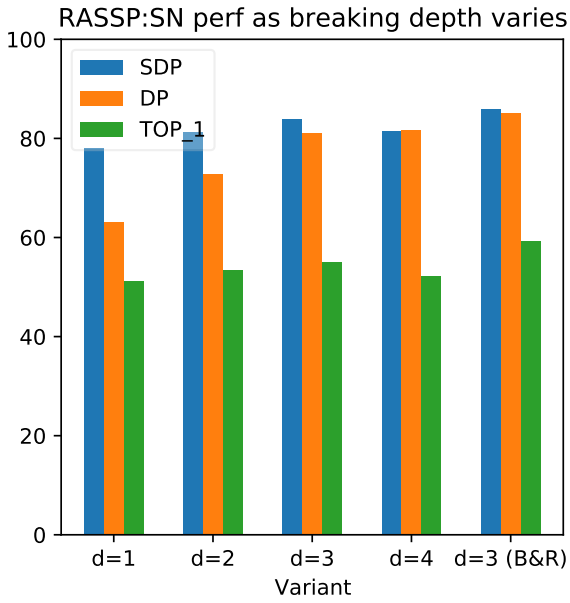

Figure 2.12: Performance of SubsetNet as depth of bond breaking increases. We fix a SubsetNet architecture and dataset (NIST17 Mainlib) and vary the depth to which we break bonds, affecting the number of generated substructures and atom subsets. Training is terminated after 1000 epochs and the final performance on the validation set is reported here. We see that as depth increases to $d = 3$ performance increases, but tapers off at $d = 4$. In addition, adding hydrogen rearrangements (B&R) boosts performance over simply doing more bond breaking.

2.5 Discussion

Previous efforts to learn machine learning models from mass spectral data have focused on better rules-based fragment enumeration schemes or used machine learning (graph neural networks, transformers) to directly predict spectra from molecule embeddings (SMILES strings, fingerprint hashes, etc). Comprehensive substructure enumeration methods tend to have high recall at the cost of low precision, whereas machine learning tends to help recover that precision. In this work, we combine a physically-plausible substructure enumeration process and GNNs, demonstrating that such a fusion outperforms all previous models. We present *SubsetNet* and *FormulaNet*, two models for predicting EI-MS spectra. *FormulaNet* significantly outperforms all previous methods of EI-MS spectral prediction, achieving an average SDP of 92.9% and DP of 93.5% over the largest publicly-available database of EI-MS

spectra. In addition, our predicted spectra may be evaluated indirectly by utilizing them in a library matching (database lookup) task. Here, we also outperform previous methods, achieving a recall at 10 of 98.0%. *SubsetNet* does much better at generalization in the low-data regime, by leveraging more fine-grained information about substructures. Such performance approaches the limits of experimental data (see Fig. 2.7). We generate EI-MS spectra predictions for 73.2M molecules from PubChem and make them freely available.

All computational approaches to predicting EI-MS spectra approaches are fundamentally limited by available data. The largest publicly-available spectral library to date is still the NIST Mass Spectral Library [NIST]. Experimentalists from around the world are free to contribute EI-MS spectra measured at 1 Da resolution to the library. As higher-resolution tandem MS/MS machines come online, spectral databases will increasingly consist of heterogeneous data, mixing experimental spectra measured at many different resolution scales. Importantly, because RASSP predicts a probability distribution over fragments with known exact mass peak distributions, it can be used to predict spectra at arbitrary resolutions by simply changing how we bin the binning of predicted probabilities. As such, our approach is the first approach that can be used to leverage data from many different sources, due to the ability to train against high and low-resolution data simultaneously. It is common to use some form of dot product or cosine similarity as a spectral similarity metric with which to measure forward spectral prediction performance and library matching. However, in higher-resolution tandem MS/MS it is expected that false-positive rate may matter even more. Future work would look at importance of different metrics in measuring spectral prediction performance and integrating supervision from both higher-resolution EI-MS spectral data as well as other types of metadata, such as ionization energy and experimental apparatus.

Each of the modules (subset and subformula enumeration vs. machine learning model for the fragments) can be improved independently. For computational ease, our enumeration process generates fragments by breaking up to and including 3 bonds, and also does all possible

hydrogen rearrangements. There are more exotic fragmentation schemes that we have ignored, and including them can potentially improve the recall of the generated fragments. The graph neural networks we use only consider the atoms. No information about the bonds, besides their bond order, is considered. These models may potentially be improved by incorporating edge information and changes to the model architecture, such as a novel bipartite atom-bond message-passing scheme or other improvements. Together, future improvements may improve both the recall and the precision of our forward model.

An accurate *in silico* forward model for predicting EI-MS spectra can be applied to library search and compound identification. Running similarity search over spectral databases using repeated spectral measurements obtained from NIST Replib achieves an error rate of 1% at 10 using $\text{DP}_{1,0.5}$, which sets the lower bound on library matching accuracy given current EI-MS hardware. By augmenting existing spectral databases with *in silico* spectral predictions from our forward model, we can massively increase the number of molecule candidates considered, potentially increasing the ability for scientists to discover novel and rare compounds. However, the search problem quickly becomes computationally limited. A typical query over the 300K molecules in NIST Mainlib takes about 100ms. Improvements in the computational efficiency of the library matching / database search task can arise from more efficient similarity metrics, approximate computations, and dimensionality reduction via approaches like nearest-neighbor hashing or locality-sensitive hashing. Recent work has already shown that deep learning-based similarity measures can dramatically improve accuracy over simpler cosine similarity measures in database lookup tasks [Matyushin et al., 2020, Ji et al., 2020].

In the long run, we expect computational spectral approaches to enable novel applications. For example, computationally-obtained spectra may be used to augment metabolomics studies by enabling researchers to automatically match spectra to molecules that have never been experimentally studied. Future work could use a good computational forward model for EI-MS to generate large amounts of training data that could then be used as supervision

for an inverse model to further automate this and other types of molecular identification problems. The runtime of these forward models may be improved by further algorithmic improvements to the substructure generation step as well as the machine learning models.

CHAPTER 3

STOCHASTIC SUM-OF-SQUARES FOR PARAMETRIC POLYNOMIAL OPTIMIZATION

Global polynomial optimization is an important tool across applied mathematics, with many applications in operations research, engineering, and physical sciences. In various settings, the polynomials depend on external parameters that may be random. In this chapter, we discuss a stochastic sum-of-squares (S-SOS) algorithm based on the sum-of-squares hierarchy that constructs a series of semidefinite programs to jointly find strict lower bounds on the global minimum and extract candidates for parameterized global minimizers. We prove quantitative convergence of the hierarchy as the degree increases and use it to solve unconstrained and constrained polynomial optimization problems parameterized by random variables. By employing n-body priors from condensed matter physics to induce sparsity, we can use S-SOS to produce solutions and uncertainty intervals for sensor network localization problems containing up to 40 variables and semidefinite matrix sizes surpassing 800×800. This chapter is adapted from the publication [Zhu et al., 2024] (to appear in Neurips 2024). Additional background and supporting material external to the core ideas outlined in this chapter can be found in Chapter 6.

3.1 Introduction

Many effective nonlinear and nonconvex optimization techniques use local information to identify local minima. But it is often the case that we want to find global optima. Sum-of-squares (SOS) optimization is a powerful and general technique in this setting.

The core idea is as follows: suppose we are given polynomials $g_1, \ldots, g_m, f$ where each function is on $\mathbb{R}^d \to \mathbb{R}$ and we seek to determine the minimum value of f on the closed set $\mathcal{S}$: $\mathcal{S} = \{x \in \mathbb{R}^d \mid g_i(x) \geq 0 \; \forall \; i = 1, \ldots, m\}$. Our optimization problem is then to find

$\inf_{x \in \mathbb{R}^d}\{f(x)|x \in \mathcal{S}\}$. An equivalent formulation is to find the largest constant $c \in \mathbb{R}$ (i.e. the tightest lower bound) that can be subtracted from f such that $f - c \geq 0$ over the set $\mathcal{S}$. This reduction converts a polynomial optimization problem over a semialgebraic set to the problem of checking polynomial non-negativity. This problem is NP-hard in general [Garey and Johnson, 2009], therefore one instead resorts to checking if $f - c$ is a sum-of-squares (SOS) function, e.g. in the unconstrained setting where $\mathcal{S} = \mathbb{R}^d$ one seeks to find some polynomials $h_k : \mathbb{R}^d \to \mathbb{R}$ such that $f - c = \sum_k h_k^2$. If such a decomposition can be found, then we have an easily checkable certification that $f - c \geq 0$, as all sum-of-squares are non-negative but not all non-negative functions are sum-of-squares.

Notably, if we restrict the h_k to have maximum degree s, the search for a degree-$2s$ SOS decomposition of a function can be automated as a semidefinite program (SDP) [Nesterov, 2000, Lasserre, 2001, Laurent, 2009]. Solving this SDP for varying degrees s generates the well-known Lasserre (SOS) hierarchy. A given degree s corresponds to a particular level of the hierarchy. Solving this SDP produces a lower bound c_s which has been proven to converge to the true global minimum $c^* = \inf_x f(x)$ as s increases, with finite convergence ($c_s = c^*$ at finite s) for functions with second-order local optimality conditions [Nie, 2014, Bach and Rudi, 2023] and asymptotic convergence with milder assumptions thanks to representation theorems for positive polynomials from real algebraic geometry [Putinar, 1993, Schmüdgen, 2017]. Further work has elucidated both theoretical implications [Putinar, 1993, Lasserre, 2001, 2018, 2023] and useful applications of SOS to disparate fields [Parrilo, 2000, Nie, 2009, de Klerk, 2008, Nie, 2014, Bach and Rudi, 2023, Ahmadi and Majumdar, 2019, Papp and Yildiz, 2019] (see further discussion in Section 6.2).

Motivated by the sum-of-squares certification for a lower bound c on a function $f(x)$, we generalize to the case where the function to be minimized has additional parameters, i.e. $f(x, \omega)$ where x are variables and ω are parameters drawn from some probability distribution $\omega \sim \nu(\omega)$. We seek a function $c(\omega)$ that is the tightest lower bound to $f(x, \omega)$ everywhere:

$f(x,\omega) \geq c(\omega)$ with $c(\omega) \to \inf_x f(x,\omega)$. This setting was originally presented in [Lasserre, 2010] as a "Joint and Marginal" approach to parametric polynomial optimization. With the view that $\omega \sim \nu(\omega)$ and seeking to parameterize the minimizers $x^*(\omega) = \operatorname{argmin}_x f(x,\omega)$, we are reminded of some of the prior work in polynomial chaos, where a system of stochastic variables is expanded into a deterministic function of those stochastic variables [Sudret, 2008, Najm, 2009].

Contributions and outline. Our primary contributions are a quantitative convergence proof for the Stochastic Sum-of-Squares (S-SOS) hierarchy of semidefinite programs (SDPs), a formulation of a new hierarchy (the cluster basis hierarchy) that uses the structure of a problem to sparsify the SDP, and numerical results on its application to the sensor network localization problem.

In Section 3.2, we review the S-SOS hierarchy of SDPs [Lasserre, 2010] and its primal and dual formulations (Section 3.2.1). We then detail how different hierarchies can be constructed (Section 3.2.2). Finally, in Section 3.2.3 (complete proofs in Section 6.5.2) we specialize to compact $X \times \Omega$ and outline the proofs for two theorems on quantitative convergence (the gap between the optimal values of the degree-$2s$ S-SOS SDP and the "tightest lower-bounding" optimization problem goes $\to 0$ as $s \to \infty$) of the S-SOS hierarchy for trigonometric polynomials on $[0,1]^n \times [0,1]^d$ following the kernel formalism of [Fang and Fawzi, 2021, Bach and Rudi, 2023, Slot, 2023]. The first one applies in the general case and the second one applies to the case where $d = 1$.

In Section 3.3 we review the hierarchy's applications in parametric polynomial minimization and uncertainty quantification, focusing on several variants of sensor network localization on $X \times \Omega = [-1,1]^n \times [-1,1]^d$. We present numerical results for the accuracy of the extracted solutions that result from S-SOS, comparing to other approaches to parametric polynomial optimization, including a simple Monte Carlo-based method.

3.2 Stochastic Sum-of-squares (S-SOS)

Notation

Let $\mathcal{P}(S)$ be the space of polynomials on S, where $S \in \{X, \Omega\}$. $X \subseteq \mathbb{R}^n$ and $\Omega \subseteq \mathbb{R}^d$, respectively, where X and Ω are (not-necessarily compact) subsets of their respective ambient spaces $\mathbb{R}^n$ and $\mathbb{R}^d$. A polynomial in $\mathcal{P}(X)$ can be written as $p(x) = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} c_\alpha x^\alpha \in \mathcal{P}(X)$ (substituting $n \to d, x \to \omega, X \to \Omega$ for a polynomial in Ω). Let $x := (x_1, \ldots, x_n), \omega := (\omega_1, \ldots, \omega_d)$, α be a multi-index (size given by context), and c_α be the polynomial coefficients. Let $\mathcal{P}^s(S)$ for some $s \in \mathbb{Z}_{\geq 0}, S \in \{X, \Omega\}$ denote the subspace of $\mathcal{P}(S)$ consisting of polynomials of degree $\leq s$, i.e. polynomials where the multi-indices of the monomial terms satisfy $||\alpha||_1 \leq s$. $\mathcal{P}_{\text{SOS}}(X \times \Omega)$ refers to the space of polynomials on $X \times \Omega$ that can be expressible as a sum-of-squares in x and ω jointly, and $\mathcal{P}^s_{\text{SOS}}(X \times \Omega)$ be the same space restricted to polynomials of degree $\leq s$. Additionally, $W \succcurlyeq 0$ for a matrix W denotes that W is symmetric positive semidefinite (PSD). Finally, $\mathbb{P}(\Omega)$ denotes the set of Lebesgue probability measures on Ω. For more details, see Section 6.1.

3.2.1 Formulation of S-SOS hierarchy

We present two formulations of the S-SOS hierarchy that are dual to each other in the sense of Fenchel duality [Rockafellar, 2015, Boyd and Vandenberghe, 2004]. The primal problem seeks to find the tightest lower-bounding function and the dual problem seeks to find a minimizing probability distribution. Note that the "tightest lower bound" approach is dual to the "minimizing distribution" approach, otherwise known as a "joint and marginal" moment-based approach originally detailed in [Lasserre, 2010].

Primal S-SOS: The tightest lower-bounding function

Consider a polynomial $f(x,\omega) : \mathbb{R}^{n+d} \to \mathbb{R}$ with $x \in X \subseteq \mathbb{R}^n, \omega \in \Omega \subseteq \mathbb{R}^d$ equipped with a probability measure $\nu(\omega)$. We interpret x as our optimization variables and ω as noise parameters, and seek a lower-bounding function $c^*(\omega)$ such that $f(x,\omega) \geq c^*(\omega)$ for all x, ω. In particular, we want the tightest lower bound $c^*(\omega) = \inf_{x \in X} f(x,\omega)$. Note that even when $f(x,\omega)$ is polynomial, the tightest lower bound $c^*(\omega)$ can be non-polynomial. A simple example is the function $f(x,\omega) = (x-\omega)^2 + (\omega x)^2$, which has $c^*(\omega) = \inf_x f(x,\omega) = \omega^4/(1+\omega^2)$ (Section 6.6.1).

For us to select the "best" lower-bounding function, we want to maximize the expectation of the lower-bounding function $c(\omega)$ under $\omega \sim \nu(\omega)$ while requiring $f(x,\omega) - c(\omega) \geq 0$, giving us the following optimization problem over L^1-integrable lower-bounding functions:

$$\begin{aligned} p^* = \sup_{c \in L^1(\Omega)} \quad & \int c(\omega) \mathrm{d}\nu(\omega) \\ \text{s.t.} \quad & f(x,\omega) - c(\omega) \geq 0 \end{aligned} \tag{3.1}$$

Even if we restricted $c(\omega)$ to be polynomial so that the residual $f(x,\omega) - c(\omega)$ is also polynomial, we would still have a challenging nonconvex optimization problem over non-negative polynomials. In SOS optimization, we take a relaxation and require the residual to be SOS: $f(x,\omega) - c(\omega) \in \mathcal{P}_{\text{SOS}}(X \times \Omega)$. Doing the SOS relaxation of the non-negative Equation (3.1) and restricting $c(\omega)$, i.e. $f(x,\omega) - c(\omega)$ to polynomials of degree $\leq 2s$ gives us Equation (3.2), which we call the primal S-SOS degree-$2s$ SDP:

$$\begin{aligned} p^*_{2s} = \sup_{c \in \mathcal{P}^{2s}(\Omega), W \succcurlyeq 0} \quad & \int c(\omega) \mathrm{d}\nu(\omega) \\ \text{s.t.} \quad & f(x,\omega) - c(\omega) = m_s(x,\omega)^T W m_s(x,\omega) \end{aligned} \tag{3.2}$$

where $m_s(x,\omega)$ is a basis function $X \times \Omega \to \mathbb{R}^{a(n,d,s)}$ containing monomial terms of degree

$\leq s$ written as a column vector, and $W \in \mathbb{R}^{a(n,d,s)\times a(n,d,s)}$ a symmetric PSD matrix. Here, $a(n,d,s)$ represents the dimension of the basis function, which depends on the degree s and on the dimensions n, d. For this formulation to find the best degree-$2s$ approximation to the lower-bounding function, we require $g(x,\omega) = m_s(x,\omega)^T W m_s(x,\omega)$ to span $\mathcal{P}^{2s}(X \times \Omega)$. Selecting all combinations of standard monomial terms of degree $\leq s$ suffices and results in a basis function with size $a(n,d,s) = \binom{n+d+s}{s}$.

Dual S-SOS: A minimizing distribution

The formal dual to Equation (3.1) (proof of duality in Section 6.5.1) seeks to find a "minimizing distribution" $\mu(x,\omega)$, i.e. a probability distribution that places weight on the minimizers of $f(x,\omega)$ subject to the constraint that the marginal $\mu_X(\omega)$ matches $\nu(\omega)$:

$$\begin{aligned} d^* = \inf_{\mu \in \mathbb{P}(X\times\Omega)} \quad & \int f(x,\omega) \mathrm{d}\mu(x,\omega) \\ \text{s.t.} \quad & \int_X \mathrm{d}\mu(x,\omega) = \mu_X(\omega) = \nu(\omega) \end{aligned} \tag{3.3}$$

where we have written $\mathbb{P}(X \times \Omega)$ as the space of joint probability distributions on $X \times \Omega$ and $\mu_X(\omega)$ is the marginal of $\mu(x,\omega)$ with respect to ω, obtained via disintegration.

For the primal, we considered polynomials of degree $\leq 2s$. We do the same here. The formal dual becomes a tractable SDP, where the objective turns into moment-minimization and the constraints become moment-matching. Following [Lasserre, 2001, Nie, 2009], let $M \in \mathbb{R}^{a(n,d,s)\times a(n,d,s)}$ be the symmetric PSD moment matrix with entries defined as $M_{i,j} = \int_{X\times\Omega} m_s^{(i)}(x,\omega) m_s^{(j)}(x,\omega) d\mu(x,\omega)$ where $m_s^{(i)}(x,\omega)$ is the i-th element of the basis function m_s. Let $y \in \mathbb{R}^{b(n,d,s)}$ be the moment vector of independent moments that completely specifies M, e.g. in the case that we use all standard monomials of degree $\leq s$ and have $a(n,d,s) = \binom{n+d+s}{s}$, then $b(n,d,s) = \binom{n+d+2s}{2s}$. We write $M(y)$ as the moment matrix that is formed from these independent moments. We have $y_{\alpha(i,j)} =$

$\int_{X\times\Omega} m_s^{(i)}(x,\omega)m_s^{(j)}(x,\omega)d\mu(x,\omega)$ where the multi-index $\alpha(i,j) \in \mathbb{Z}_{\geq 0}^{n+d}$ corresponds to the sum of the multi-indices corresponding to the i-th entry and the j-th entry of $m_s(x,\omega)$.

We write $f(x,\omega)$ in terms of the monomials $f(x,\omega) = \sum_{||\alpha||_1 \leq 2s} f_\alpha [x,\omega]^\alpha$, where $[x,\omega]$ is the concatenation of the $n+d$ variables from x,ω and $\alpha \in \mathbb{Z}_{\geq 0}^{n+d}$ is a multi-index. Note that every monomial $[x,\omega]^\alpha$ has a corresponding moment y_α: $\int [x,\omega]^\alpha \mathrm{d}\mu(x,\omega) = y_\alpha$. We then observe that the integral in the objective reduces to a dot product between the coefficients of f and the moment vector:

$$\int f(x,\omega)\mathrm{d}\mu(x,\omega) = \int \sum_\alpha f_\alpha [x,\omega]^\alpha \mathrm{d}\mu(x,\omega) = \sum_\alpha f_\alpha y_\alpha$$

After converting the distribution-matching constraint $\mu_X(\omega) = \nu(\omega)$ in (3.3) into equality constraints on the moments of ω up to degree $2s$, we obtain the following dual S-SOS degree-$2s$ SDP:

$$\begin{aligned} d_{2s}^* = \inf_{y\in\mathbb{R}^{b(n,d,s)}} \quad & \sum_{||\alpha||_1 \leq 2s} f_\alpha y_\alpha \\ \text{s.t.} \quad & M(y) \succcurlyeq 0 \\ & y_\alpha = m_\alpha \; \forall \; (\alpha, m_\alpha) \in \mathcal{M}_\nu \end{aligned} \tag{3.4}$$

We write $\mathcal{M}_\nu$ as the set of (α, m_α) representing the moment-matching constraints on ω^α up to degree-$2s$, i.e. we want to set $\int_{X\times\Omega} \omega^\alpha \mathrm{d}\mu(x,\omega) = \int_\Omega \omega^\alpha \mathrm{d}\nu(\omega) = m_\alpha$ for all multi-indices $\alpha \in \mathbb{Z}_{\geq 0}^d$ with $||\alpha||_1 \leq 2s$. There are $\binom{d+2s}{2s}$ multi-indices $\alpha \in \mathbb{Z}_{\geq 0}^{n+d}, ||\alpha||_1 \leq 2s$ where only the d entries associated with ω are non-zero, and therefore the number of moment-matching constraints is $|\mathcal{M}_\nu| = \binom{d+2s}{2s}$. Note that the moment matrix $M(y) \in \mathbb{R}^{a(n,d,s)\times a(n,d,s)}$ is a symmetric PSD matrix and is the dual variable to the primal W. Observe also that we require the moments of $\nu(\omega)$ of degree up to $2s$ to be bounded. (3.4) is often a more convenient form than (3.2), especially when working with additional equality or inequality constraints, as we

will see in Section 3.3. For concrete examples of the primal and dual SDPs with explicit constraints, see Section 6.3.

3.2.2 Variations

In this section, we detail two ways of building a hierarchy, one based on the maximum degree of monomial terms in the basis function (Lasserre) and a novel one based on the maximum number of interactions occurring in the terms of the basis function (cluster basis). To define any SOS hierarchy, we first select a monomial basis. Some examples include the standard monomial basis $x_1, \ldots, x_n$, trigonometric/Fourier 1-periodic monomial basis $\sin x_1, \cos x_1, \ldots, \sin x_n, \cos x_n$), or others. Using this basis, we write down a basis function $m(x)$ which comprises some combinations of monomials. Squared linear combinations of the basis functions then span a SOS space of functions: $\mathcal{H} : \{(\sum_i h_i m_i(x))^2\}$.

Standard Lasserre hierarchy

In the Lasserre hierarchy, the basis function $m_s(x)$ is composed of all combinations of monomials up to degree $s \in \mathbb{Z}_{>0}$ and a given level of the hierarchy is set by the maximum degree s. The basis function consists of terms x^α with α a multi-index and $||\alpha||_1 \leq s$. The degree-$2s$ SOS function space parameterized by this basis function is that spanned by $m_s(x)^T W m_s(x)$ for PSD W, i.e. the functions that can result from squaring any linear combination of degree-s polynomials that can be generated from our basis $m_s(x)$. As we increase the degree s, our basis function gets larger and our S-SOS SDP objective values converge to the optimal value of the "tightest lower-bounding" problem Equation (3.1) [Lasserre, 2010].

Cluster basis hierarchy

In this section, we propose a cluster basis hierarchy, wherein we utilize possible spatial organization of the problem to sparsify the problem and reduce the size of the SDP that must be solved [Vandenberghe, 2017, Chen et al., 2023]. The cluster basis is a physically motivated prior often used in statistical and condensed matter physics, where we assume that our degrees of freedom can be arrayed in space, with locally close variables interacting strongly (kept in the model) and globally separated variables interacting weakly (ignored). Moreover, one may also keep only the terms with interactions between a small number of degrees of freedom, such as considering only pairwise or triplet interactions between particles.

In the cluster basis hierarchy, a given level of the hierarchy is defined both by the maximum degree of a variable t and the desired body order b. Body order denotes the maximum number of interacting variables in a given monomial term, e.g. $x_i^a x_j^b x_k^c$ would have body order 3 and total degree $a+b+c$. The basis function $m_{b,t}$ consists of terms x^α with α a multi-index, $||\alpha||_0 \leq b$ (at most b interacting variables can occur in a single term), and $||\alpha||_\infty \leq t$ (each variable can have up to degree t. The maximum degree of the basis function $m_{b,t}$ is then $s = bt$. If we are to compare $m_{b,t}$ from the cluster basis hierarchy with m_s from the Lasserre hierarchy, we find that even when $bt = s$ we still have strictly fewer terms, e.g. in the case where $b = 2, t = 2, s = 4$ we have m_s containing terms of the form x_i^4 but $m_{b,t}$ only has degree-4 terms of the $x_i^2 x_j^2$. For further details, see discussion in Section 6.7.4.

3.2.3 Convergence of S-SOS

As we increase the degree s (either s in the Lasserre hierarchy or b, t in the cluster basis hierarchy) we would expect the SDP objective values p_{2s}^* (Equation (3.2)) to converge to the optimal value p^* and the lower bounding function $c_{2s}^*(\omega)$ to converge to the tightest lower bound $c^*(\omega) = \inf_x f(x, \omega)$. In this work we refer to $p_{2s}^* \to p^*$ and $d_{2s}^* \to d^*$ interchangeably as strong duality occurs in practice despite being difficult to formally verify

(Section 6.4). This convergence is a common feature of SOS hierarchies. In this section we show that using polynomial $c^*_{2s}(\omega)$ to approximate $c^*(\omega)$ still allows for asymptotic convergence in L^1 as $s \to \infty$. We further show how this can be improved with other choices of approximating function classes beyond polynomial $c(\omega)$. We specialize to the particular case of trigonometric polynomials $f(x,\omega), c(\omega)$ on $X = [0,1]^n$ and compact $\Omega \subset \mathbb{R}^d$ and prove asymptotic convergence of the degree-$2s$ S-SOS hierarchy as $s \to \infty$.

$\ln s/s$ convergence using a polynomial approximation to $c^*(\omega)$

We would like to bound the gap between the optimal lower bound $c^*(\omega) = \inf_{x \in X} f(x,\omega)$ and the lower bound $c^*_{2s}(\omega)$ resulting from solving the degree-$2s$ primal S-SOS SDP, i.e.

$$0 \leq c^*(\omega) - c^*_{2s}(\omega) \leq \varepsilon(f,s) \; \forall \; \omega \in \Omega. \tag{3.5}$$

To that end, we need to understand the regularity of c^*. Without further assumptions, we may assume c^* to be Lipschitz continuous, per Proposition 3.2.1.

With Equation (3.5) we may then integrate

$$0 \leq \int_\Omega \inf_x f(x,\omega) - c^*_{2s}(\omega) \mathrm{d}\nu(\omega) \leq |\Omega| \varepsilon(f,s)$$

where we control ε in terms of the degree s. If we can drive $\epsilon \to 0$ as $s \to \infty$ then we are done.

Proposition 3.2.1 (Theorem 2.1 in [Clarke, 1975])**.** *Let $g : X \times Y \to \mathbb{R}$ be polynomial. Then $y \mapsto \inf_{x \in X} g(x,y)$ is Lipschitz continuous.*

Theorem 3.2.1 (Asymptotic convergence of S-SOS)**.** *Let $f : [0,1]^n \times \Omega \to \mathbb{R}$ be a trigonometric polynomial of degree $2r$, $c^*(\omega) = \inf_x f(x,\omega)$ the optimal lower bound as a function of ω, and ν any probability measure on compact $\Omega \subset \mathbb{R}^d$. Let s refer to the degree*

of the basis in both x, ω terms and the degree of the lower-bounding polynomial $c(\omega)$, i.e. $m_s([x,\omega]) : \mathbb{R}^{n+d} \to \mathbb{R}^{a(n,d,s)}$ is the full basis function of terms $[x,\omega]^\alpha$ with $||\alpha||_1 \leq s$ and $c(\omega)$ only has terms ω^α with $||\alpha||_1 \leq s$.

Let p^*_{2s} be the solution to the following S-SOS SDP (c.f. Equation (3.2)) with $m_s(x,\omega)$ a spanning basis of trigonometric monomials with degree $\leq s$:

$$\begin{aligned} p^*_{2s} = \sup_{c \in \mathcal{P}^{2s}(\Omega), W \succcurlyeq 0} & \int c(\omega) \mathrm{d}\nu(\omega) \\ s.t. \quad & f(x,\omega) - c(\omega) = m_s(x,\omega)^T W m_s(x,\omega) \end{aligned}$$

Then there is a constant $C > 0$ depending only on $||f - \bar{f}||_F, ||c^* - \bar{c}^*||, r, \Omega, n, d$ such that the following holds:

$$\int_\Omega \left[c^*(\omega) - c^*_{2s}(\omega)\right] \mathrm{d}\nu(\omega) \leq C \frac{\ln s}{s}$$

where $\bar{f}$ denotes the average value of the function f over $[0,1]^n$, i.e. $\bar{f} = \int_{[0,1]^n} f(x)dx$ and $||f(x)||_F = \sum_{\hat{x}} |\hat{f}(\hat{x})|$ denotes the norm of the Fourier coefficients. Thus we have asymptotic convergence of the S-SOS SDP hierarchy to the optimal value p^* of Equation (3.1) as we send $s \to \infty$.

Proof. The following is an outline of the proof. For complete details, including the full theorem and proof, please see Section 6.5.2.

We define a trigonometric polynomial (t.p.) $c^*_a(\omega)$ of degree s_c that approximates the lower-bounding function such that $c^*(\omega) = \inf_x f(x,\omega) \geq c^*_a(\omega)$. The error integral breaks apart into two terms, one bounding the approximation error between $c^*(\omega)$ and $c^*_a(\omega)$, and the other bounding the error between the approximate lower-bounding t.p. $c^*_a(\omega)$ and the SOS lower-bounding t.p. $c^*_{2s}(\omega)$.

We then follow the proofs of [Fang and Fawzi, 2021, Bach and Rudi, 2023, Slot, 2023] wherein we define an invertible linear operator T that constructs a SOS function out of a non-negative function, and show that such an operator exists for sufficiently large s. The core

modification is to the operator T which is defined as an integral operator over two kernels $q_x(x), q_\omega(\omega)$, i.e.

$$Th(x,\omega) = \int_{X\times\Omega} |q_x(x-\bar{x})|^2 |q_\omega(\omega-\bar{\omega})|^2 h(\bar{x},\bar{\omega}) \mathrm{d}\bar{x}\mathrm{d}\bar{\omega}$$

□

$1/s$ convergence using a piecewise-constant approximation to $c^*(\omega)$

Prior work [Bach and Rudi, 2023] achieves $1/s^2$ convergence for the regular SOS hierarchy without further assumptions. In the previous section, we could only achieve $\ln s/s$ due to the need to first approximate the tightest lower-bounding function $c^*(\omega)$ with a polynomial approximation, which converges at a slower rate. To accelerate the convergence rate, we want to control the regularity of $c^*(\omega)$. We can achieve $1/s$ by approximating the $c^*(\omega)$ pointwise instead of using a smooth parameterized polynomial. By constructing a domain decomposition of Ω and finding a SOS approximation in x for each domain, we can stitch these together to build a piecewise-constant approximation to the lower-bounding function c^*.

In the one-dimensional case $\Omega \subset \mathbb{R}$ (full proof in Section 6.5.2) we achieve the following:

Proposition 3.2.2. *Let $\Omega \subset \mathbb{R}$ be a compact interval and f be a trigonometric polynomial of degree $2r$. Let $\{\omega_i\}$ be equidistant grid points in Ω and s_p the number of such points. Denote by $c_s^*(\omega_i)$ the best SOS approximation of degree s of $x \mapsto f(x,\omega_i)$ and define*

$$c_s^* = \sum_{i=1}^{s_p} c_s^*(\omega_i) 1_{[\omega_i,\omega_{i+1}]}.$$

Then we have for some constant C' depending only on $\max_{\omega_i} ||f(\omega_i, \cdot) - \bar{f}(\omega_i, \cdot)||_F, r, n, \Omega, s_p$*:*

$$\int_\Omega c^*(\omega) - c_s^*(\omega) \mathrm{d}\omega \leq \max_{\omega_i} \|f(\omega_i, \cdot) - \bar{f}(\omega_i, \cdot)\|_F \left[1 - \left(1 - \frac{6r^2}{s^2}\right)^{-n}\right] |\Omega| + \frac{C}{s_p} \leq C' \frac{1}{s^2}$$

3.3 Numerical experiments

We present two numerical studies of S-SOS demonstrating its use in applications. The first study (Section 3.3.1) numerically tests how the optimal values of the SDP Equation (3.2) p_{2s}^* converge to p^* of the original primal Equation (3.1) as we increase the degree. The second study (Section 3.3.2) evaluates the performance of S-SOS for solution extraction and uncertainty quantification in various sensor network localization problems.

3.3.1 Simple quadratic SOS function

As a simple illustration of S-SOS, we test it on the SOS function

$$f(x, \omega) = (x - \omega)^2 + (\omega x)^2 \tag{3.6}$$

with $x \in \mathbb{R}, \omega \in \mathbb{R}$. The lower bound $c^*(\omega) = \inf_x f(x, \omega)$ can be computed analytically as $c^*(\omega) = \omega^4/(1 + \omega^2)$. Assuming $\omega \sim \text{Uniform}(-1, 1)$, we get that the objective value for the "tightest lower-bounding" primal problem Equation (3.1) is $p^* = \int_{-1}^{1} \frac{\omega^4}{2(1+\omega^2)} d\omega = \frac{\pi}{4} - \frac{2}{3} \approx 0.1187$. For further details, see Section 6.6.

We are interested in studying the quantitative convergence of the S-SOS hierarchy numerically. The idea is to solve the primal (dual) degree-$2s$ SDP to find the tightest polynomial lower bound (the minimizing probability distribution) for varying degrees s. As s gets larger, the basis function $m_s(x)$ gets larger and the objective value of the SDP Equation (3.2) p_{2s}^* should converge to the theoretical optimal value p^*.

In Figure 3.1 we see very good agreement between p^* and p_{2s}^* with exponential convergence

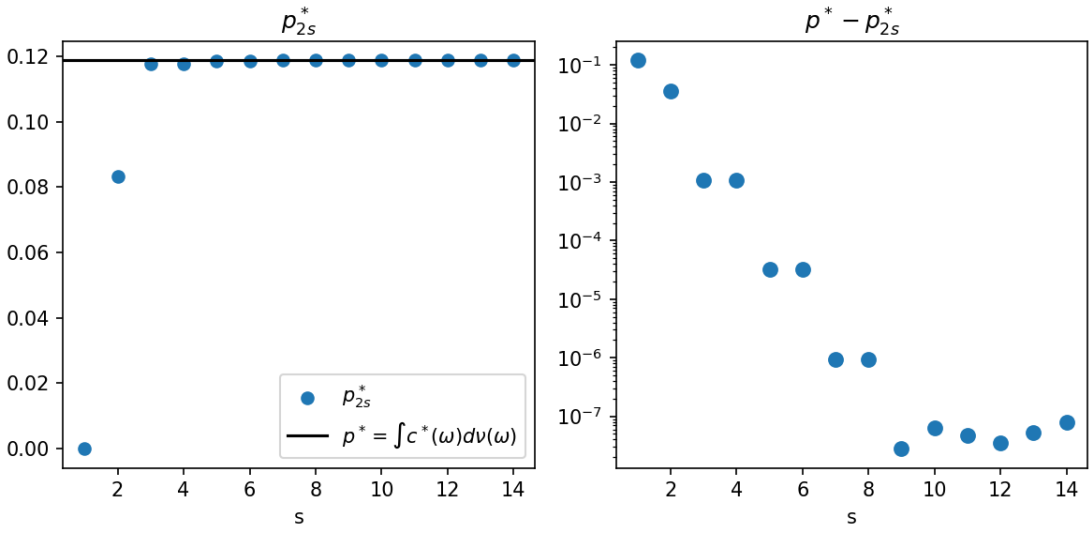

Figure 3.1: Comparison between the objective value p^*_{2s} from solving the degree-$2s$ S-SOS SDP and the objective value p^* resulting from the best-possible lower bound $c^*(\omega)$ for noise drawn as $\omega \sim \text{Uniform}(-1, 1)$. $p^* = \int c^*(\omega) d\nu(\omega) = \frac{\pi}{4} - \frac{2}{3} \approx 0.1187$ is plotted as the line in black and the p^*_{2s} values are shown as blue dots (left) with the gap between the values $p^* - p^*_{2s}$ (right).

as s increases. This is much faster than the rate we found in Section 3.2.3, but agrees with the exponential convergence results from [Bach and Rudi, 2023] achieved with local optimality assumptions. Due to the simplicity of (3.6), it's not surprising that we see much faster convergence. In fact, for most typical functions, we might expect convergence much faster than the worst-case rate. The tapering-off of the convergence rate is likely attributed to the numerical tolerance used in our solver (CVXPY/MOSEK), as we observed that increasing the tolerance shifts the best-achieved gap higher.

3.3.2 Sensor network localization

Sensor network localization (SNL) is a common testbed for global optimization and SDP solvers due to the high sensitivity and ill-conditioning of the problem. In SNL, one seeks to recover the positions of N sensors $X \in \mathbb{R}^{N \times \ell}$ positioned in $\mathbb{R}^\ell$ given a set of noisy observations of pairwise distances $d_{ij} = ||x_i - x_j||$ between the sensors [Nie, 2009, So and Ye, 2007]. To

have a unique global minimum and remove symmetries, sensor-anchor distance observations are often added, where several sensors are anchored at known locations in the space. This can improve the conditioning of the problem, making it "easier" in some sense.

Definitions

We define a SNL *problem instance* with $X \in [-1,1]^{N\times\ell}$ as the ground-truth positions for $\mathcal{S} = \{1,2,\ldots,N\}$ sensors, $A \in [-1,1]^{K\times\ell}$ as the ground-truth positions for $\mathcal{A} = \{1,2,\ldots,K\}$ anchors, $\mathcal{D}_{ss}(r) = \{d_{ij} = ||x_i - x_j|| : i,j \in \mathcal{S} \text{ and } d_{ij} \leq r\}$ as the set of observed sensor-sensor distances and $\mathcal{D}_{sa}(r) = \{d_{ik} = ||x_i - a_k|| : i \in \mathcal{S}, k \in \mathcal{A} \text{ and } d_{ik} \leq r\}$ as the set of observed sensor-anchor distances, both of which depend on some sensing radius r.

Writing $x_i, a_k \in [-1,1]^\ell$ as the unknown positions of the i-th sensor and the k-th anchor, we can write the potential function to be minimized as a polynomial:

$$f(x,\omega;X,A,r) = \underbrace{\sum_{d_{ij}\in\mathcal{D}_{ss}(r)} (||x_i - x_j||_2^2 - d_{ij}(\omega)^2)^2}_{\text{sensor-sensor interactions}} + \underbrace{\sum_{d_{ik}\in\mathcal{D}_{sa}(r)} (||x_i - a_k||_2^2 - d_{ik}(\omega)^2)^2}_{\text{sensor-anchor interactions}} \tag{3.7}$$

The observed sensor-sensor and sensor-anchor distances $d_{ij}(\omega), d_{ik}(\omega)$ can be perturbed arbitrarily, but in this work we focus on linear uniform noise, i.e. for a subset of observed distances we have $d_{ij,k}(\omega) = d_{ij}^* + \epsilon\omega_k$ with $\omega_k \sim \text{Uniform}(-1,1)$. Other noise types may be explored, including those including outliers, which may be a better fit for robust methods (Section 6.7.2).

Equation (3.7) contains soft penalty terms for sensor-sensor terms and sensor-anchor terms. We can see that this is a degree-4 polynomial in the standard monomial basis elements, and a global minimum of this function is achieved at $f(X, \mathbf{0}^d; X, A, r) = 0$ (where the distances have not been perturbed by any noise). In general for non-zero ω (measuring distances under

noise perturbations) we expect the function minimum to be > 0, as there may not exist a configuration of sensors $\hat{X}$ that is consistent with the observed noisy distances.

We can also support equality constraints in our solution, in particular hard equality constraints on the positions of certain sensors relative to known anchors. This corresponds to removing all sensor-anchor soft penalty terms from the function and instead selecting $N_H < N$ sensors at random to exactly fix in known positions via equality constraints in the SDP. The SDP is still large but the effective number of variable sensors has been reduced to $N' = N - N_H$.

A given SNL *problem type* is specified by a spatial dimension ℓ, N sensors, K anchors, a sensing radius $r \in (0, 2\sqrt{\ell})$, a noise type (linear), and anchor type (soft penalty or hard equality). Once these are specified, we generate a random *problem instance* by sampling $X \sim \text{Uniform}(-1, 1)^n, A \sim \text{Uniform}(-1, 1)^d$. The potential $f(x, \omega)$ for a given instance is formed (either with sensor-anchor terms or not, with terms kept based on some sensing radius r, and noise variables appropriately added).

The number of anchors is chosen to be as few as possible so as to still enable exact localization, i.e. $K = \ell + 1$ anchors for a SNL problem in ℓ spatial dimensions. The SDPs are formulated with the help of SymPy [Meurer et al., 2017] and solved using CVXPY [Diamond and Boyd, 2016, Agrawal et al., 2018] and Mosek [ApS, 2023] on a server with two Intel Xeon 6130 Gold processors (32 physical cores total) and 256GB of RAM. For an expanded discussion and further details, see Section 6.7.

Evaluation metrics

The accuracy of the recovered solution is of primary interest, i.e. our primary evaluation metric should be the distance between our extracted sensor positions x and the ground-truth sensor positions X, i.e. $\text{dist}(x, X)$. Because the S-SOS hierarchy recovers estimates of the sensor positions $\mathbb{E}[x_i]$ along with uncertainty estimates $\text{Var}[x_i]$, we would like to

measure the distance between our ground-truth positions X to our estimated distribution $p(x) = \mathcal{N}(\mathbb{E}[x], \mathrm{Var}[x])$. The Mahalanobis distance δ_M (Equation (3.8)) is a modified distance metric that accounts for the uncertainty [Mahalanobis, 1936]. We use this as our primary metric for sensor recovery accuracy.

$$\delta_M(X, \mathcal{N}(\mu, \Sigma)) := \sqrt{(X-\mu)^T \Sigma^{-1} (X-\mu)} \tag{3.8}$$

As our baseline method, for each problem instance we apply a basic Monte Carlo method detailed in Algorithm 2 (Section 6.7.3) where we sample $\omega \sim \nu(\omega)$, use a local optimization solver to find $x^*(\omega) = \inf_x f(x, \omega)$, and use this to estimate $\mathbb{E}_{\omega\sim\nu}[x], \mathrm{Var}_{\omega\sim\nu}[x]$. Note that though this non-SOS method achieves some estimate of the dual SDP objective $\int f(x,\omega) d\mu(x,\omega)$, it is not guaranteed to be a lower bound.

Results

Recovery accuracy. In Table 3.1 we see a comparison of the S-SOS method and the MCPO baseline. Each row corresponds to one SNL problem type, i.e. we fix the physical dimension ℓ, the number of anchors $K = \ell+1$, and select the sensing radius r and the noise scale ϵ. We then generate $L = 20$ random instances of each problem type, corresponding to a random realization of the ground-truth sensor and anchor configurations $X \in [-1,1]^{N\times\ell}, A \in [-1,1]^{K\times\ell}$, producing a $f(x,\omega)$ that we then solve the SDP for (in the case of S-SOS) or do pointwise optimizations for (in the case of MCPO). Each method outputs estimates for the sensor positions and uncertainty around it as a $\mathcal{N}(\mathbb{E}[x], \mathrm{Cov}[x])$, which we then compute δ_M for (see Equation (3.8)), treating each dimension as independent of each other (i.e. X as a flat vector). Each instance solve gives us one observation of δ_M or each method, and we report the median and the $\pm 1\sigma_{34\%}$ values over the $L = 20$ instances we generate.

Table 3.1: Comparison of S-SOS and MCPO solution extraction accuracy. We present the Mahalanobis distance δ_M (Equation (3.8)) of the the true sensor positions X^* to the extracted distribution $\mathcal{N}(\mathbb{E}[x], \mathrm{Var}[x])$ over solutions recovered from S-SOS for varying SNL problem types. ℓ is the spatial dimension, r is the sensing radius used to cutoff terms in the potential $f(x, \omega)$, ϵ is the noise scale, N_H is the number of hard equality constraints used (sensors fixed at known locations), N_C is the number of clusters used (see Section 6.7.4), and N is the number of sensors used. Each SNL problem instance has $K = \ell + 1$ anchors used in the potential (if $N_H = 0$). The MCPO values are estimated with $T = 50$ Monte Carlo iterates. Each entry is $\hat{\mu} \pm \hat{\sigma}$ where $\hat{\mu}$ is the median and robust standard-deviation ($\sigma_{34\%}$) estimated over 20 runs of the same problem type with varying random initializations of the sensor positions. The entries with the lowest median δ_M are bolded. We also compare the number of elements in the full basis a_f, the cluster basis a_c, and the reduction multiple when using the cluster basis a_f/a_c. When passing to the cluster basis, a_f/a_c is how much the semidefinite matrix shrinks by.

Parameters						Basis comparison			M-distance (δ_M)	
ℓ	r	ϵ	N_H	N_C	N	a_f	a_c	a_f/a_c	S-SOS	MCPO
1	0.5	0.3	0	1	10	78	78	1x	$\mathbf{0.94 \pm 0.22}$	2.61 ± 3.86
1	1.0	0.3	0	1	10	78	78	1x	$\mathbf{0.29 \pm 0.16}$	1.10 ± 0.58
1	1.5	0.3	0	1	10	78	78	1x	$\mathbf{0.11 \pm 0.11}$	0.86 ± 0.52
1	1.5	0.3	2	1	10	78	78	1x	$\mathbf{0.24 \pm 0.37}$	1.06 ± 1.28
1	1.5	0.3	4	1	10	78	78	1x	$\mathbf{0.10 \pm 0.03}$	0.61 ± 0.41
1	1.5	0.3	6	1	10	78	78	1x	$\mathbf{0.06 \pm 0.04}$	0.48 ± 0.32
1	1.5	0.3	8	1	10	78	78	1x	$\mathbf{0.04 \pm 0.02}$	0.31 ± 0.17
2	1.5	0.1	0	9	9	406	163	2.5x	$\mathbf{2.86 \pm 0.94}$	1562.39 ± 596.29
2	1.5	0.1	0	9	15	820	317	2.6x	$\mathbf{3.25 \pm 1.19}$	1848.65 ± 650.45

3.4 Discussion

In this work, we discuss the stochastic sum-of-squares (S-SOS) method to solve global polynomial optimization in the presence of noise, prove two asymptotic convergence results for polynomial f and compact Ω, and demonstrate its application to parametric polynomial minimization and uncertainty quantification along with a new cluster basis hierarchy that enables S-SOS to scale to larger problems. In our experiments, we specialized to sensor network localization and low-dimensional uniform random noise with small n, d. However, it is relatively straightforward to extend this method to support other noise types (such

as Gaussian random variates without compact support, which we do in Section 6.6.4) and support higher-dimensional noise with $d \gg 1$.

Scaling this method to larger problems $n \gg 1$ is an open problem for all SOS-type methods. We take the approach of sparsification, by making the cluster basis assumption to build up a block-sparse W. We anticipate that methods that leverage sparsity or other structure in f will be promising avenues of research, as well as approximate solving methods that avoid the explicit materialization of the matrices W, M. For example, we assume that the ground-truth polynomial possesses the block-sparse structure because our SDP explicitly requires the polynomial $f(x, \omega)$ to exactly decompose into some lower-bounding $c(\omega)$ and SOS $f_{\text{SOS}}(x, \omega)$. Relaxing this exact-decomposition assumption and generalizing beyond polynomial $f(x, \omega), c(\omega)$ may require novel approaches and would be an exciting area for future work.

CHAPTER 4

GENERATIVE DIFFUSION PROCESSES: A DEEP DIVE INTO SCORE FUNCTION STRUCTURE

4.1 Introduction

Diffusion models have seen tremendous success in modern generative modeling applications, ranging from image generation to audio synthesis. What are the core ideas? This chapter of the thesis will cover the basics of diffusion models, and then explore how we can relax some of the assumptions made in the standard diffusion model to improve its performance. Section 4.2 will cover the basics of diffusion processes in physics and probabilistic generative modeling. We will review some of the core ideas in statistical physics that relate to diffusion, including random walks, the Langevin equation, and the Fokker-Planck equation. Section 4.3 will introduce some of the methods we will use, including the core ideas of diffusion processes we use and the modeling approaches we take, which focus on interpretable and simple function approximators. Section 4.4 will present a series of vignettes that explore the structure of simple diffusion processes, focusing on cases where the score function is analytically known and numerically verifying the behavior.

4.2 Background

To understand diffusion generative models, we will need to understand both diffusion processes in physics and the principles of probabilistic generative modeling. In this section, we will discuss how diffusion processes arise naturally in physics, and how seeking to model them mathematically led mathematicians and physicists to the Fokker-Planck equation, which is the most general dynamics governing the conserved transport of a density. After discussing Monte Carlo methods briefly, we then dive into diffusion processes in generative modeling, starting

with a look at early work in diffusion models, ranging from the denoising score matching idea of [Vincent, 2011], to the DDPMs of [Sohl-Dickstein et al., 2015], and culminating in the line of work [Song et al., 2018, 2019, Song and Ermon, 2020, Song et al., 2021] that advance the continuous-time ideas of SDE and ODE diffusion.

4.2.1 Diffusion in physics

Random walk in 1D

In physics, diffusion arises as an irreversible process where particles spread out from regions of high concentration to regions of low concentration, typically from thermalization or random motion. Brownian motion, as discovered by Brown in 1827 and later explained by Einstein in 1905, is a classic example of diffusion. But an even more simple case where the basic physics can be understood is that of the discrete random walk on a 1D lattice. Consider a particle that starts at the origin and moves left or right with equal probability at each time step. The probability distribution of the particle's position at time t is given by the binomial distribution:

$$p(x,t) = \frac{1}{2^t}\binom{t}{\frac{t+x}{2}}$$

where $x \in \{-t, -t+1, \ldots, t-1, t\}$ is the position of the particle at time t. We can see that the mean position of the particle is $\mathbb{E}[x(t)] = 0$ and the variance is $\mathrm{Var}[x(t)] = t$. As $t \to \infty$, the distribution converges to the normal distribution with the same parameters via the central limit theorem, giving us

$$p(x,t) \to \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{x^2}{2t}\right)$$

This is the simplest example of a diffusion process, where we have a single particle at a known place at time $t = 0$.

Continuous-time Langevin and the Fokker-Planck equation

Now we can pass to the continuous time limit. Consider $x(t) \in \mathbb{R}^d$ a stochastic process that describes the position of a particle at time t, evolving under the influence of a deterministic drift term $a(x,t) \in \mathbb{R}^d$ and a stochastic term with some diffusion coefficient $b(x,t) \in \mathbb{R}$. The diffusion coefficient can be non-scalar but for our purposes it suffices to consider the scalar case. The resulting stochastic differential equation (SDE) is the Langevin equation and is given by:

$$dx(t) = a(x,t)dt + b(x,t)dW(t)$$

where $dW(t)$ is the Wiener process, a continuous-time stochastic process that is the limit of a random walk with $\langle dW(t) \rangle = 0$, $\langle dW(t)^2 \rangle = dt$.

Suppose we initialize a density of particles at time $t = 0$ and ask: What happens to the distribution of particles at time t? The equation governing the time evolution of this density is the Fokker-Planck equation. There are several formal ways to derive the Fokker-Planck equation, but a simpler way may be to consider that Ito's lemma gives us for some test function $f(x,t)$:

$$df(x(t),t) = \left(\frac{\partial f}{\partial t} + a(x,t)\frac{\partial f}{\partial t} + \frac{1}{2}b(x,t)^2 \frac{\partial^2 f}{\partial x^2} \right) dt + b(x,t)\frac{\partial f}{\partial t} dW(t)$$

Now take the time derivative and expectation of both sides and use the fact that $\langle dW(t) \rangle = 0$ to get

$$\frac{d}{dt}\langle f(x(t),t) \rangle = \int \left(\frac{\partial f}{\partial t} + a(y,t)\frac{\partial f}{\partial y} + \frac{1}{2}b(y,t)^2 \frac{\partial^2 f}{\partial y^2} \right) p(y,t) dy$$

Integrating by parts and assuming boundary terms vanish (due to decay conditions at $\pm\infty$ on f), we get:

$$\frac{d}{dt}\langle f(x(t),t) \rangle = \int f(y,t) \left(-\frac{\partial p}{\partial t} - \frac{\partial}{\partial y}(a(x,t)p(y,t)) + \frac{1}{2}\frac{\partial^2}{\partial y^2}(b(y,t)^2 p(y,t)) \right) dy$$

Now let $f(x,t)$ be a martingale, which means that the whole expression should be zero. The PDE inside the parentheses is set to zero, giving us the Fokker-Planck equation:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(a(x,t)p(x,t)) + \frac{1}{2}\frac{\partial^2}{\partial x^2}(b(x,t)^2 p(x,t))$$

Indeed, any density that evolves according to a Fokker-Planck equation can be thus termed a diffusion process.

The effect of the drift and diffusion terms

As we will see, the choice of the drift and diffusion terms $a(x,t)$ and $b(x,t)$ is crucial in determining the behavior of the diffusion process.

No drift term, constant diffusion term: Heat equation Suppose there is no drift term $a(x,t) = 0$ and the diffusion term is $b(x,t) = \sqrt{2}$. Then the Fokker-Planck equation becomes:

$$\frac{\partial p}{\partial t} = \frac{1}{2}\frac{\partial^2}{\partial x^2}p(x,t)$$

This is the heat equation, which describes the diffusion of heat in a medium. In an unbounded domain, the solution to the heat equation is the Gaussian distribution:

$$p(x,t) = \frac{1}{\sqrt{4\pi t}}\exp\left(-\frac{x^2}{4t}\right)$$

We can see that the variance of the distribution grows linearly with time, which is a characteristic of diffusion processes. Indeed, in this case we also see there is no stationary distribution, as the variance grows indefinitely.

Drift term, no diffusion term: Advection equation In the case where there is a drift term but no diffusion term, we lose the stochastic element and the process becomes

deterministic. The Fokker-Planck equation becomes the advection equation:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}(a(x,t)p(x,t))$$

This equation describes the transport of particles by a velocity field $a(x,t)$.

Drift and diffusion terms In other cases with a drift term, the stationary distribution can be found by setting the time derivative to zero. For example, if $a(x,t) = -\nabla V(x)$ and $b(x,t) = \sqrt{2}$, where $V(x)$ is a potential function, then we find the stationary distribution is given by the Boltzmann distribution:

$$p(x) = \frac{1}{Z}\exp(-V(x))$$

where Z is the normalization constant.

In this case, we see that the drift term pushes the particles towards regions of low potential energy, while the diffusion term spreads the particles out in the absence of a potential gradient. The strength of the diffusion term can be thought of as a measure of the noise in the system, and in physics the higher the noise, the higher the "temperature".

Monte Carlo methods and diffusion

Beyond being useful for classical and quantum statistical mechanics, the Fokker-Planck equation has found applications in Monte Carlo methods. Thinking about what the stationary density is in the presence of drift and diffusion terms automatically suggests a strategy for generating samples from any Gibbs (Boltzmann) probability density, i.e. one of the form

$$p(x) = \frac{1}{Z}\exp(-V(x)).$$

This is especially useful in cases where the normalization constant Z may be intractable. Fear not – Langevin Monte Carlo will at least enable us to draw samples! We can simply initialize samples x_0 from any distribution that has support at least covering the support of $p(x)$, and then run the Langevin dynamics to generate trajectories that will be distributed according to the target distribution $p(x)$. This is the idea behind Langevin Monte Carlo, a powerful tool for sampling from high-dimensional distributions.

Langevin Monte Carlo Let $p(x) = \frac{1}{Z}\exp(-V(x))$ be the target distribution. Consider the overdamped Langevin SDE:

$$dx(t) = -\nabla V(x)dt + \sqrt{2}dW_t \tag{4.1}$$

where W_t is a d-dimensional Brownian motion. As can be seen from plugging the expression into the Fokker-Planck equation, the stationary distribution of this process is the Boltzmann distribution $p(x) \propto \exp(-V(x))$. Assume $V(x)$ is L-smooth, i.e. continuously differentiable and $\exists$ a constant L such that for all $x, y \in \mathbb{R}^d$,

$$||\nabla V(x) - \nabla V(y)|| \leq L||x - y||$$

This SDE can be proven to define a Markov semigroup $(P_t)_{t\geq 0}$ with $P_t p_0$ converging to p in total variation so long as p_0 has sufficient support [Roberts and Tweedie, 1996, Dwivedi et al., 2018].

We are interested in numerical solutions to this SDE. A common way to do this is to discretize the SDE and use the first-order Euler-Maruyama method:

$$x_{n+1} = x_n - \nabla V(x_n)\Delta t + \sqrt{2\Delta t}z_n$$

where $z_n \sim N(0, I_d)$.

One important caveat: the character of $V(x)$ matters. In particular, Langevin Monte Carlo is effective when $V(x)$ is convex, i.e. $p(x)$ is log-concave. Intuitively, we can see this by considering that the Langevin dynamics will push particles towards regions of low potential energy, and the diffusion term will spread them out. Thus, it is most effective at capturing unimodal densities. If the potential is not convex, the particles may get stuck in local minima [Brooks et al., 2011].

It is common to add an additional accept-reject step based on the detailed balance conditions of Metropolis-Hastings. Known as Metropolis-adjusted Langevin dynamics (MALA), this method can improve the acceptance rate of the samples.

4.2.2 Generative modeling and diffusion processes

It is useful to distinguish between *discriminative* models and *generative* models. Generative models seek to model the joint distribution of the data $p(x, y)$ whereas the discriminative approach models the conditional distribution $p(y|x)$. One can easily see that generative modeling methods should be a superset of discriminative modeling methods, as a trained generative model that provides the conditional data likelihood $p(x|y)$ and a prior $p(y)$ over labels will give rise to a discriminative model $p(y|x)$ via Bayes' theorem:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Our intuition suggests that in many cases we would expect discriminative models to outperform generative models *if* the only metric that matters is classification accuracy – and this is indeed borne out in practice [Ng and Jordan, 2001].

However, there are many reasons we may want to trade off classification accuracy for the ability to model the entire data distribution. Some tasks that require one to model the full distribution of data include the following:

- **Density estimation**: Given an input x, we would like to estimate the likelihood $p(x)$. A well-learnt $p(x)$ can be used for other applications such as anomaly detection, where we would compare likelihood of a new data point to a threshold.

- **Denoising**: Given a corrupted input x', we may want to recover the original input x by finding the most likely x given x', i.e. $\text{argmax}_x p(x|x')$.

- **Sampling**: We would like to draw new samples $x \sim p(x)$, and perhaps even condition on some other variables y. Generating high-fidelity samples from data can be quite useful in practical scenarios, as we have seen in recent applications to image and video generation (creative applications), audio synthesis (text-to-speech, music generation), natural language and even protein and DNA sequence generation.

None of these tasks can be accomplished easily by traditional discriminative models. They require us to expand our horizons. An introduction to generative modeling for all these tasks could take up a whole thesis, but in this thesis we will only discuss a few specific cases, starting with diffusion and denoising models.

Energy-based models

A common approach in generative modeling when given some dataset of samples $\mathcal{D} = \{x_1, \ldots, x_n\}$ with $x \in \mathbb{R}^d$ is to specify some parameterized model $p_\theta(x)$ and fit it via maximum-likelihood, i.e.

$$\theta^* = \text{argmax}_\theta \sum_{i=1}^{n} \log p_\theta(x_i)$$

$p_\theta(x)$ must be a density and therefore we must have $\int p_\theta(x)dx = 1$. An easy way to enforce this normalization is to specify $p_\theta(x)$ as an exponential family distribution, i.e.

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(-E_\theta(x))$$

where $E_\theta(x)$ is an energy function that we want to fit and $Z(\theta)$ is the normalization constant. Computing the partition function $Z_\theta = \int \exp(-E_\theta(x))dx$ is generally intractable, which typically renders maximum-likelihood infeasible for most reasonable energy functions.

Previous approaches to dealing with this conundrum have included:

- **Strong constraints on the form of the model**: Employing strong constraints on the model form can enable the partition function Z_θ to be easily computed. This includes normalizing flow models, which explicitly model the transformation between a simple base distribution (e.g., a Gaussian) and the target distribution. The advantage of normalizing flows is that they provide a tractable Jacobian determinant, which allows for exact computation of the normalization constant, bypassing the need for approximating $Z(\theta)$. Normalizing flows construct an invertible transformation $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ between a base distribution $p_u(u)$ and the desired distribution $p_\theta(x)$, where $x = f_\theta(u)$. The probability density is computed as:

$$p_\theta(x) = p_u(f_\theta^{-1}(x)) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right|$$

- **Approximate inference methods**: When exact computation of $Z(\theta)$ is infeasible, approximate methods such as Monte Carlo estimation or importance sampling can be employed. These methods approximate the gradient of the log-likelihood (which involves $Z(\theta)$) by sampling from the model's distribution or a surrogate distribution. Contrastive divergence (CD) is a well-known approximation method that sidesteps the need to compute $Z(\theta)$ by focusing on minimizing the difference between two distributions during training. Specifically, CD uses a combination of sampling and optimization steps to efficiently estimate the gradients.

- **Variational methods**: Another class of methods uses variational techniques to approximate the log-partition function. These methods introduce an auxiliary distribution

$q(x)$ and optimize a lower bound on the log-likelihood by minimizing the Kullback-Leibler (KL) divergence between $q(x)$ and $p_\theta(x)$. While this approach doesn't give exact likelihoods, it provides a tractable way to train EBMs by circumventing direct computation of $Z(\theta)$.

- **Score matching and denoising score matching**: The topic of our work, these techniques bypass the normalization constant directly and focus on matching the Stein score function of a density, i.e. the gradient of the log-probability $\nabla_x \log p(x)$. Denoising score matching introduces noise into the data and trains the model to reconstruct the clean data, further simplifying training. These methods provide an alternative to likelihood-based training by focusing on matching local properties of the data distribution.

Previous work in diffusion models

Motivating score matching Physics has long played a role in providing intuition and theory to guide the development of new machine learning models. We saw earlier that ideas from the physics of diffusion could be used to generate samples from a Boltzmann distribution (Gibbs density), in particular using the overdamped Langevin diffusion. A Gibbs density $p(x) \propto \exp(-V(x))$ is specified by an energy function $V(x)$. Actually evaluating the density $p(x)$ is generally challenging, but generating samples can be done easily.

Now suppose we are asked: given some samples $\mathcal{D} = \{x_1, \ldots, x_N\}$ (i.e. an empirical data distribution $p_{\mathcal{D}}(x)$), how do we fit an energy-based model (Gibbs density) to it?

This is where score matching comes in. For some density $p(x)$, we define the score function as

$$s(x) := \nabla_x \log p(x)$$

If we can approximate $s(x)$ with some parameterized score function $s_\theta(x)$, then we are done –

we've obtained the (negative) energy function for the Gibbs density and can apply Monte Carlo techniques to draw new samples from $p_{\mathcal{D}}(x)$.

Naïvely we might expect that minimizing the explicit score matching loss (Fisher divergence between the true score function and our score model):

$$\mathcal{L}_{\text{ESM}} = \mathbb{E}_{x \sim p_{\mathcal{D}}(x)} \left[\frac{1}{2} ||s_\theta(x) - \nabla_x \log p(x)||_2^2 \right] \tag{4.2}$$

would get us there. But the issue with this is that the ground truth $\nabla_x \log p(x)$ is not known, particularly when we just have samples from some data distribution $\{x\} \sim p_{\mathcal{D}}(x)$. Several tricks have been developed to circumvent this restriction, most notably denoising score matching [Vincent, 2011]. First, we make the observation that the explicit score matching loss can be rewritten implicitly [Hyvarinen, 2005]:

$$\mathcal{L}_{\text{ESM}} = \underbrace{\mathbb{E}_{x \sim p_{\mathcal{D}}(x)} \left[\frac{1}{2} ||s_\theta(x)||^2 + \text{div} \cdot s_\theta(x) \right]}_{\mathcal{L}_{\text{ISM}}} + \text{constant} \tag{4.3}$$

Technically, we can stop here as $\mathcal{L}_{\text{ISM}}$ is now tractable. However note that we take the divergence of our score function model $s_\theta(x)$. When doing stochastic gradient descent against samples from the dataset $x \sim \mathcal{D}$, the divergence term introduces second-order derivatives of the score function. This limits how complex $s_\theta(x)$ can be.

Note also that in the finite-sample setting we have a limited number of samples in our dataset $\mathcal{D}$. If we were to optimize a stochastic version of $\mathcal{L}_{\text{ISM}}$, we might want to smooth out the "lumpiness" of the sampled data. This motivates passing to the explicit score matching of our score model $s_\theta(x)$ with a non-parametric Parzen window density estimator, which applies additional smoothing to the empirical data distribution $p_{\mathcal{D}}(x)$. i.e.

$$\mathcal{L}_{\text{ESM},q_\sigma} = \mathbb{E}_{q_\sigma(x')} \left[\frac{1}{2} ||s_\theta(x') - \nabla_{x'} \log q_\sigma(x')||^2 \right]$$

where $q_\sigma(x)$ is a Gaussian kernel density estimator with bandwidth σ.

We then take further inspiration from the denoising autoencoder literature. Instead of explicitly matching the score function where we convolve the empirical data distribution with a Gaussian kernel, what if we instead required the score function to act as a denoiser, i.e. it would map the noisy sample x' back to the clean sample x? Let the transition kernel from x to x' be $q_\sigma(x'|x)$, typically Gaussian:

$$q(x'|x) = N(x'; x, \sigma^2 I)$$

We can imagine modifying the density $p_{\mathcal{D}}(x)$ to instead consider the joint density $p_\sigma(x, x') = q_\sigma(x'|x)p_{\mathcal{D}}(x)$. We now define the *denoising score matching* objective:

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{(x,x')\sim p_\sigma(x,x')}\left[\frac{1}{2}||s_\theta(x') - \nabla_{x'} q_\sigma(x'|x)||^2\right] \tag{4.4}$$

The idea here being that the score at some noisy point should push us towards the clean point. In fact this is equivalent to $\mathcal{L}_{\text{ESM},q_\sigma}$!

This final score matching loss is the workhorse powering modern diffusion models today. Specifically, with the Gaussian kernel

$$q_\sigma(x'|x) = N(x'; x, \sigma^2 I)$$

we have

$$\nabla_{x'} \log q_\sigma(x'|x) = \frac{x - x'}{\sigma^2}.$$

Hence the interpretation "denoising": if we take a step along the score function, we take out some of the noise that was added to the sample.

Denoising score matching and Denoising diffusion probabilistic models The precursors to modern diffusion models all settled on the idea of progressively adding noise to data.

The earliest work that applies diffusion physics to probabilistic generative models, [Sohl-Dickstein et al., 2015] proposed an approach where one constructs a Markov chain of latent variables $x_0, x_1, \ldots, x_T$ where x_T is pure Gaussian noise, and x_0 is the data distribution. The forward diffusion process resembles a discretized Langevin SDE wherein we have the transition kernel

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$$

where α_t is a function of t. α_t is referred to as the "variance schedule". The forward trajectory, starting with a sample x_0 from the data distribution $p_0(x)$ and ending with a sample x_T drawn from a simpler distribution $p_T(x) \approx N(x; 0, I_d)$, is obtained after T iterations of the transition kernel:

$$p(x_T) = p_0(x_0) \prod_{t=1}^{T} q(x_t|x_{t-1})$$

The reverse process has the exact same functional form as the forward process even with Gaussian transition kernels [Feller, 2015]. However, their mean and covariances must be learned from data. [Sohl-Dickstein et al., 2015] parametrize each transition kernel $p_\theta(x_t|x_{t-1})$ with a neural network. The model is trained via a variational lower bound, i.e.

$$\mathcal{L} = \mathbb{E}_{x_0 \sim p_0(x)} \left[\log p_T(x_0) - \sum_{t=1}^{T} \mathbb{E}_{x_{t-1} \sim p_\theta(x_{t-1}|x_t)} \left[\log p_\theta(x_t|x_{t-1}) \right] \right]$$

Despite the initial success of [Sohl-Dickstein et al., 2015], the model was limited in its expressivity. It took several years for tweaks to be made to enable diffusion models to generate high-quality samples for high-dimensional data. [Ho et al., 2020] operated with a similar framework but made the realization that one could reparameterize the reverse process:

instead of modeling the means of the transition kernels, one could also model the noise ϵ. Sampling from the reverse process then resembled Langevin dynamics.

[Song and Ermon, 2020] propose a Noise Conditional Score Network (NCSN) which seeks to learn the score function $s_\theta(x, \sigma)$ from minimizing the denoising score matching objectives:

$$\theta^* = \text{argmin}_\theta \mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[\mathbb{E}_{\sigma \sim q(\sigma)} \left[\| s_\theta(x, \sigma) - \nabla_x \log p_\sigma(x) \|^2 \right] \right]$$

Once the score function model is learned, one can sample from the denoised distributions $p_\sigma(x)$ by running Langevin dynamics to get samples from the reverse process:

$$x_t = x_{t-1} + \frac{\sigma^2}{2} \nabla_x \log p_\sigma(x_{t-1}) + \sqrt{\sigma} z_t$$

where $z_t \sim N(0, I_d)$.

SDE and ODE diffusion Subsequent work generalized this discrete-time case to a continuous-time setting. In the continuous case, the data is described by a stochastic differential equation (SDE) or its deterministic counterpart, an ordinary differential equation (ODE), for modeling the forward and reverse processes.

[Song et al., 2021] introduced a general framework for diffusion models by modeling the forward noising process as a continuous-time SDE. Specifically, they start with the Ito SDE:

$$dx = f(x, t)dt + g(t)dw$$

where w is the standard Wiener process, $f(x, t)$ is the drift term, and $g(t)$ is the diffusion coefficient. The reverse-time SDE can be written as [Anderson, 1982]:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)d\bar{w}$$

where dt is a negative time increment, $d\bar{w}$ is a reverse-time Wiener process, and $\nabla_x \log p_t(x)$ is the score function of the time-varying density.

It is also possible to do this in a deterministic fashion. Instead of an SDE, Song et al 2021 also propose an ODE formulation:

$$dx = [f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)]dt$$

The deterministic ODE formulation is particularly useful for modeling the reverse-time process, as it is easier to integrate and does not require the addition of noise. However, it can be shown that the stochasticity in the SDE approach can lead to more robust results and better convergence [Nie et al., 2024].

Diffusion vs autoregression and the unreasonable effectiveness of diffusion models for image data One might be tempted to suggest that diffusion models might be the end of the line for generative modeling. Diffusion models have have tremendous success in image modeling in particular. The forward process erases the original image in a progressive fashion. While the noise level remains small, large-scale features of the image are still preserved and readily identifiable (Figure 4.1).

[Rissanen et al., 2023, Dieleman, 2024] suggest that diffusion models implicitly model images as a coarse-to-fine process. When adding the isotropic Gaussian noise that is universally used, combined with the fact that naturally-occurring images have a $1/f^{\alpha}$-type spectral density, one can see that when following the reverse process, diffusion models successively generate features from coarse to fine scale (Figure 4.2).

The success of diffusion modeling in image data may be attributed in large part to this fact. Explicitly taking this into account has provided alternative and fruitful directions for research in generative modeling. [Guth et al., 2022] accelerate score-based modeling by applying diffusion to modeling the wavelet coefficients instead of raw image data, which

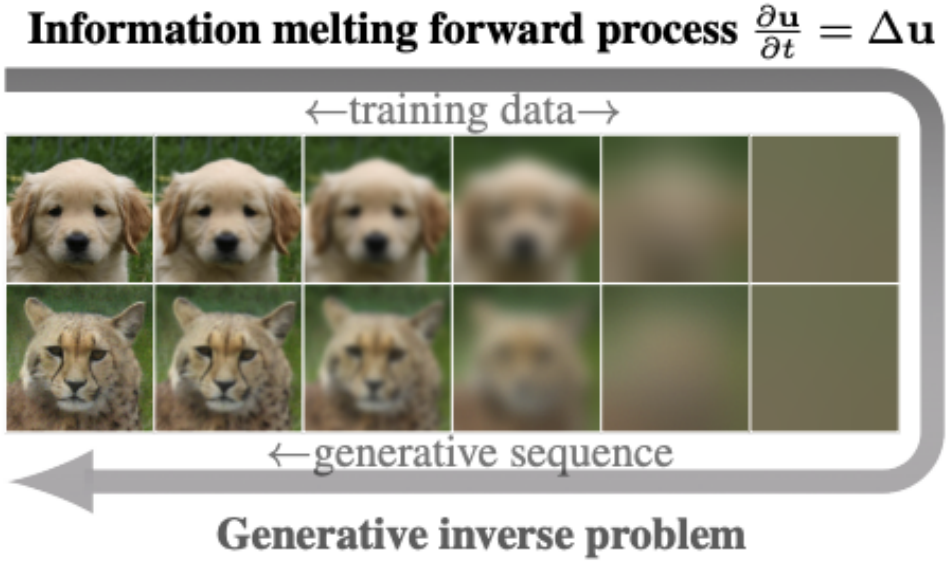

Figure 4.1: Forward and reverse process examples (from [Rissanen et al., 2023]). We can see that at the initial stages of the forward process nearly all the structure remains identifiable. At small times we see that the high-frequency structure starting to blur, and at large times even the low-frequency structure starts to blur.

they show to be a more natural and well-conditioned procedure requiring fewer time steps. [Rissanen et al., 2023] explicitly apply the heat equation to images and show how to invert the heat dissipation process.

Statistical efficiency The practice of diffusion modeling ran on well ahead of the theory. Using score matching as a training method to fit a density is well-principled as the estimator is known to be consistent. However, little was known in what circumstances we could expect score matching to perform similarly to that of maximum likelihood, whether it was "statistically efficient" in the sense of being asymptotically equivalent to maximum likelihood, and if not, how much worse.

In the early works on diffusion, multimodality and low-dimensional manifold structure were conjectured as sources of difficulty for score matching, leading to the idea of annealing the density towards a more well-behaved Gaussian by convolving the data distribution with

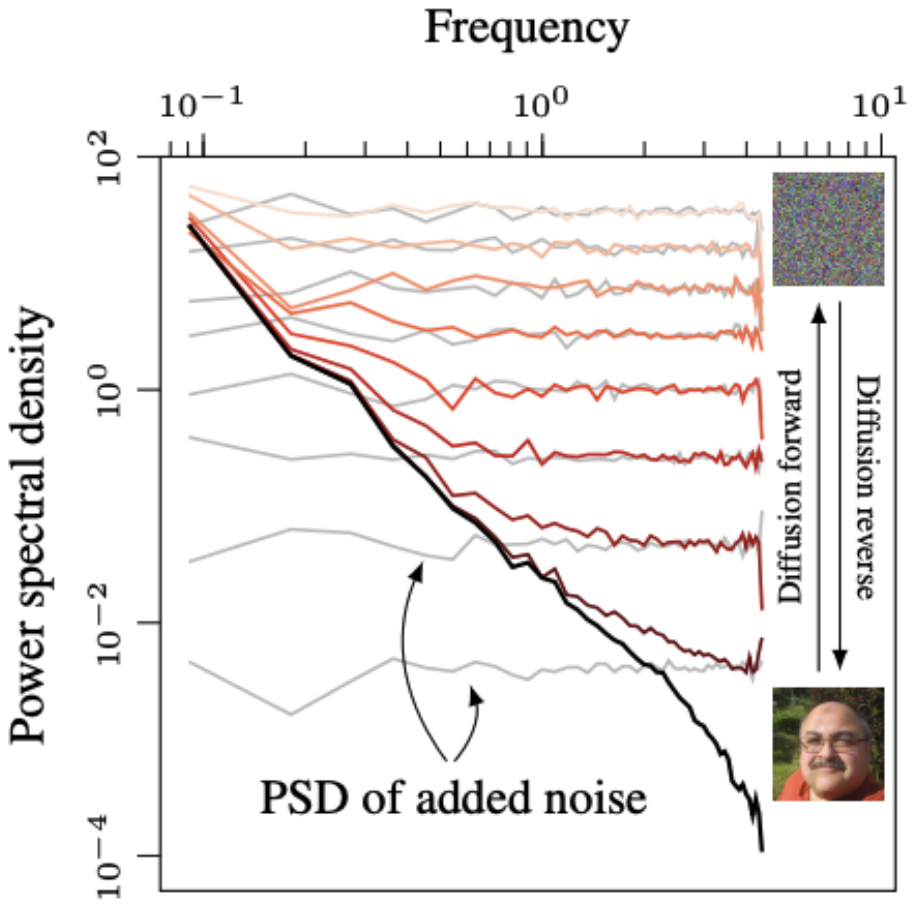

Figure 4.2: Spectral density for images and the result of adding isotropic Gaussian noise at increasing scales (from [Rissanen et al., 2023]).

Gaussians of increasing variance. Because regions of the density with low probability will necessarily have fewer samples to estimate the score with, it is necessary to anneal the density slowly towards one that is more diffuse and easier to learn.

Recent theoretical work [Koehler et al., 2022] has formalized these intuitions, demonstrating that the statistical efficiency of score matching depends on the properties of the underlying distribution, particularly the isoperimetric properties (Poincare, log-Sobolev, and isoperimetric constants), which characterize the mixing time of Langevin dynamics. In particular, multimodal distributions with well-separated modes have very large such constants, which increase the mixing time. In these settings, score matching can easily be much less efficient than maximum likelihood.

Despite this loss of efficiency in theory, score matching is often the only practical solution for very high-dimensional problems. Theory has not yet extended to cover the case of the annealing strategies prominently featured in modern diffusion techniques. Future work may yet uncover interesting connections between the behavior and mixing times of high-dimensional Langevin dynamics and the practical reality of diffusion models.

Common themes

We can distill the core ideas of basic diffusion modeling into a few key points:

1. Pick an easy-to-sample from target distribution The standard here is to select the standard Gaussian, as it is the easiest distribution to sample from. It is isotropic, has a simple density, and is the stationary distribution of the heat equation.

2. Transform your data distribution into the target as smoothly and as simply as possible. Once we've picked the target distribution, we need to find a way to transform our data distribution into the target. Because we've previously selected the standard Gaussian, we can use the Langevin SDE to do this. This naturally admits a closed-form solution for

the trajectory of any particle $x_0 \to x_T$.

3. Use a function approximator to learn the score function. Now that we can draw samples from the density $p_t(x)$, we can use a neat trick to estimate the score function. We will train a function approximator (typically a neural network) to learn the score function, which is the gradient of the log-density:

$$s(x, t) := \nabla_x \log p_t(x)$$

4. Use numerical integration to reverse the process and generate samples. Once this is complete, we can generate samples from the target distribution x_T and numerically integrate the reverse Langevin SDE using our learned score function. If $s_\theta(x, t)$ is "close" to the true score function, we expect the reverse process to generate samples that are close to the original data distribution.

A more complete taxonomy of possible improvements that can be made to this standard diffusion modeling process is shown in Figure 4.3.

We can see that improvements to the target distribution and the forward diffusion process would be classified as diffusion process design, whereas improvements to the function approximation fitting step would be likelihood optimization, and anything related to avoiding the hefty computational cost of numerically integrating the reverse Langevin SDE would be classified as sampling acceleration.

We will not review all of these improvements in this chapter, but we will focus on a few key areas where we believe improvements can be made. Here are a few natural questions that arose in the course of our investigations:

- Can we build an intuition for what the score function does, what it means? Can we do this in simple cases and more complex cases?
- If the data is generated from some energy-based model (Gibbs density) with structure

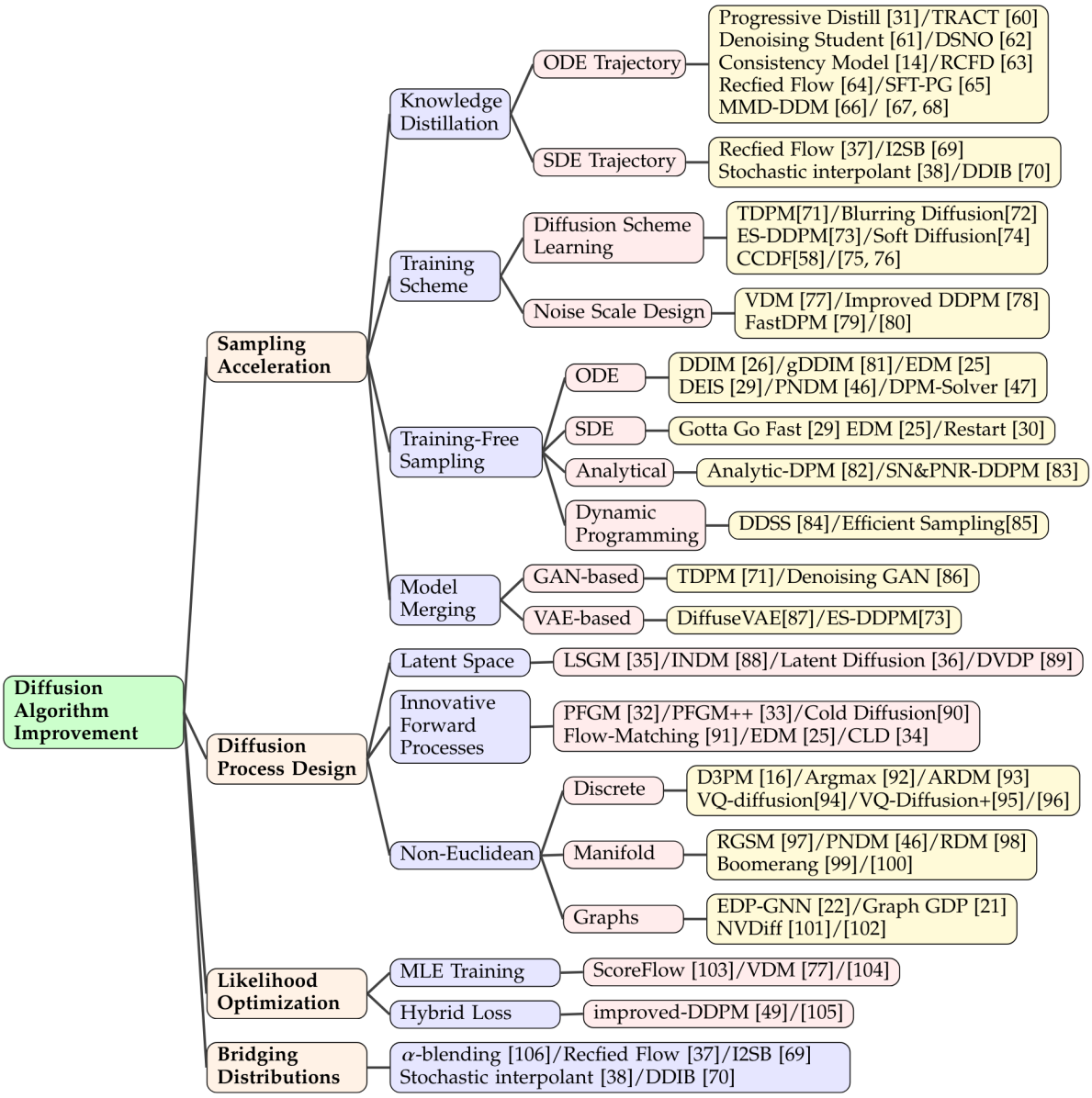

Figure 4.3: A taxonomy of improvements to diffusion models (sourced from "A Survey on Generative Diffusion Models", [Cao et al., 2023])

in the terms of $E(x)$, can we recover that from the samples? Does that show up in the score function itself?

4.3 Methods

In order to answer some of the previously posed questions, we will need to outline a few methods. Out of the zoo of generative diffusion processes we select a particular one that we use primarily. In addition, we also outline our modeling approaches to approximating the score function $s_\theta(x, t)$.

4.3.1 Standard diffusion

Diffusion SDE formulation

We proceed according to the SDE framework per [Song et al., 2021]. Consider the Ito SDE comprised of a drift vector and a scalar diffusion coefficient on the Brownian motion:

$$dx = f(x,t)dt + g(t)dW, t \in [0, \infty) \tag{4.5}$$

where $x \in \mathbb{R}^d$, $f(x,t) : \mathbb{R}^d \times [0,\infty) \to \mathbb{R}^d$, $g(t) : [0,\infty) \to \mathbb{R}_{\geq 0}$, and W_t is the standard Wiener process. If $f(x,t)$ and $g(t)$ are piecewise-continuous then the forward SDE has a unique solution [Oksendal, 1992]. It is common to truncate time to a maximum value, and we do the same, letting $t \in [0, T]$.

There are three types of forward processes commonly examined, variance preserving (VP),

variance exploding (VE), and sub-variance preserving (sub-VP):

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw \tag{VP}$$

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)(1 - \exp(-2\int_0^t \beta(s)ds)}dw \tag{sub-VP}$$

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}dw \tag{VE}$$

where $\sigma^2(t)$ is a monotonically-increasing variance function with $\sigma^2(0) = 0$. Note that the overdamped Langevin SDE (Equation 4.1) that we will discuss in some detail later corresponds to VP-SDE with $\beta(t) = 2$. The names for these SDEs derive from the fact that the variance of the VE-SDE is unbounded as $t \to \infty$ due to the monotonic increase in σ^2, whereas it can be shown that the VP-SDE always has bounded variance, and that the sub-VP SDE has variance that is upper-bounded by the corresponding VP-SDE with the same $\beta(t)$ [Song et al., 2021].

Let $p_t(x)$ denote the marginal distribution of the process at time t, $p_0(x)$ the data distribution, and $p_T(x)$ the prior/source distribution. The reverse-time SDE that corresponds to the forward process has the form [Anderson, 1982]:

$$dx = [f(x,t) - g(t)^2\nabla_x \log p_t(x)]d\bar{t} + g(t)d\bar{W}.$$

Given a good estimate for the score function $s(x,t) = \nabla_x \log p_t(x)$, we can numerically integrate this reverse-time SDE to generate samples from the data distribution $p_0(x)$.

To fit the score function, we use the denoising score matching loss

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{t\sim U(0,T)}\left[\lambda(t)\mathbb{E}_{x_0\sim p_0}\mathbb{E}_{x_t\sim q(x_t|x_0)}\left(||s_\theta(x,t) - \nabla_{x_t}\log q(x_t|x_0)||_2^2\right)\right]$$

where t is uniformly sampled from $[0, T]$, x_0 is drawn from the data distribution, and x_t is drawn from the marginal distribution as $q(x_t|x_0)$ is the Gaussian kernel associated with the forward SDE. $\lambda(t)$ is a weighting function designed to maintain the same loss magnitude at various points in time, typically of the form

$$\lambda(t) \propto \mathbb{E}[||\nabla_{x_t} \log q(x_t|x_0)||_2^2]^{-1}.$$

Variance-Preserving SDE

In the remainder of this work, we shall focus on the VP-SDE. We will specialize to the case where the prior distribution is $p_T(x) = N(x; 0, I_d)$. The VP-SDE with $\beta(t) = 2$ is as follows:

$$dx_t = -x_t dt + \sqrt{2} dw_t$$

which has as solution the Ornstein-Uhlenbeck process for $x_0 \sim p_0$

$$x(t) = e^{-t} x_0 + \sqrt{1 - e^{-2t}} z$$

where $z \sim N(0, I_d)$.

When training the score function using $\mathcal{L}_{\text{DSM}}$, we have:

$$\nabla_{x_t} \log q(x_t|x_0) = -\frac{x_t - e^{-t} x_0}{\sqrt{1 - e^{-2t}}} = -z$$

Hence the name denoising score matching: we seek a score function approximation that predicts the noise that was added to the sample.

4.3.2 Models as probes for score function structure

We will use several different model classes for $s_\theta(x,t)$ all employing various ideas including the cluster basis expansion, radial basis functions, and separation of variables. The reason to employ these assumptions, rather than jumping straight to a multi-layer perception or U-net (as commonly seen in practical diffusion modeling work), is that we want these functions to be more interpretable. Our objective is to compare models based on their fitting performance, evaluate their sample complexity, and see whether they can uncover the structure of the problem. This last part requires the use of regularization, in particular group lasso, a generalization of lasso that treats entire groups of variables as terms to be thrown into a L_1 loss.

Cluster basis with radial kernel expansion

In physics, one often makes the assumption that a function of n variables may be well-approximated by a sum of simpler functions that each only mix $k \ll n$ variables. This is guided by the observation that many systems in practice have complex many-body potentials that emerge from the sum of 2-body interactions, such as the potential landscapes formed by classical particles interacting under $1/r^2$-type potentials (gravity, electrostatic) as well as more exotic quantum phenomena.

We refer to this as the *cluster basis* assumption. It forms the basis of one of our workhorse models, used as a score function approximation. The idea is to place radial basis kernels $\exp(-||(x,t) - c||^2)$ at evenly-spaced points throughout space and time. We place the restriction that each kernel can only have two spatial variables and one time variable. Let the score function be a vector-valued function on space and time $s_\theta(x,t) : \mathbb{R}^d \times [0,\infty) \to \mathbb{R}^d$:

$$s_\theta(x,t) = \sum_{i \le j} s_\theta^{(ij)}(x_i, x_j, t) \tag{4.6}$$

with the individual functions in the sum decomposing further as

$$s_\theta^{(ij)}(x_i, x_j, t) = \vec{\theta}^{(ij)} \cdot \Phi(x_i, x_j, t) \tag{4.7}$$

$$= \sum_k \theta_k^{(ij)} \exp(-\gamma ||(x_i, x_j, t) - c_k||_2^2) \tag{4.8}$$

where γ is a fixed inverse variance parameter for the radial basis functions and the c_k are chosen to be evenly-spaced points in a compact subset of $\mathbb{R}^2 \times [0, \infty)$, e.g. $[-L, L]^2 \times [0, T]$ where L, T is chosen large enough to contain the support of the score function we want to estimate.

Product radial kernel expansion

Inspired by the closed-form formula for the score function of a correlated Gaussian diffusing to a standard Gaussian (Section 4.4.1), we make the assumption that the score function is a sum of functions that separate in the spatial and time variables:

$$s_\theta(x, t) = \sum a(x)b(t).$$

For modeling the spatial dependence, we make use of the cluster basis ideas outlined earlier. We have:

$$s_\theta(x, t) = \begin{pmatrix} s_1(x, t) \\ \vdots \\ s_d(x, t) \end{pmatrix}$$

Each component is a sum of functions that depend only on two and three variables, with the maximum order of dependence in spatial variables being two (hence 2-cluster):

$$s_k(x, t) = \sum_i s_{ik}(x_i, t) + \sum_{i<j} s_{ijk}(x_i, x_j, t) \tag{4.9}$$

We assume functional forms that are sums of products that where the spatial and time variables completely separate:

$$s_{ik}(x,t) = \vec{\theta}_{ik} \cdot \Phi(x,t) = \sum_{\ell m} \theta_{\ell m}^{(ik)} a_\ell(x) b_m(t) \tag{4.10}$$

$$s_{ijk}(x,y,t) = \vec{\theta}_{ijk} \cdot \Phi(x,y,t) = \sum_{\ell m} \theta_{\ell m}^{(ijk)} a_\ell(x,y) b_m(t) \tag{4.11}$$

Here $\vec{\theta}_{ik}, \vec{\theta}_{ijk}$ denote our variable weights to be fit (that are then indexed into with ℓ, m), $\Phi(x,t), \Phi(x,y,t)$ denote a basis function vector that is formed by taking the outer product of a basis in spatial variables and a basis in time.

In this work we use radial basis functions as our basis in space and exponentially-decaying basis functions as our basis in time:

$$a_\ell(x) = \exp(-\lambda ||x - c_\ell||^2)$$

$$a_\ell(x,y) = \exp(-\lambda ||(x,y) - \vec{c}_\ell||^2)$$

$$b_m(t) = \exp(-\gamma_m |t|)$$

Note that ℓ indexes over a predefined grid of evenly-spaced points $c_\ell \in \mathbb{R}, \vec{c}_\ell \in \mathbb{R}^2$ that covers the region where we want to model the score function and ℓ indexes over a set of real decay parameters γ_m that we want to fit to data as well.

The parameters that are to be fit include $\vec{\theta}_{ik}, \vec{\theta}_{ijk}$ (the weights on the product terms) and the decay parameters γ_m.

Fitting

In our numerical work, all the models are fit via stochastic gradient descent on $\mathcal{L}_{\text{DSM}}$. When relevant, the learning rate and regularization weights are provided. As we utilize the VP-SDE

and $\mathbb{E}_{x_t|x_0}[||\nabla_{x_t} \log q(x_t|x_0)||_2^2] \sim O(1)$ is constant in time, we may set $\lambda(t) = 1$.

Group lasso

In addition to the denoising score-matching loss $\mathcal{L}_{\text{DSM}}$, we also add a group lasso regularization term to encourage sparsity on the weights, but treating "groups" of parameters as individual terms to encourage to be zero. The idea behind group lasso is simple. Suppose we have $\theta \in \mathbb{R}^w$ as a parameter vector. Let $G = (g_1, \ldots, g_m)$ be a mutually-exclusive partitioning of the indices, i.e. $g_i \subseteq [1, \ldots, w]$, $\cup_{i=1}^m g_i = [1, \ldots, w]$ and $g_i \cap g_j = \varnothing$ for all $i \neq j$.

These partitions typically correspond to semantically meaningful groupings of the variables. For example, $\vec{\theta}_{ij}$ from Equation 4.7 and $\vec{\theta}_{ik}, \vec{\theta}_{ijk}$ from Equations 4.10 and 4.11, corresponding to the weights on individual 2-cluster functions, are natural groups. We might start with a model that has terms corresponding to all $\binom{d}{2}$ pairs of variables x_i, x_j but if the score function has 2-cluster structure we would want to encourage the fitting process to discover that.

The lasso penalty [Tibshirani, 1996] is famous for encouraging sparsity by applying a L_1 norm to the parameters, i.e.

$$\mathcal{L}_{\text{Lasso}} = ||\theta||_1 = \sum_{i=1}^{w} |\theta_i|$$

The group lasso [Yuan and Lin, 2006, Jacob et al., 2009, Mao, 2020] instead treats the L_2 norm of a parameter group $||\theta_{g_i}||_2$ as a quantity that goes inside the absolute value, so we write

$$\mathcal{L}_{\text{Group lasso}} = ||\theta||_G = \sum_{i=1}^{w} \sqrt{\ell_i} ||\theta_{g_i}||_2$$

where $\theta_{g_i} = \{\theta_k : k \in g_i\}$ is the parameter vector for partition g_i and ℓ_i is the length of θ_{g_i}.

In our case, the resulting loss looks like:

$$\mathcal{L} = \mathcal{L}_{\text{DSM}} + \lambda \mathcal{L}_{\text{Group lasso}}. \tag{4.12}$$

4.4 Vignettes

Now, we will examine a few of the common themes discussed in the previous section on diffusion models, take a look at a few simple cases, and try relaxing certain assumptions in the hopes that we will arrive at a deeper understanding of how diffusion models work and how they may be improved.

4.4.1 Score function structure in simple cases

First, we examine the score function for a delta function, which gives us a concrete example of how scores behave for highly concentrated distributions. Next, we analyze the score function for a correlated Gaussian distribution and discuss how conditioning and matrix inversion play a role in learning the score. Lastly, we will touch upon sample complexity considerations, demonstrating how the difficulty of learning the score function can change with time during the diffusion process. These examples help highlight the mathematical structure of score functions and prepare us for tackling more complex cases in generative models.

Delta function

In the simplest case, consider the data distribution to be a delta function concentrated at some point $c \in \mathbb{R}^d$, i.e. $p_0(x) = \delta(x - c)$. We aim to diffuse it towards the standard Gaussian $p_T(x) = N(x; 0, I_d)$. With Langevin dynamics, we have trajectories:

$$x_t = e^{-t}c + \sqrt{1 - e^{-2t}}z$$

with $z \sim N(0, I_d)$. The explicit density at intermediate times is:

$$p_t(x) = N(x; e^{-t}c, (1 - e^{-2t})I_d)$$

and thus the score function is:

$$s(x,t) = \nabla_x \log p_t(x) = -(1-e^{-2t})^{-1}(x - e^{-t}c)$$

The score function points in the direction of increasing density by definition. For small t, the score always points away from x towards c, and for large t, the score points towards the origin. During the reverse process, following the score will transport us towards c. However, note that as $t \to 0$ the score blows up. This ill-conditioning makes it difficult to model the score near $t = 0$.

Diffusing a correlated Gaussian to standard Gaussian

Let's look at the case where we start with data drawn from a correlated Gaussian and we want to distribute it to a standard Gaussian. Of course, this is not useful for generative purposes as we know we may directly draw samples from a correlated Gaussian, but as a simple problem it still provides us insight for how more complex distributions may behave.

Start with the data distribution ($x \in \mathbb{R}^d$):

$$p_0(x) = N(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$

We want to diffuse it towards the standard Gaussian $N(x;0,I_d)$ as $t \to \infty$. With the Langevin dynamics from before (Eq. **??**), we have

$$x_t = e^{-t}x_0 + \sqrt{1-e^{-2t}}z.$$

Therefore we have the explicit density for all time t

$$p_t(x) = N(x; e^{-t}\mu, e^{-2t}\Sigma + (1-e^{-2t})I_d)$$

and the score function

$$s(x,t) = \nabla_x \log p_t(x) = -(e^{-2t}\Sigma + (1-e^{-2t})I_d)^{-1}(x - e^{-t}\mu) = -K(t)^{-1}(x - e^{-t}\mu).$$

Letting $K(t) = e^{-2t}\Sigma + (1-e^{-2t})I_d$, we can see that the score function is a time-varying linear function in x requiring the inversion of a matrix.

Suppose that our goal is to generate samples via reverse diffusion. We use some score function model $s_\theta(x,t)$ and minimize the denoising score-matching loss

$$\begin{aligned}
\theta^* &= \text{argmin}_\theta \mathcal{L}_{\text{DSM}}(\theta) \\
&= \mathbb{E}_{x_t \sim p_t(x)} \left[||s_\theta(x,t) - \tilde{s}(x_t,t)||_2^2 \right] \\
&= \mathbb{E}_{x_0 \sim p_0(x), x_t \sim p_t(x)} \left[\left|\left| s_\theta(x,t) + \left(\frac{x_t - e^{-t}x_0}{\sqrt{1-e^{-2t}}} \right) \right|\right|_2^2 \right] \\
&= \mathbb{E}_{x \sim p_0(x)} \left[||s_\theta(x,t) + z||_2^2 \right]
\end{aligned}$$

We can already see from this objective alone that the target score function at some point $x_t = e^{-t}x_0 + \sqrt{1-e^{-2t}}z$ for $x_0 \sim p_0$ and $z \sim N(x; 0, I_d)$

$$\tilde{s}(x_t,t) = -z = \frac{x_t - e^{-t}x_0}{\sqrt{1-e^{-2t}}}$$

blows up as $t \to 0$. We have a singularity at $t = 0$ just as in the previous delta function case. This suggests that we cannot discretize time too finely near $t = 0$ when implementing this numerically.

Conditioning as $t \to \infty$ Let λ_i be the eigenvalues of the data covariance Σ. The corresponding eigenvalues of $K(t)$ are

$$\lambda_i(K(t)) = e^{-2t}(\lambda_i - 1) + 1$$

and the condition number of $K(t)$ is

$$\kappa(K(t)) = \frac{\max_i[e^{-2t}(\lambda_i - 1) + 1]}{\min_i[e^{-2t}(\lambda_i - 1) + 1]}$$

As we let $t \to \infty$, we have $\kappa(K(t)) \to 1$, since I_d is perfectly conditioned.

Note additionally that the mapping $\lambda_i(\Sigma) \to \lambda_i(K(t))$ is affine and therefore must preserve the ordering of the eigenvalues for all t. Letting λ_1 be the largest eigenvalue and λ_d the smallest eigenvalue of Σ, we may write

$$\kappa(K(t)) = \frac{e^{-2t}(\lambda_1 - 1) + 1}{e^{-2t}(\lambda_d - 1) + 1}$$

We have

$$\frac{d\kappa}{dt} = 2e^{2t}\frac{\lambda_d - \lambda_1}{(e^{2t} + \lambda_d - 1)^2} \leq 0$$

as $\lambda_1 \geq \lambda_d$. Thus, the condition number monotonically approaches 1 from above.

In this simple scenario, we observe that the true score function is a linear map that requires a matrix $K(t)$ to be inverted. This matrix $K(t)$ has conditioning that monotonically improves as $t \to \infty$. The score function is "easiest to learn" when the data is perfectly uncorrelated.

Sample complexity "Hard to learn" in this setting refers primarily to the sample complexity, the number of data points one would need to sample from an oracle generating samples $x_t \sim p_t(x)$. Suppose such an oracle existed and our goal was to model the full trajectory of $p_t(x)$. One approach would be to go to $t = 0$ and estimate $K(0) = \Sigma$ to an error $||\hat{\Sigma} - \Sigma||_F \leq \varepsilon$. From classical multivariate statistics [Anderson, 2003] we have that the sample covariance matrix follows a Wishart distribution $\hat{\Sigma}_n \sim W_p(\Sigma, n - 1)$ where n is the number of samples we use to estimate

$$\hat{\Sigma}_n = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$$

We know that

$$\mathbb{E}\left[\left|\left|\hat{\Sigma}_n - \Sigma\right|\right|_F^2\right] \leq C\frac{||\Sigma||_F^2}{n}$$

where C is a constant depending on the dimension d. Therefore to estimate $\hat{\Sigma}$ to within ε Frobenius norm requires $O(||\Sigma||_F/\sqrt{n})$ samples.

But can we do better by picking any other time t? Well, at time t we have

$$p_t(x) = N(x; e^{-t}\mu, e^{-2t}\Sigma + (1 - e^{-2t})I_d)$$

and so estimating $K(t)$ would require $O(||K(t)||_F/\sqrt{n})$ samples. Note that we have

$$||K(t)||_F = \sqrt{\sum_{i=1}^{d} \lambda_i(K(t))^2}$$

With $\lambda_i(K(t) = e^{-2t}(\lambda_i - 1) + 1$ we can also see that $||K(t)||_F$ is monotonically decreasing over time. So we actually don't want to go to $t = 0$ to do the estimation! This suggests that we may be better off selecting a large time, estimating the covariance $K(t)$ there to within error ε, and use the imputed Σ thus derived.

This solution works in this specific instance where we know the form of $p_t(x)$ in advance. However, it's worth noting that due to the discretization error introduced by any numerical scheme involved in the backwards pass, small errors made late in time (early in time for the reverse diffusion) can accumulate as we proceed in reverse time to draw samples.

One mode in two dimensions

Assume $d = 2$. For illustration purposes, suppose we have $x = [x_1; x_2]^T$ and x_1, x_2 are zero mean with correlation ρ, and each have variance σ_1^2, σ_2^2, i.e.

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

The explicit density is $p_t(x) = N(x; 0, \Sigma(t))$ for $t \in [0, \infty)$ where $\Sigma(t)$ is given as

$$\begin{aligned} \Sigma(t) &= e^{-2t}\Sigma + (1 - e^{-2t})I_2 \\ &= \begin{pmatrix} e^{-2t}(\sigma_1^2 - 1) + 1 & e^{-2t}\rho\sigma_1\sigma_2 \\ e^{-2t}\rho\sigma_1\sigma_2 & e^{-2t}(\sigma_2^2 - 1) + 1 \end{pmatrix} \end{aligned}$$

The score function is $s(x, t) = -(\Sigma(t))^{-1}x$ with

$$s(x,t) = -\begin{pmatrix} e^{-2t}(\sigma_1^2 - 1) + 1 & e^{-2t}\rho\sigma_1\sigma_2 \\ e^{-2t}\rho\sigma_1\sigma_2 & e^{-2t}(\sigma_2^2 - 1) + 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Numerics Now that we have a ground-truth score function to compare against, we can proceed to verify our results numerically. All our results moving forward are specialized to the case where $\sigma_1 = 0.5, \sigma_2 = 1, \rho = -0.8$, and the maximum time is $T = 2$. Our first goal is to sanity check the score function. In Figure 4.4 we have plotted samples from $p_0(x) = N(x; 0, \Sigma)$ in blue and samples from $p_T \approx N(0, I_2)$ in orange, with the true score function at those times overlaid in black arrows. We can see that the score function points in the direction of increasing density.

We then perform Euler-Maruyama to reverse samples from p_T to p_0 using the true (oracle)

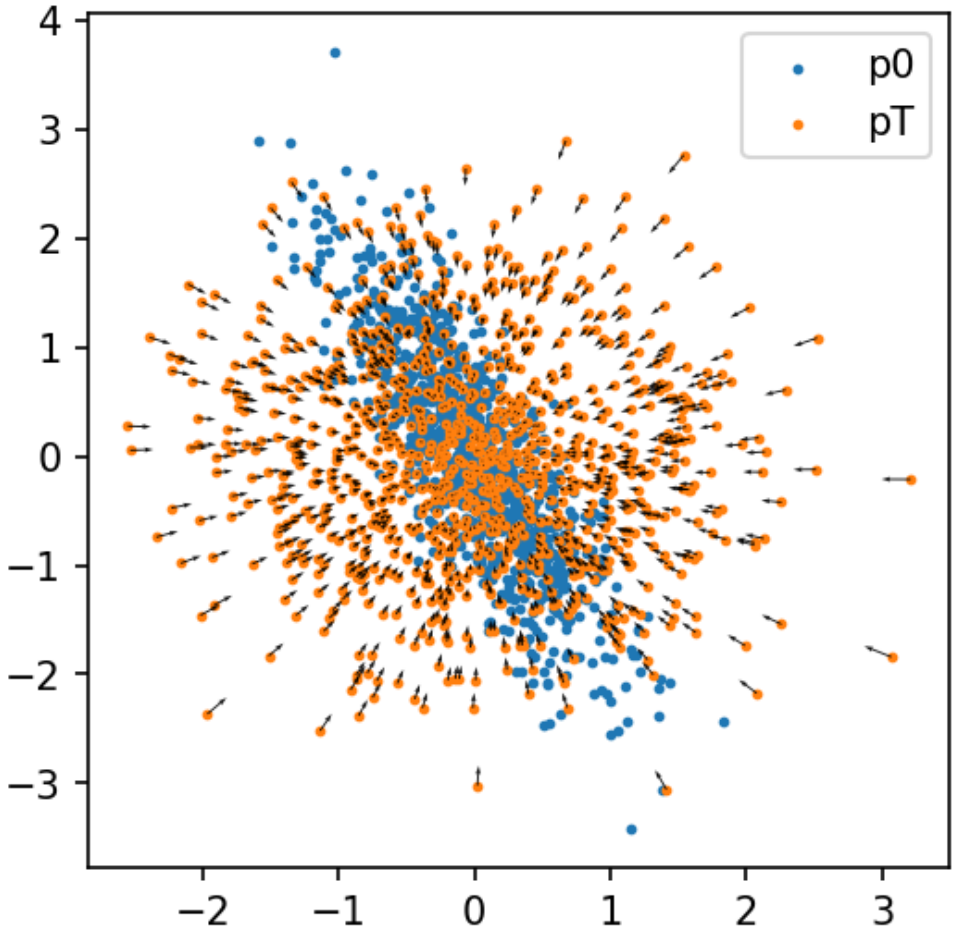

Figure 4.4: Samples from p_0 and p_T where $\sigma_1 = 0.5, \sigma_2 = 1, \rho = -0.8, T = 2$. The true score function $s(x, t)$ directions are plotted in small black arrows.

score function $s(x,t)$ derived above. We use the following metrics to assess for "quality of fit", all deriving from the literature comparing one multivariate Gaussian to another:

- Mahalanobis distance (M-distance): Given samples $\{x\}$ alleged to be from a multivariate normal $N(\mu,\Sigma)$, we compute the distance $D_M(x,N(\mu,\Sigma)) = \sqrt{(x-\mu)^T\Sigma^{-1}(x-\mu)}$. Assuming that the samples are truly drawn from $N(\mu,\Sigma)$, the distribution of distances should follow a χ^2 distribution with d degrees of freedom [Anderson, 2003]. We can then compute p-values or run a χ^2 goodness-of-fit test on the distances to assess for statistical significance.

- Kullback-Leibler divergence (KL divergence): We estimate the sample mean and covariance $\hat{\mu},\hat{\Sigma}$ from our samples and then used the closed-form expression for the KL divergence of two multivariate Gaussians $P = N(\hat{\mu},\hat{\Sigma}), Q = N(\mu,\Sigma)$ has a closed-form solution [Murphy, 2012]:

$$D_{KL}(P||Q) = \frac{1}{2}\left(\mathrm{Tr}(\Sigma^{-1}\hat{\Sigma}) + (\mu-\hat{\mu})^T\Sigma^{-1}(\mu-\hat{\mu}) - d + \log\frac{\det\Sigma}{\det\hat{\Sigma}}\right)$$

- 2-Wasserstein distance (W distance): The Wasserstein distance (earth mover's distance) between two probability distributions can be viewed as the minimum energetic cost of transporting one distribution to the other. The 2-Wasserstein distance for 2 Gaussians conveniently has a closed-form expression [Givens and Shortt, 1984]:

$$W_2^2(P,Q) = ||\hat{\mu}-\mu||^2 + \mathrm{Tr}[\hat{\Sigma}+\Sigma-2\sqrt{\hat{\Sigma}^{1/2}\Sigma\hat{\Sigma}^{1/2}}]$$

We can see in Figure 4.5 the metrics chosen all seem consistent throughout time, with slight errors emerging as $t \to 0$.

Now that we've verified that Euler-Maruyama with the oracle score function suffices to reproduce $p_t(x)$ for all $t \in [0,T]$, we now train a score function approximator $s_\theta(x,t)$. We use

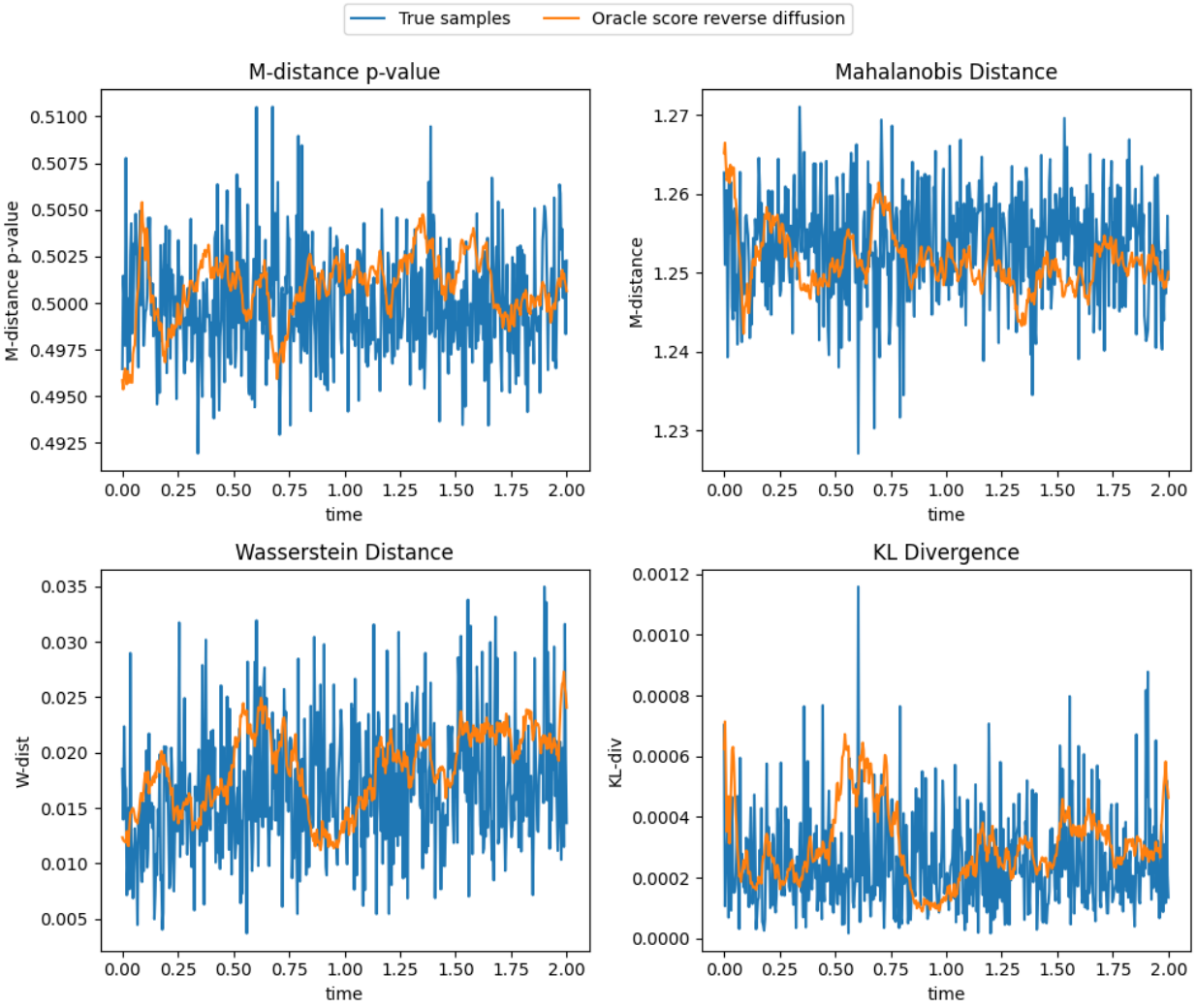

Figure 4.5: Statistics (P-value of the M-distance, M-distance, 2-Wasserstein distance, and KL divergence of the Euler-Maruyama integrated trajectories using the true score function for $\sigma_1 = 0.5, \sigma_2 = 1, \rho = -0.8, T = 2$. $B = 10000$ and $N_T = 500$ so $\Delta t = 0.004$. We can see that the metrics are consistent throughout time with slight errors emerging as $t \to 0$.

the 2-cluster radial basis function estimator in Section 4.3.2 with $N_X = 5, N_T = 10, L = 3$ with N_X, N_T denoting the number of points placed in each x, t dimension and L denoting the maximum extent of the region we generate grid points within, i.e. $c_k \in [-L, L]^2 \times [0, T]$. We use two losses. The first loss is the Fisher divergence (L_2 distance) between $s_\theta(x, t)$ and the oracle score function. The second loss is the denoising score matching loss $\mathcal{L}_{\text{DSM}}$ (Equation 4.4). We use a learning rate of $\eta = 0.1$, no group lasso loss penalty, and batch sizes of $B = 128$ for $N_B = 10^4$ batches.

In Figure 4.6 we show the cosine distances between the learned score function $s_\theta(x, t)$ and the true score $s(x, t)$ for randomly sampled points (t, x_t). When we use the oracle score matching loss, we find much better agreement over all time, but the denoising score matching (which is the only practical loss in actual situations without an oracle score) performs decently well too.

In Figures 4.7 and 4.8 we can see the gap between using the oracle score-matching loss and the denoising score-matching loss. The oracle-trained $s_\theta(x, t)$ very nearly matches the original data distribution p_0, but the denoising score-trained $s_\theta(x, t)$ is much more compressed and only models the variance of the true $p_0(x)$ well in one direction, failing to capture the much larger variation in the orthogonal direction. We see much different behavior in the statistics over time, with errors growing as $t \to 0$.

Interestingly enough, the behavior where our generated samples have isotropic variance that seems to be the minimum of the variances σ_1, σ_2 suggests almost that our minimizing the denoising score matching loss is related to minimizing $D_{KL}(P||Q)$ (where P is the anisotropic $p_0(x)$ and Q is our approximate samples). This is concordant with other literature that connects the score-matching loss to certain variational objectives, like the evidence lower bound (ELBO) in a variational framing of the problem [Sohl-Dickstein et al., 2015, Ho et al., 2020].

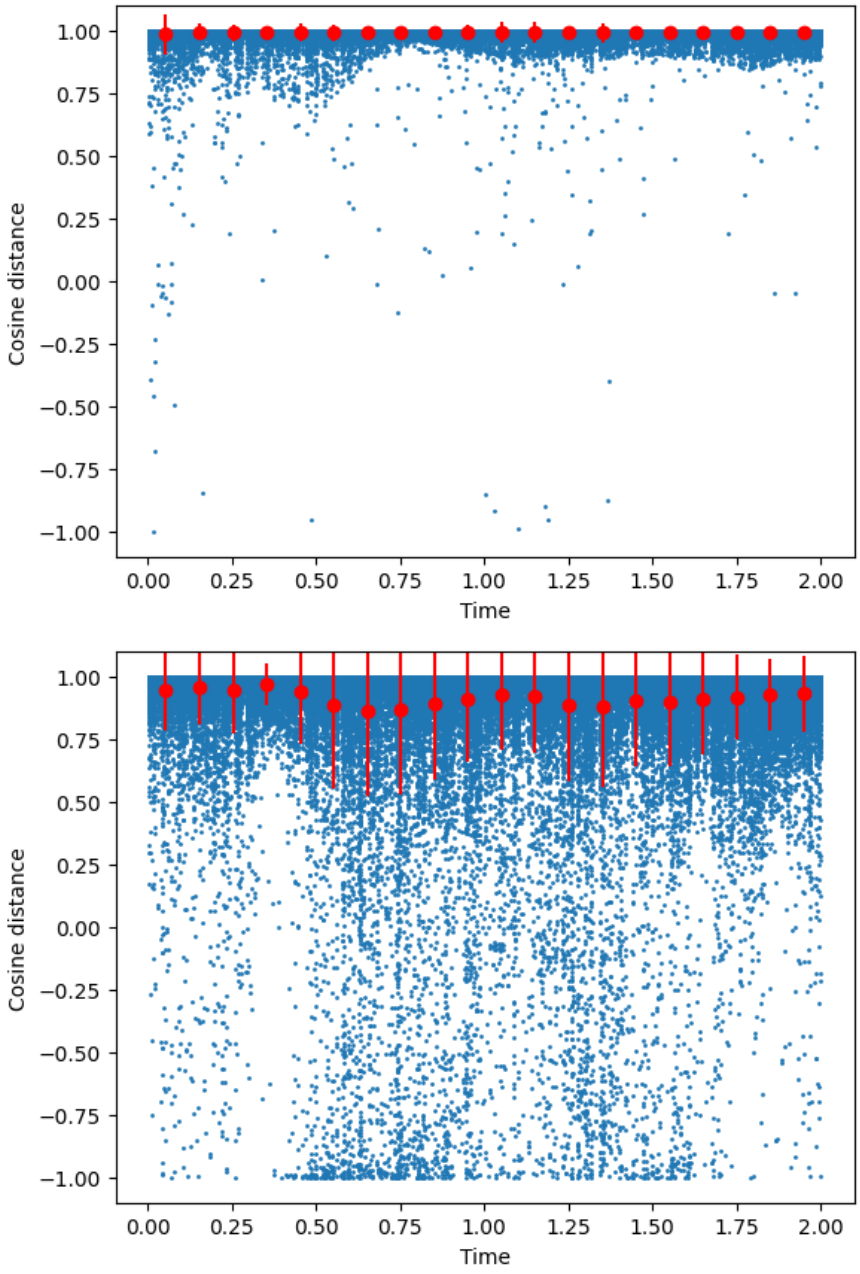

Figure 4.6: Comparison of the result of using the oracle score-matching loss and the denoising score matching loss on the resulting learned score functions $s_\theta(x, t)$. Here we generate various $x(t)$ points and evaluate the cosine distance between the true score function and the learned score function. The blue dots are samples and the red dots are binned means and standard deviations.

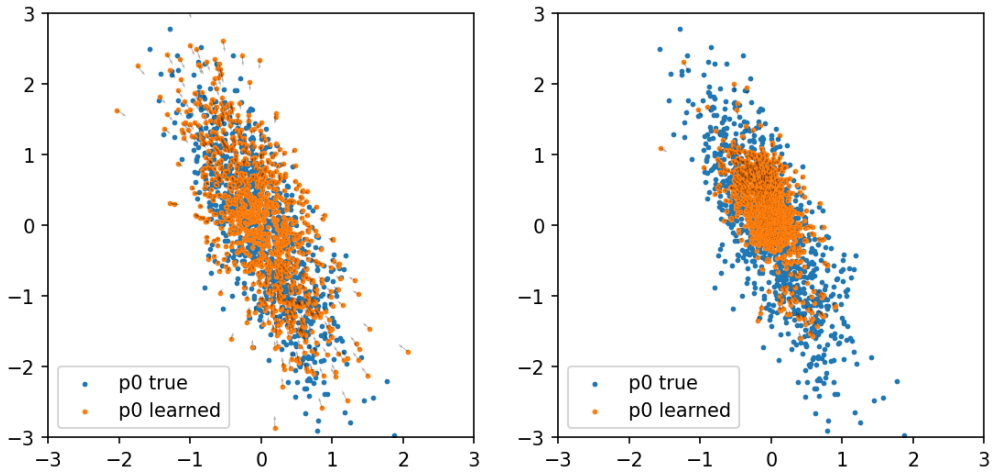

Figure 4.7: Comparison of the samples obtained at $t = 0$. The oracle (explicit) score-matching loss (Equation 4.2) was used in fitting the score function $s_\theta(x, t)$ used for the top figure since the true score function is known here, and $\mathcal{L}_{\text{DSM}}$ (Equation 4.4) was used for the bottom figure. Note how in the top figure the samples we obtain from reverse diffusion (orange) are much more reflective of the true variation of the density in both of its extremal directions, whereas in the bottom figure we can see that the generated samples are much more closely clumped.

4.4.2 Complex cases: multimodal distributions and more dimensions

Mixture of Gaussians

Now we will consider a case where we start with a mixture of Gaussians instead of a single Gaussian. Let $x_0 \sim p_0(x)$ where

$$p_0(x) = \sum_{i=1}^{K} \pi_i N(x; \mu_i, \Sigma_i)$$

and $\sum_{i=1}^{K} \pi_i = 1$. After passing it through the Langevin diffusion we get the trajectory

$$x_t = e^{-t} x_0 + \sqrt{1 - e^{-2t}} z$$

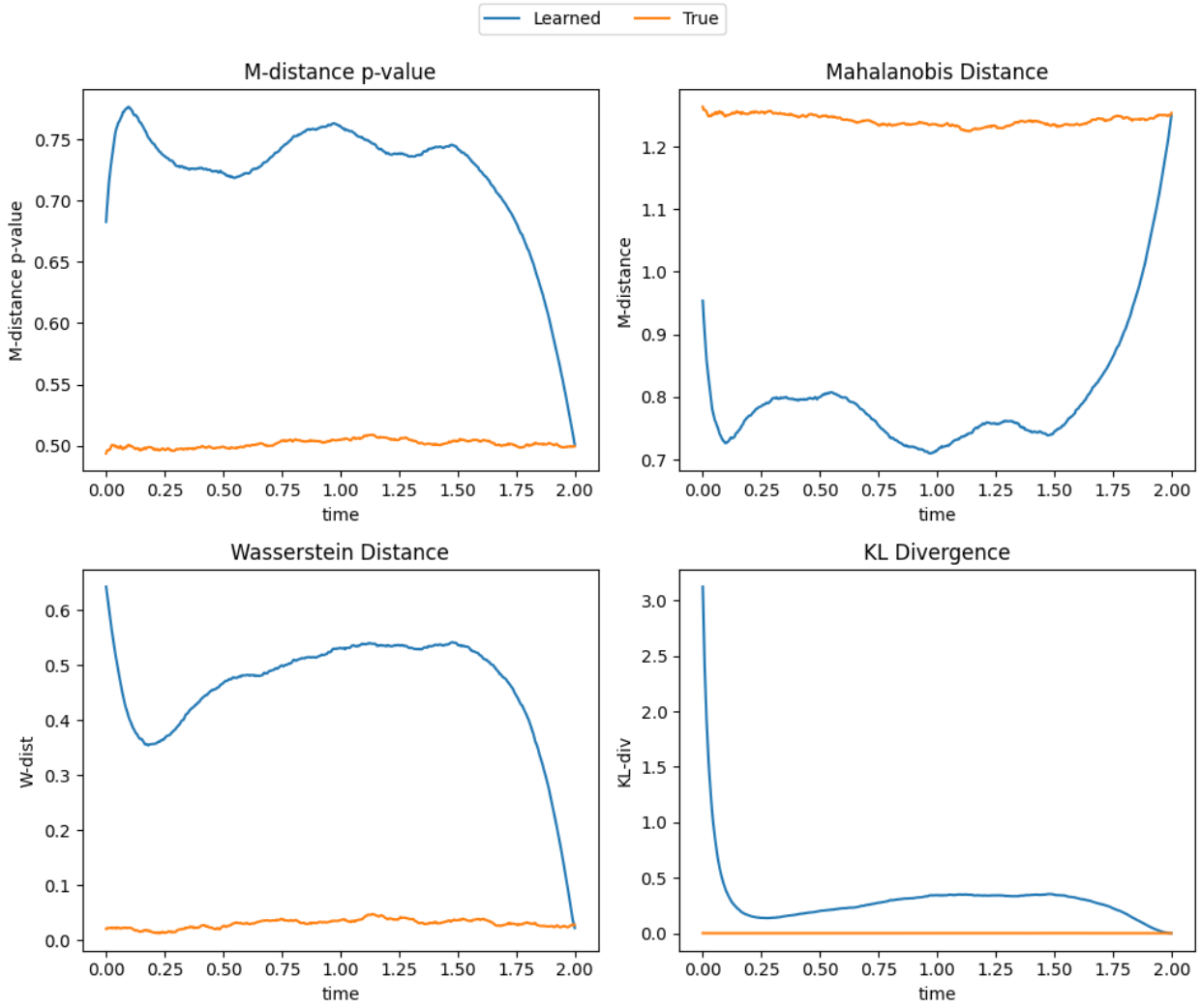

Figure 4.8: Statistics for the Euler-Maruyama reverse-diffused trajectories x_t using the learned score function $s_\theta(x,t)$ (blue) and the true score function $s(x,t)$ (orange). At a given time t, we observe $B = 10000$ samples that should match the marginal density $p_t(x)$ which is known explicitly. We compute the M-distances, the average p-value of the M-distances, the 2-Wasserstein distance (using the empirical $\hat{\mu}, \hat{\Sigma}$ estimated at each time t), and the KL divergence of the Euler-Maruyama integrated trajectories. Here, our parameters were $\sigma_1 = 0.5, \sigma_2 = 1, \rho = -0.8, T = 2, N_T = 500, \Delta t = T/N_T = 0.004$.

Note that this is closed-form and linear! As such, we observe that each component of the mixture transforms in the same way as the single Gaussian case, i.e.

$$p_t(x) = \sum_{i=1}^{K} \pi_i N(x; e^{-t}\mu_i, e^{-2t}\Sigma_i + (1 - e^{-2t})I)$$

In this case, the score function is

$$s(x,t) = \nabla_x \log p_t(x)$$

which can be derived as follows. First, we express the log of the mixture density:

$$\log p_t(x) = \log \left(\sum_{i=1}^{K} \pi_i \mathcal{N} \left(x; e^{-t}\mu_i, e^{-2t}\Sigma_i + (1 - e^{-2t})I \right) \right)$$

Taking the gradient with respect to x, we obtain the score function:

$$s(x,t) = \nabla_x \log p_t(x) = \frac{\nabla_x p_t(x)}{p_t(x)}$$

The gradient of $p_t(x)$ is given by:

$$\nabla_x p_t(x) = \sum_{i=1}^{K} \pi_i \mathcal{N} \left(x; e^{-t}\mu_i, e^{-2t}\Sigma_i + (1 - e^{-2t})I \right) \cdot \left(- \left(e^{-2t}\Sigma_i + (1 - e^{-2t})I \right)^{-1} (x - e^{-t}\mu_i) \right)$$

Thus, the score function can be written as a weighted sum of the score functions for each component, with the weights being the posterior probabilities of each component:

$$s(x,t) = - \sum_{i=1}^{K} \frac{\pi_i \mathcal{N} \left(x; e^{-t}\mu_i, e^{-2t}\Sigma_i + (1 - e^{-2t})I \right)}{p_t(x)} \cdot \left(\left(e^{-2t}\Sigma_i + (1 - e^{-2t})I \right)^{-1} (x - e^{-t}\mu_i) \right)$$

where $p_t(x)$ is the mixture density at time t.

In summary, the score function $s(x,t)$ for the mixture of Gaussians is a weighted sum of

the score functions of each Gaussian component, where the weights are the responsibilities (posterior probabilities) of each component given x.

Two modes in two dimensions

Following the results of the previous section, we'll specialize to the case where we have $k = 2$ modes in 2D:

$$p_0(x) = \pi N(x; \mu_1, \Sigma_1) + (1 - \pi)N(x; \mu_2, \Sigma_2)$$

Let $x \in \mathbb{R}^2$ and choose

$$\mu_1 = \begin{pmatrix} a \\ a \end{pmatrix} \qquad \Sigma_1 = \Sigma$$

$$\mu_2 = \begin{pmatrix} -a \\ -a \end{pmatrix} \qquad \Sigma_2 = \Sigma$$

with

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Results In the results to follow, we fix $a = 1$ and $\sigma_1 = 0.5, \sigma_2 = 1, \rho = -0.8, \pi = 1/3$.

First, we demonstrate that the score function derived earlier indeed behaves as we expect. In Figure 4.9 we can see that the score function arrows (in black) point to the nearest mode early on in the forward process at $t = 0.1$ while the later samples at $t = 0.5$ have scores pointing more towards the origin, as the target (prior) distribution we want to end up with is still the standard Gaussian. Figure 4.11 plots the M-distance and KL-divergence of the samples generated via reverse diffusion using the oracle and the learned score functions.

We see very similar behavior as with the case of one mode in two dimensions (Section 4.4.1). In Figure 4.10, we see the same behavior as before where when we switch from the

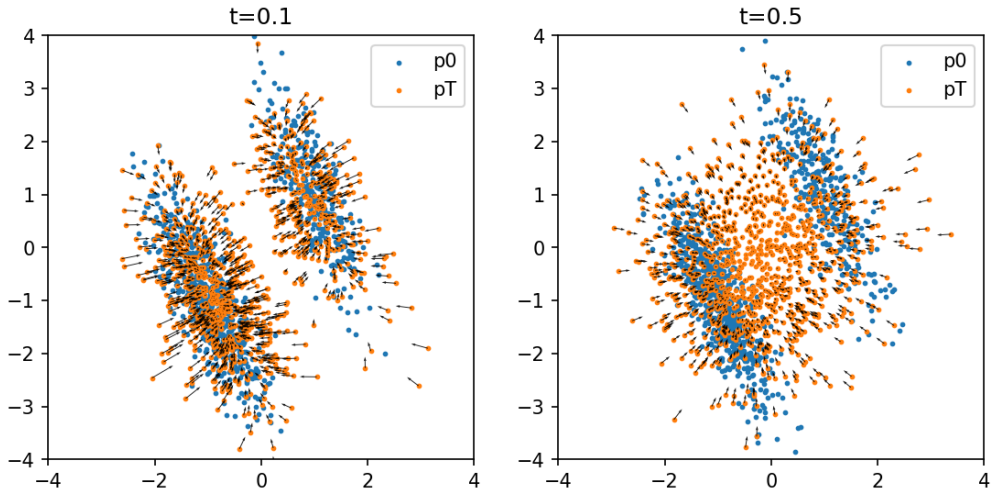

Figure 4.9: Samples from p_t for $t \in \{0.1, 0.5\}$. The true score function $s(x, t)$ directions are overlaid in black arrows.

oracle score loss (best-case performance of our model but unachievable in practical scenarios) to the denoising score matching loss the samples tend to concentrate more than the true distribution.

Two modes in ten dimensions

As we saw in the previous section, multimodality was not challenging for a relatively simple score function model in two dimensions. Let's go to more dimensions. We pick a data distribution $p_0(x) \propto \exp(-E(x))$ where the energy function is sparse in x, i.e. one that is polynomial with each term having low degree. Specifically, we select the 1D Ginzburg-Landau distribution on d sites with nearest-neighbor connections. For $x \in \mathbb{R}^d$ and parameters $a, b \in \mathbb{R}, \beta \in \mathbb{R}_+$ we have the energy function:

$$E(x) = a \sum_{i=1}^{d-1} (x_i - x_{i+1})^2 + b \sum_{i=1}^{d} (1 - x_i^2)^2$$

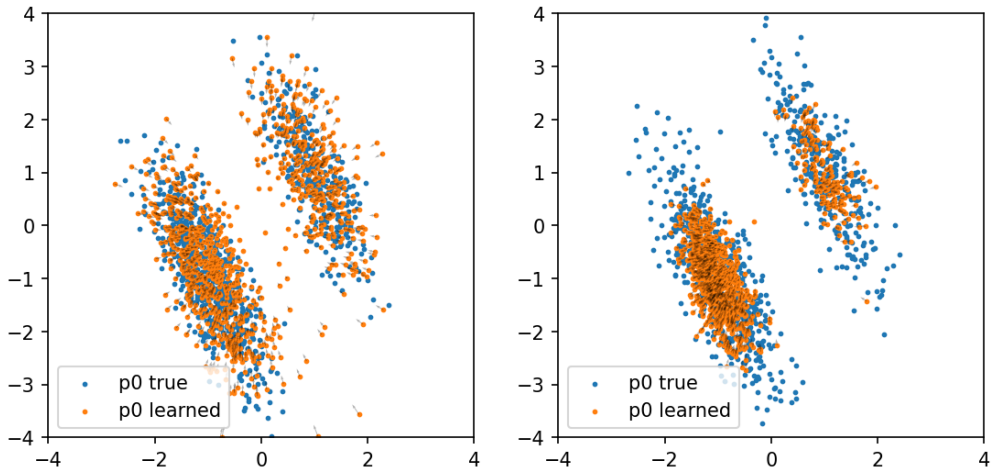

Figure 4.10: Comparison of the result of using the oracle score-matching loss and the denoising score matching loss. Here we illustrate the densities $p_0(x)$ obtained by pushing samples through the reverse diffusion process using score functions $s_\theta(x, t)$ that were trained using either the oracle loss or the denoising loss.

and the target probability density

$$p_0(x) = \frac{\exp(-\beta E(x))}{Z}$$

where Z is the intractable partition function. We can see that sampling p_0 is non-trivial, and no closed form solution for the intermediate densities $p_t(x)$ and score functions $s(x, t)$ can be found.

As the dimensionality of the problem increases, it becomes ever more important to constrain the size of our score function approximation. Recall that our score function model takes the form

$$s_\theta(x, t) = \sum_{i,j} s_\theta^{(ij)}(x, t).$$

Without a constraint on the number of (i, j) components, we must consider $O(d^2)$ in the components that must be fit. Overfitting may be acceptable in most generative model settings, especially when the test set is identical to the test set. But in our case, we are interested

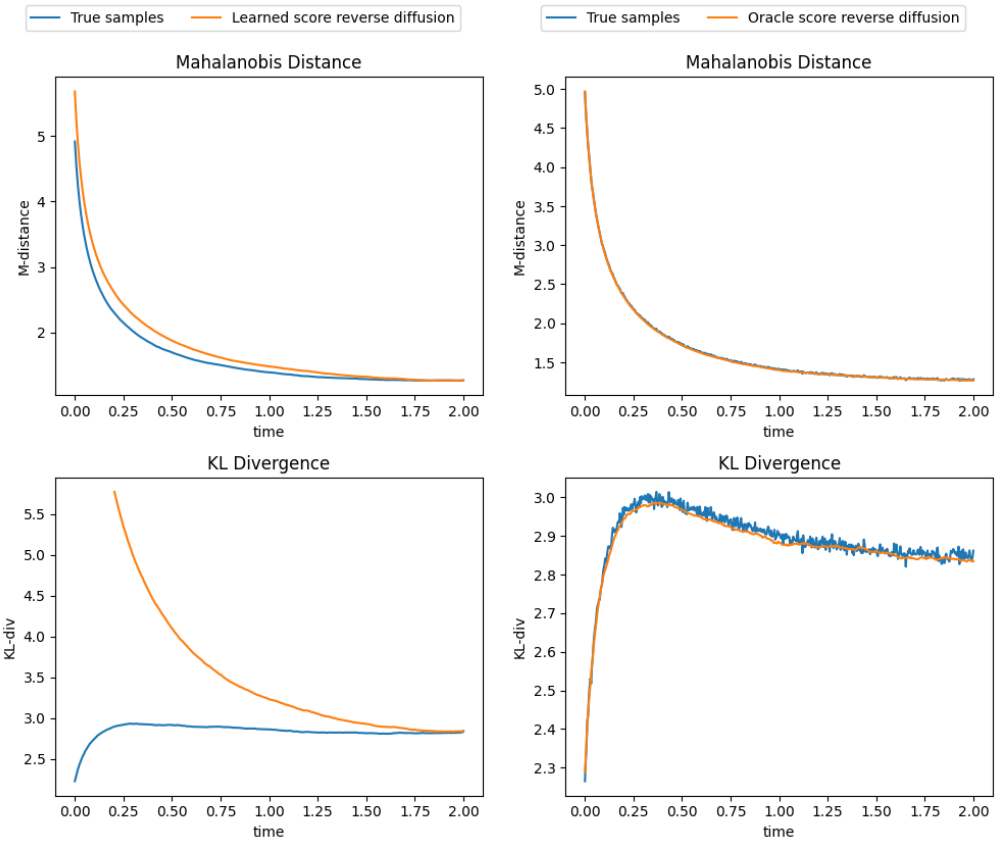

Figure 4.11: Statistics of the trajectories induced by reverse diffusion with Euler-Maruyama and the learned (left, orange) and oracle (right, orange) score functions, compared against the samples from forward diffusion (both left and right, blue).

in methods that may automatically identify the structure of the energy function and prune unnecessary variables and components. Here, we utilize the group lasso idea outlined earlier (Section 4.3.2) to encourage sparsity at the "group" level.

Generating samples with Metropolis-Hastings Note that since we are starting with an unnormalized energy-based model with access only to $E(x)$ but not the partition function Z, we need to generate approximate samples. We employ Metropolis-Hastings, a MCMC method hinted at earlier. The idea is outlined in pseudocode in Algorithm 1. For $a = b = 1$, we use Metropolis-Hastings with $\sigma = 0.1, \beta = 0.7$. Note that Metropolis-Hastings (like other MCMC) algorithms generates correlated samples. To draw T approximately independent and identically distributed samples from $p_0(x)$ using Metropolis-Hastings, we actually run it for $N = 4T$ samples, throw away the first half to give the Markov chain time to burn in, and then randomly draw (with replacement) T samples from the remaining $2T$ samples. Following this procedure, we generate a dataset $\mathcal{D}$ of samples $x_i \sim p_0$.

Algorithm 1 Metropolis-Hastings Algorithm [Metropolis et al., 1953]

Input: Initial state x_0, energy function $E(x)$, number of iterations N, inverse temperature β, proposal standard deviation σ
Output: A set of randomly selected samples $\{x_i\}$ from target distribution $p(x) \propto \exp(-\beta E(x))$
Initialize $x \leftarrow x_0$
Compute initial energy $E(x)$
Initialize sample set $S \leftarrow \{x\}$
for $i \leftarrow 1$ to $N - 1$ **do**
    Propose $x' \leftarrow x + \mathcal{N}(0, \sigma^2)$ ▷ Propose a new state using Gaussian noise
    Compute $E(x')$
    Compute acceptance probability $\alpha \leftarrow \min\left(1, \exp(-\beta(E(x') - E(x)))\right)$
    Sample $u \sim \text{Uniform}(0, 1)$
    if $u < \alpha$ **then**
        $x \leftarrow x'$ ▷ Accept the new state
    end if
    Add x to S
end for
Return Randomly select $N/4$ samples from the last $N/2$ samples in S

Model and fitting We use the cluster basis model ansatz with radial basis functions in the full sense and in the product basis sense (Section 4.3.2 and 4.3.2). Fitting is done using stochastic gradient descent on the denoising score-matching loss (Eq. 4.4) with group lasso loss (Equation 4.12), i.e.

$$\mathcal{L} = \mathcal{L}_{\text{score-matching}} + \lambda \mathcal{L}_{\text{group}}.$$

We set the group lasso scaling parameter to $\lambda = 0.003$ and the learning rate to 0.01. A batch size of 128 was used with 10000 batches total used for training, each batch being drawn i.i.d. from the forward process.

Results In Figure 4.12 we see the samples generated through reverse diffusion using the learned score function $s_\theta(x, t)$ trained using group lasso (blue) compared against samples generated through the forward diffusion process (orange). Note that the forward-diffused samples are much more spread out (higher temperature) than the reverse-diffused samples. One may identify this as a similar phenomenon as what we saw in Section 4.4.1 and 4.4.2 where the result of our reverse diffusions also severely underestimated the spread of the data, yet still capturing the essential details, such as the number and location of the modes.

In Figures 4.13 and 4.14 we can see the norms of the score function components s_{ij}, the first aggregated over all times and the second broken out into different times. The expectation is that overall we see weight on the ± 1 diagonals due to the adjacent $x_i x_{i+1}$ terms in the Ginzburg-Landau energy function, but as time increases more of the weight should lie on the diagonal as a standard Gaussian has no weights mixing any terms off the diagonal.

4.5 Discussion

In this chapter we took a close look at generative diffusion processes. In particular, we specialized to simple cases where the score function could be computed analytically, for example in the case of a delta function or a correlated Gaussian. In the case of the correlated

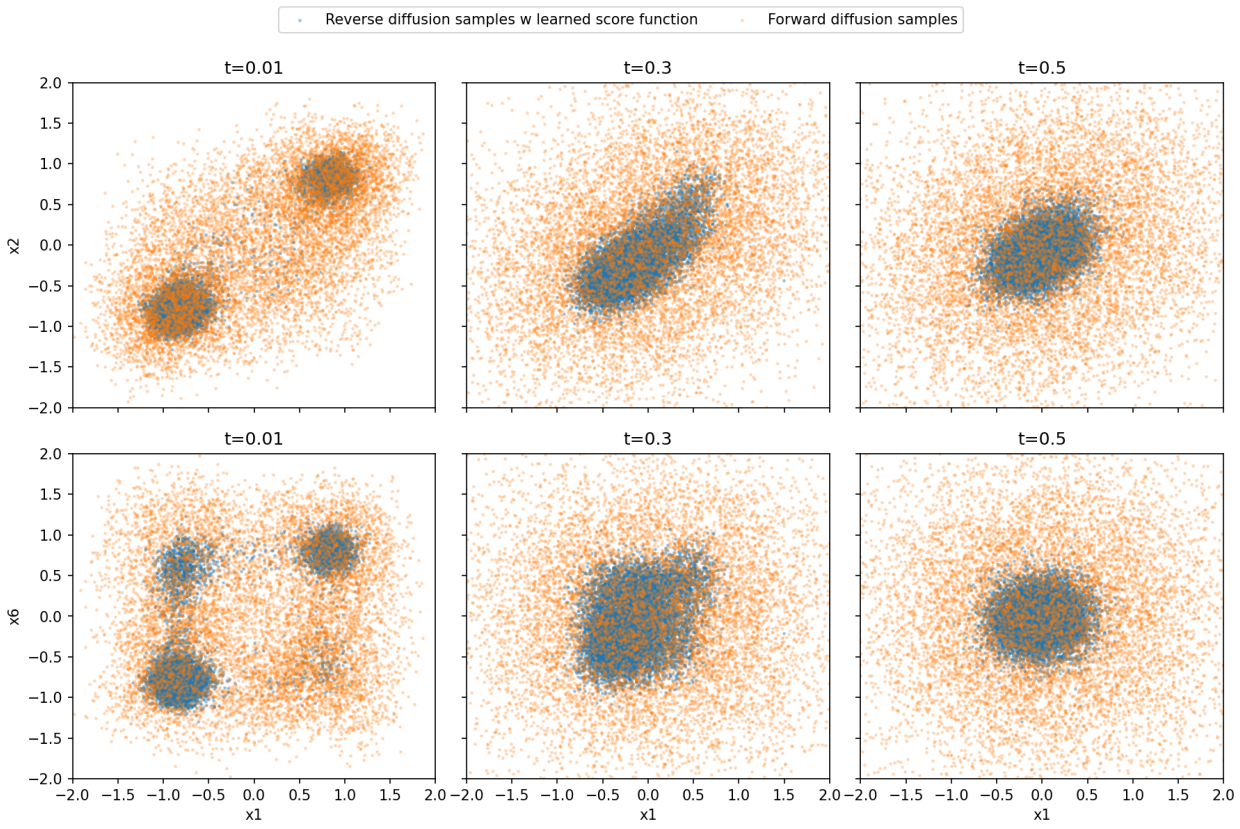

Figure 4.12: Samples generated via reverse diffusion using a learned score function model trained with group lasso, plotted in cross-section for (x_1, x_2) and (x_1, x_6). Blue dots correspond to reverse diffusion with the learned function, orange dots correspond to the training samples at the same time generated using forward diffusion.

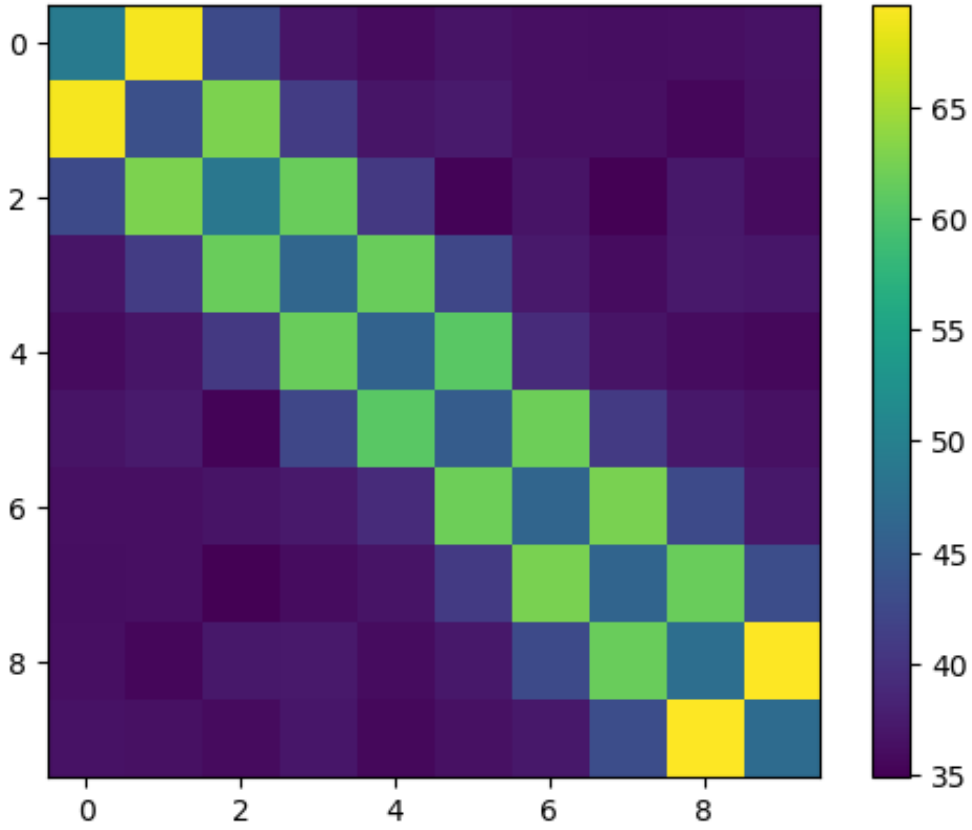

Figure 4.13: The L2 norms of θ_{ij} for all $i, j \in \{1, \ldots, d\}$, averaged over all times.

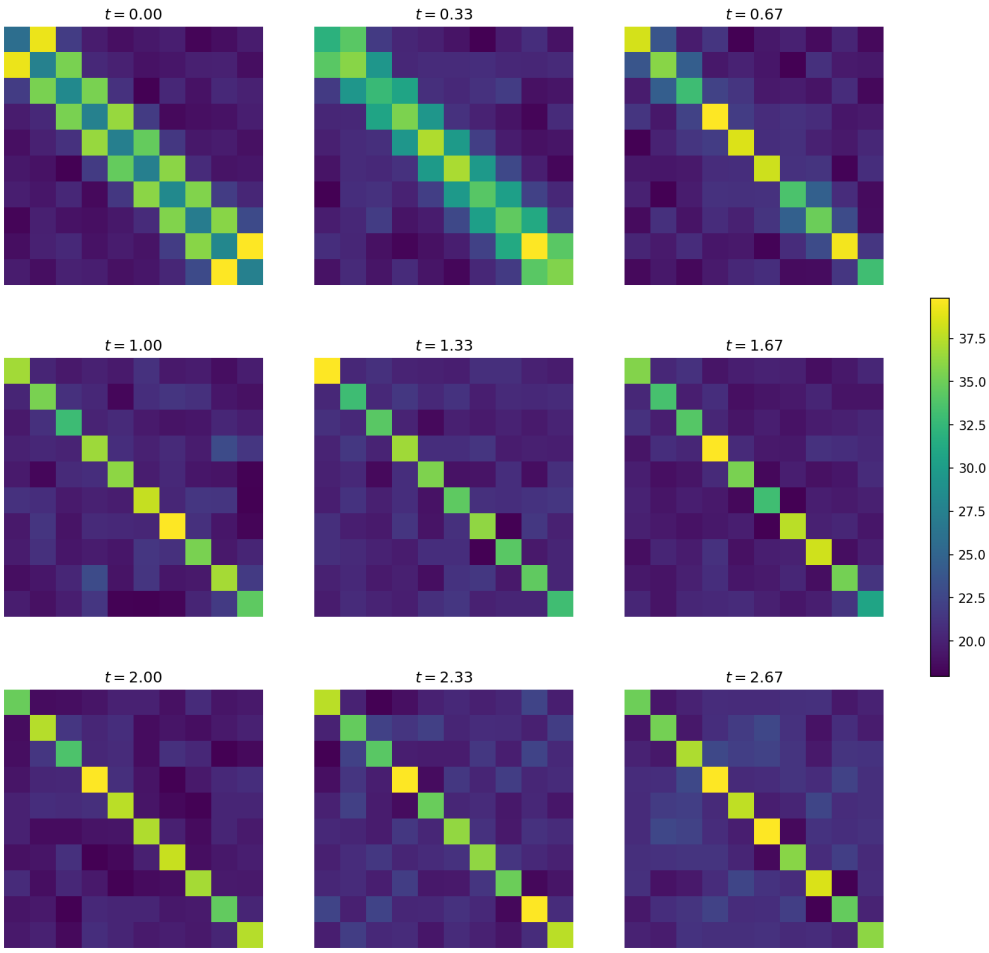

Figure 4.14: The L2 norms of θ_{ij} for all $i, j \in \{1, \ldots, d\}$ at different time knots t_k for our learned score function.

Gaussian, we found that the score function is a linear function of x, requiring the inversion of a time-varying covariance matrix. As time progresses in the diffusion process, the conditioning of this covariance matrix improves monotonically, making the score function easier to estimate accurately.

Furthermore, we explored more complex cases involving multimodal distributions, such as mixtures of Gaussians and higher-dimensional systems like the Ginzburg-Landau model. We observed that the score function becomes more intricate due to the presence of multiple modes and interactions between variables. By employing modeling approaches like the cluster basis expansion and incorporating group lasso regularization, we were able to capture the underlying structure of the score function effectively. The group lasso encouraged sparsity in the learned parameters, aligning with the inherent sparsity of the energy functions in these systems.

A key observation in all these scenarios was the fact that when trained to optimality, the learned score functions generate samples through reverse diffusion that severely underestimate the true variation in the data distributions, almost as if the temperature were lowered. We saw that going from the oracle score function (when known) to the denoising score matching loss could potentially explain this.

There remain still other questions that we did not have a chance to address in this chapter and would be fruitful directions for future work:

- Can we do better than the Langevin SDE with the standard Gaussian as our target? If we change the target distribution to be some other easy-to-sample from distribution, can we converge faster and will our transport path $p_0 \to p_T$ be more efficient?

- How can we measure the efficiency of a transport plan once we have it? What are some key metrics?

- How do these initial results change as we move to higher-dimensional problems? What about some real datasets?

By dissecting both simple and complex cases, we gained valuable insights into the challenges of learning score functions, particularly regarding conditioning, sample complexity, and the impact of multimodality. Our methods hopefully provide a foundation for exploring more sophisticated models that can handle high-dimensional, structured data.

CHAPTER 5

SUPPLEMENT TO DATA-DRIVEN DEEP LEARNING IN THE STRUCTURED PREDICTION OF ELECTRON-IONIZATION MASS SPECTRA

5.1 Data

5.1.1 Preprocessing

All spectra in the NIST 2017 dataset was extracted from the installed software provided upon purchase [NIST]. After scraping and cleaning (including filtering for only the molecules containing HCONFSPCl atoms) we are left with 241,028 molecule-spectra experiments across 237,189 unique molecules (de-duped using INCHI keys) in `nist17-mainlib`, and 63,741 molecule-spectra experiments across 23,200 unique molecules (de-duped using INCHI keys) in `nist17-replib`. Using INCHI keys as unique identifiers for molecules (hash collisions are possible but extremely rare), we confirm that there are zero molecules common to both `nist17-mainlib` and `nist17-replib`. This is because `nist17-mainlib` is the "Main" library, and `nist17-replib` (the "Replicate" library) is a collection of spectral experiments for molecules that were replicated at least 2 or more times. This database is not used during training and is only used during the library matching task. For other datasets (see Table 5.1), we report the number of rows, corresponding to the number of total molecule-spectra experiments. More complete dataset information, including lists of molecules and their metadata, will be made available upon request.

To keep the training runtime at an acceptable level, we filter the training set based on the maximum observed peak mass, the max number of atoms, and the max number of unique fragment formula. In this work, we train and evaluate against molecules with all mass peaks ≤ 511 Daltons, ≤ 48 atoms and ≤ 4096 max unique fragment formula. However, the way

that the models are structured allow us to perform inference against molecules of arbitrary size. We do so when running inference against `pubchem-pred`.

As mentioned in the Main Text, we subdivide our `nist17-mainlib` dataset into train (41.7% of `nist17-mainlib`) / test (10.4% of `nist17-mainlib`) splits by computing the CRC32 checksum of the `morgan4` molecular fingerprint for each molecule and subdividing into splits based on the last digit of the value. This minimizes the chance that exactly identical molecules-spectra experiments or overly-similar molecules are placed into both the training and test sets. Hashed fingerprints ending in $[0, 1]$ were used as the test split, and all others were used as the train split. The function we used to compute the CRC32 checksum of the fingerprint is `morgan4_crc32`, available in `rassp.util` in our released code.

5.1.2 Resolution

All spectra in NIST 2017 are reported at integer Dalton resolution. For downstream training and evaluation, we represent spectra as vectors $s \in \mathbb{R}^{512}$, with bin i containing the observed intensity at charge-to-mass ratio i. For example, bin 1 contains the intensity for $m/z = 1$, bin 2 contains the intensity for $m/z = 2$, and so on. In this work we do not consider fragments ionized to charge $z > 1$, so the spectra may be directly read off as intensities for a given mass value $m/z = m$. We refer to charge/mass ratio and mass interchangeably in this work.

5.1.3 Synthetic high-resolution data

In Section 2.4.3, we discuss performance of SN and FN against a high-resolution synthetic dataset generated by running CFM-ID against molecules from PubChem. We randomly sample 110,000 molecules (100,000 used for training and 10,000 used as a held-out eval set) from PubChem that contain only HCONFSPCl atoms, ≤ 48 atoms, max fragment formula ≤ 4096, and molecular weight ≤ 512. We then run CFM-ID against these molecules using the EI-MS model weights and default configuration as provided by the CFM-ID authors[Allen

Table 5.1: Datasets referenced in this work. The `smallmols` datasets are sourced from NIST 2014 [Allen et al., 2016, NIST], the `nist17` datasets are sourced from NIST 2017 [NIST], and the `pubchem` datasets are sourced from PubChem [Kim et al., 2020].

Name	# Mols	Source	Mean # atoms	Max # atoms	Max unique formula	Max weight
`smallmols-orig`	17,322	NIST14	25.4	87	151,700	772.1
`smallmols-filtered`	13,281	NIST14	24.0	48	4,095	504.0
`nist17-mainlib`	241,028	NIST17	42.3	255	3,427,050	1,674.8
`nist17-replib`	63,741	NIST17	30.1	173	823,680	967.0
`nist17-train`	100,438 (41.7%)	NIST17	30.1	48	4,096	509.7
`nist17-test`	25,205 (10.4%)	NIST17	30.0	48	4,096	510.0
`pubchem-clean`	90,844,616	PubChem	47.7	128	43,868,720	2,046.2
`pubchem-pred`	73,198,384	PubChem	-	-	-	-
`pubchem-clean-filtered`	27,960,210	PubChem	33.7	48	4,096	512.0

et al., 2014]. CFM-ID outputs a list of fragments (in `smiles` form) and the corresponding prediction intensities. We use these results as synthetic high-resolution data, because the fragments have known exact mass and can be binned at arbitrary resolution.

Using the synthetic data, we then construct a dataset by binning at $\Delta m = 0.050$ Dalton resolution. Because we consider molecules with weight up to 512, the dataset contains spectra with $512/\Delta m = 10240$ bins.

We then train SN and FN from scratch against this dataset, varying the number of molecules over 3 orders of magnitude via subsampling: 1k, 10k, and 100k. The metrics reported in Fig. 10 are obtained by taking the best-performing model on each run and evaluating it against the held-out test set of 10k molecules, also binned at $\Delta m = 0.050$ Dalton resolution. Due to the difference in binning between the 0.050 Dalton resolution experiments and the 1 Dalton resolution experiments, test SDPs and other metrics are not directly comparable. However, the relative comparisons between models as we increase the size of the training set are meaningful.

Table 5.2: Features used for each atom

Feature name	Dimensions
Atomic number (integer)	1
Atomic number (one-hot)	8
Valence (integer)	1
Total valence (one-hot)	6
Aromatic (boolean)	1
Hybridization (one-hot)	8
Formal charge (one-hot)	3
Default valence (one-hot)	6
Ring size (one-hot)	5
Total hydrogens (one-hot)	6
Total dimensions	45

5.2 SubsetNet and FormulaNet in detail

5.2.1 Input featurization

Suppose $X \in N_A \times D$ to be our feature matrix for a molecule of N_A atoms and D per-atom features. Using the features listed in Table 5.2, we have $D = 45$ in this work.

- Atomic number (integer)
- Atomic number (one-hot)
- Total valence (integer)
- Is aromatic (boolean)
- Hybridization (one-hot)
- Formal charge (one-hot)
- Covalent radius (float)
- van der Waals radius (float)
- Default valence (one-hot)

- Total hydrogens (one-hot)

In addition, we also generate the symmetric adjacency matrix which contains bond order information $A \in \{0,1\}^{N_A \times N_A \times 4}$, storing 1 in bin 1 for a single bond, bin 2 for a hybridized bond, bin 3 for a double bond, and bin 4 for a triple bond.

Our featurization pipeline is common to both SubsetNet and FormulaNet, and converts the molecule into a tuple (X, A).

5.2.2 Model details

Graph neural networks for computing molecule and atom embeddings

We cite some useful references for understanding and utilizing GNNs for this and related problems [Sanchez-Lengeling et al., 2021, Zhu et al., 2020].

The first phase of SubsetNet and FormulaNet are GNNs that ingest the per-atom features and the adjacency matrix (X_0, A) and outputs per-atom features/embeddings X_L. Specifically, the GNN is a mapping $f : \mathbb{R}^{N_A \times D_0}, \{0,1\}^{N_A \times N_A \times 4} \to \mathbb{R}^{N_A \times D_L}$.

SubsetNet. The layers are as follows:

- Batch normalization
- 16 layers of message-passing graph convolutional layers
  - 512×512 Weight matrix multiply (first layer converts from the input feature dimension $D = 45$ to 512)
  - Adjacency matrix masking
  - Sum with bias
  - LeakyRELU
  - Residual sum with the previous layer's output

  - Instance normalization - `Batchnorm1d`

The output is a matrix of per-atom feature vectors $X_L \in \mathbb{R}^{N_A \times D_L}$. In our case we set N_A to be a maximum of 64 atoms and $D_L = 512$.

FormulaNet. The first phase of FormulaNet, like SubsetNet, is a GNN. Featurization proceeds as before, and the GNN layers are as follows:

- Batch
- 16 layers of message-passing graph convolutional layers
  - 512×512 Weight matrix multiply (first layer converts from the input feature dimension $D = 45$ to 512)
  - Adjacency matrix masking
  - Sum with bias
  - LeakyRELU
  - Residual sum with the previous layer's output
  - Layer normalization - `LayerNorm1d`

The main difference between SubsetNet and FormulaNet's GNN component is the normalization used within each layer.

Parametrizing a probability distribution over the subformula and subsets

In the previous phase, we took as input per-atom feature vectors $X_0 \in \mathbb{R}^{N_A \times D_0}$ and output per-atom embeddings $X_L \in \mathbb{R}^{N_A \times D_L}$. We combine these per-atom embeddings with a separately-constructed enumerations over the possible fragments to produce a probability distribution over the fragments.

SubsetNet. In SubsetNet, the relevant fragments are represented as atom subsets.

The atom subsets are obtained via a direct fragmentation and subset enumeration procedure wherein we recursively break all the bonds out to a given breaking depth $d = 3$, compute the resulting connected components, and throw away information about the edges and retain only the atoms that were present in connected components together as atom subsets. Each atom subset is stored as a vector $s_i \in \{0, 1\}^{N_A}$ with 1 if the corresponding atom was present in the subset, and 0 if not.

The per-atom embeddings from the first phase X_L is then combined with the atom subsets (obtained via direct fragmentation and subset enumeration) to generate per-subset embeddings. Let the atom subset indicator matrix be $S \in \{0, 1\}^{N_S \times N_A}$. The matrix multiplication SX_L gives us a matrix of per-subset embeddings $X_S \in \mathbb{R}^{N_S \times D_L}$, which corresponds to doing a linear combination of the per-atom embeddings for only the atoms present in each subset.

In addition, for each subset we also take its chemical formula and generate a cumulative one-hot binary feature vector. Since we restrict to molecules containing only HCONFSPCl atoms (8 unique elements), we require constraints on the maximum number of allowed atoms for each element. The maximum allowed elements for each element in HCONFSPCl respectively was $[50, 46, 30, 30, 30, 30, 30, 30]$. The corresponding embedding size for any single formula is the sum of the max allowed elements, here 276. Hence, we have the per-subset embeddings $X_S \in \mathbb{R}^{N_S \times D_L}$ and the per-subset chemical formula embeddings $X_{SF} \in \mathbb{R}^{N_S \times 276}$.

The second phase combines the per-subset embeddings X_S and the per-subset chemical formula embeddings X_{SF} via a fully-connected layer, and then additionally pass it through two more fully-connected layers to reduce the per-subset embeddings down to per-subset logit scores, which are then converted into subset probabilities via softmax.

FormulaNet. In FormulaNet, the relevant fragments are represented as chemical formula. This is essentially taking the atom subset information from above, and taking a quotient operation over the subsets where we identify all subsets that have the same chemical formula

as equivalent.

We generate the set of subformulae for a given molecule. As before in SubsetNet, we produce a cumulative one-hot binary feature vector for each subformula.

The formula embeddings and per-atom embeddings X_L from the first phase are then mapped into the same space and an attention operation is taken, amounting to a pairwise comparison between all formula and all atom embeddings. The resulting similarities are converted by softmax into values between 0 and 1, and then used to scale and reduce the per-atom embeddings down to a per-subformula embedding.

Like SubsetNet, the next phase combines the per-subformula embeddings with the per-formula one-hot embeddings using a `GRUCell`. The output is passed through three fully-connected layers (each containing 128 units) to get a per-formula logit score, just as SubsetNet combines the per-subset embeddings with the per-subset chemical formula.

Further details for model implementation are available in our provided code.

5.2.3 Hyperparameters

The loss function used was a simple MSE loss against the square root of spectral intensities. Scaling the intensities by a power of 0.5 in the loss function was intended to de-emphasize outlier intensities. Both models were trained to convergence using the Adam optimizer with learning rate 0.0002.

5.2.4 Training

Both models were trained to convergence after 20 passes over the full `nist17-train` dataset, which took 100 hours on a workstation with 2 Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz CPUs and 2 NVIDIA RTX 2080 Ti GPUs. Inference on small molecules with ≤ 48 atoms and ≤ 4096 max formula on the same workstation achieved an average of 16 molecules per second. All training and inference was performed using 32 threads and a single GPU.

5.2.5 Reproducing results

In our publicly-released code, we provide the model code and configuration files used when producing our results. The NIST dataset is proprietary and cannot be released by us. Instead, we provide the first 100 molecules from the `smallmols-orig` dataset [Allen et al., 2016]. The INCHI and SMILES strings are provided, in addition to high-res predicted spectra obtained by running the publicly-available CFM-ID EI-MS model [Allen et al., 2016]. We provide a script that trains a model against this dataset, performs basic forward inference, and computes metrics for the forward prediction task and the library matching task.

Pretrained model weights, including the best FormulaNet and SubsetNet models we trained, are not included with the code package due to size constraints but are publicly-available at `https://people.cs.uchicago.edu/~ericj/rassp/`. Instructions for use are included in the code package `README.md`.

5.3 Comparisons to other models

5.3.1 CFM-ID [Allen et al., 2016]

CFM-ID provided the exact smiles strings corresponding to the smallmols dataset. In order to get the most favorable comparison for CFM-ID, we used the provided spectra (which performed better than the spectra output by the model using the weights provided) as our benchmark in Fig. 6. However, due to lack of coverage of our dataset, we used the provided EI-MS weights and the default configuration to generate spectra from PubChem molecules for the synthetic dataset employed in producing Fig. 10.

5.3.2 NEIMS [Wei et al., 2019]

We retrained the NEIMS model on the same `nist17-train` dataset. Note that the NEIMS code accepts `.tfrecord` format only. In addition, the code expects spectra to be normalized

Table 5.3: Forward model runtime

Model name	Runtime (ms) per mol	Mols per second
CFM-ID	300,000 [Allen et al., 2016]	0.0033
NEIMS	5[Wei et al., 2019]	200
SubsetNet	53	19
FormulaNet	23	44

to have a maximum magnitude of 999 (as detailed in the original paper) [Wei et al., 2019]. We did not use the provided model weights due to their training set containing molecules from both our train and test sets. There is no guarantee that we trained the model optimally, however we did train for a much longer period (100 epochs or passes over our NIST 2017 training set) with the default hyperparameters to ensure that our comparison would be as favorable to the original work as possible.

5.3.3 Runtime

Forward model runtime information is detailed in Table 5.3.

CFM-ID numbers and NEIMS numbers are pulled from the reported numbers in the original papers.

All training, inference, and benchmarks were performed on a server with 1 Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz CPU and 1 NVIDIA RTX 2080 Ti GPUs. Inference runtimes were computed using 16 PyTorch CPU workers for loading data.

5.4 PubChem inference

We take our best-performing FormulaNet model and evaluate it on `pubchem-pred`, containing 73.2M small molecules from the PubChem database. All molecules and predicted spectra in `pubchem-pred` will be made available at our public website `spectroscopy.ai`.

5.5 Analysis of molecular similarity vs performance

All mentions of molecular similarity refers to the Tanimoto similarity (AKA Jaccard similarity or the ratio of Intersection over Union), defined on two binary arrays as:

$$\text{TanimotoSimilarity}[\vec{f}, \vec{g}] = \frac{\vec{f} \& \vec{g}}{\vec{f} || \vec{g}}$$

where the & operator represents a bitwise-AND operation and the || operator represents a bitwise-OR operation. This similarity measure is a real number in $[0, 1]$.

In these studies, we used the default RDKit fingerprint for molecules (2048-dimension binary bitvector) [Landrum].

5.5.1 Forward spectral prediction performance and similarity

For every molecule in our test set ($n = 25205$), we find its nearest neighbor in the training set ($n = 100438$) as measured by similarity discussed above. We present the scatter plot of SDP (Y-axis) scattered against the similarity to training (X-axis) in Fig. 5.1 below.

Figure 11 (Main Text) is the same data, but additionally binned for clarity. We bin the similarity in deciles (round to the nearest 10%) and compute the 10%-50%-90% percentile values within each bin. We present the number of molecules in each similarity bin of Figure 11 in Table 5.4. Note that as the similarity decreases, we have fewer molecules in each bin. The values in lower bins are expected to be more noisy for this reason.

5.5.2 Library matching performance and similarity

For every molecule in the NIST Replicate Library ($n = 63741$) we find its nearest neighbor in the NIST Main Library ($n = 241028$) as measured by similarity discussed above.

Because matching rank is a heavily skewed value that ranges over several orders of magnitude (the most common matching rank is < 10, but matching rank can often reach

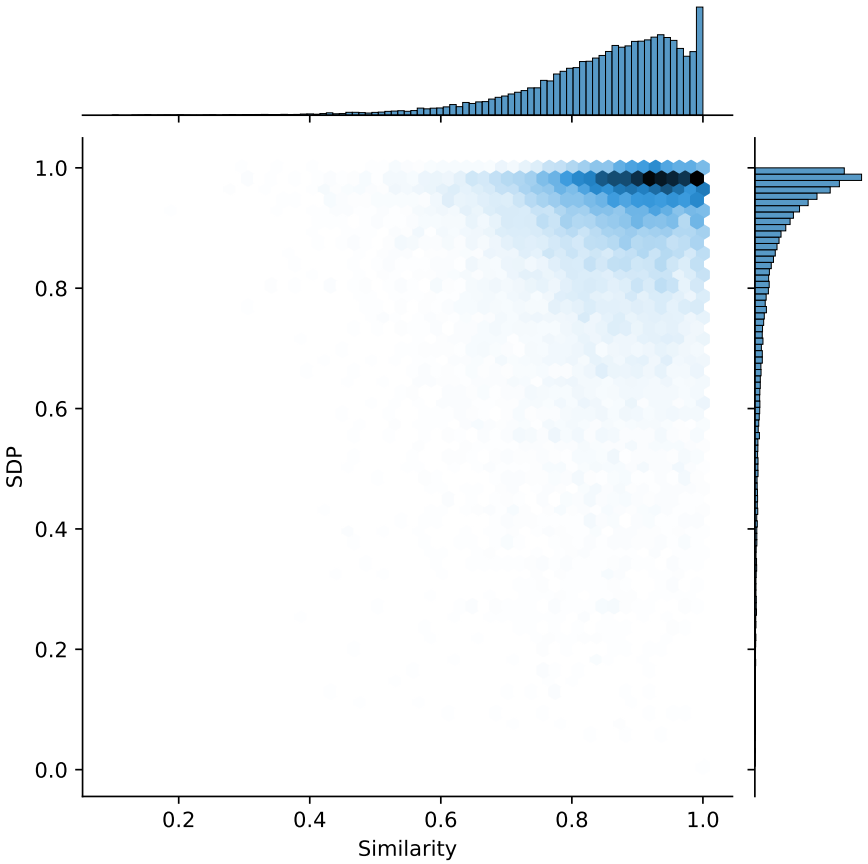

Figure 5.1: SDP vs similarity hex jointplot

100 or 1000), we do not report 10%-50%-90% percentiles but rather compute the mean in log-space. We use logarithms to base 10.

We present the scatter plot of log10(rank) (Y-axis) scattered against similarity (X-axis) in Fig. 5.2.

Unlike the forward spectral prediction analysis, we only bin into "low similarity" $< 90\%$ and "high similarity" $\geq 90\%$ molecules here, and compute the mean log10(rank) over each bin. Percentiles make little sense for this data because a majority of molecules have a rank of 1 (log10 rank of 0). The low similarity molecules ($n = 29339$) had a mean log10 rank of 0.110 and the high similarity molecules ($n = 18771$) had a mean log10 rank of 0.135.

Table 5.4: Number of molecules in each similarity bin for Main Text Figure 11

Decile	n
0%	0
10%	5
20%	25
30%	64
40%	176
50%	440
60%	1146
70%	2756
80%	6550
90%	10117
100%	5188

5.6 Additional statistical analysis

We would like to understand how our reported performance metrics (SDP and others for forward spectral prediction, matching rank for library matching / database lookup) vary as our models are trained on different subsets of the data. To do so, we split our dataset into 5 cross-validation splits, and trained 5 different FormulaNet models to 1000 epochs using each split (choose 4 for training, hold 1 out for testing).

For the forward performance, the standard-deviation of headline (the value we report in the abstract) mean SDP (evaluated on the held-out test set, which changes from training run to training run) we see on the order of 0.10%. At the level of individual molecules, we get an average run-to-run std-dev in SDP of 2.1% (over 5 runs).

For the library matching task, we looked at the dispersion in rankings between the 5 models. Because each model is trained on a different subset of data (80% is selected and 20% is held out), there is likely to be gaps where a certain model will fail to rank the query molecule highly. We see this borne out in practice. 91% of the time all 5 models will rank the query molecule in the top 10 molecules and achieve a median rank dispersion (the max delta between the highest rank and the lowest rank achieved by any of the five models) of 0.0 and an average rank dispersion of 13.6. The other 9% of the time, we see a median rank

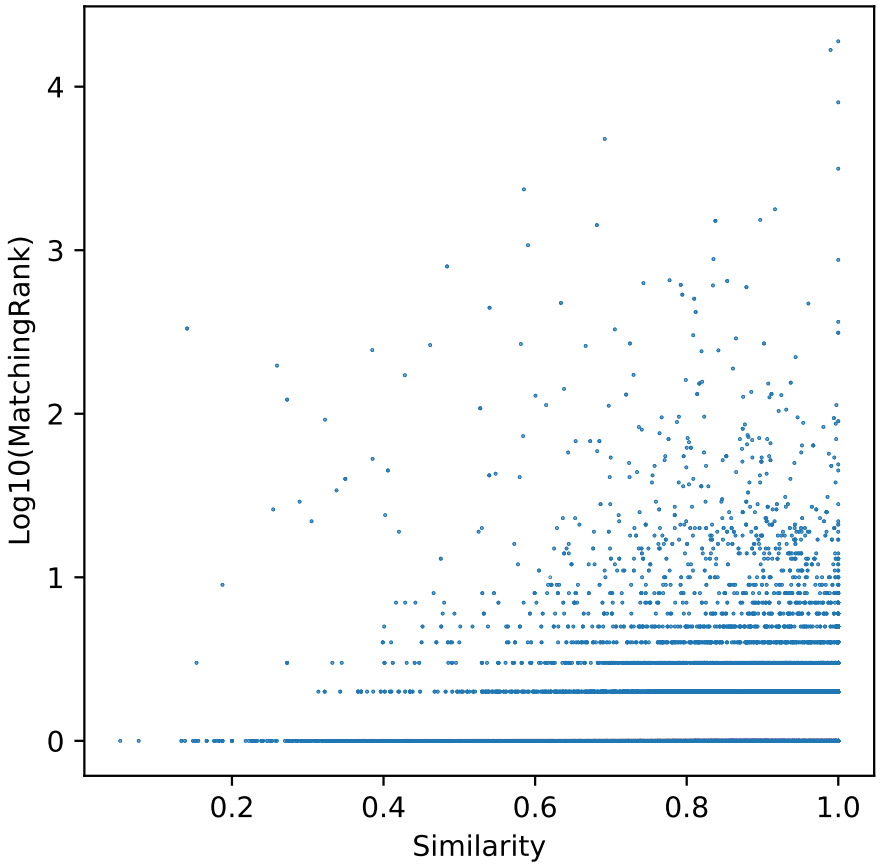

Figure 5.2: Log10(MatchingRank) vs similarity scatter plot

dispersion of 24 and an average rank dispersion of 152.1, suggesting that when a model does fail, it fails spectacularly in comparison to the others.

This suggests that accumulating more data on diverse molecular structures is key in achieving the best possible performance on both tasks.

5.7 Discussion

5.7.1 Glucose example

Given a molecular graph $G = (V, E)$ where the vertexes correspond to atoms and the edges correspond to bonds, our subset enumeration process outputs a list of atom subsets. These

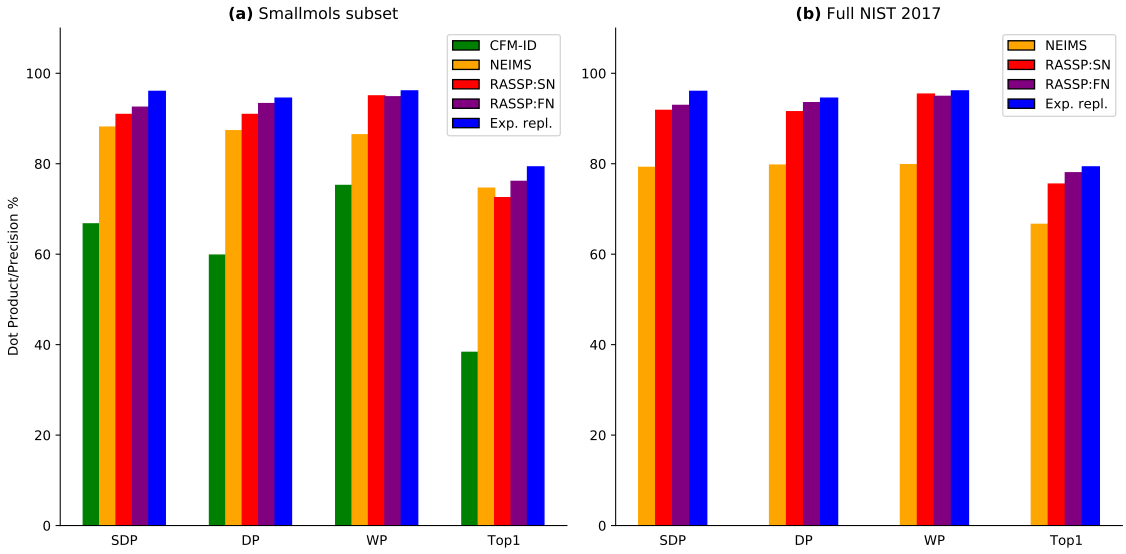

Figure 5.3: EI-MS prediction performance. Similar to the figure in the Main Text, but the bars here represent the mean value over the entire dataset, and Top-1 accuracy (Top1) is reported instead of Weighted False-Positive Rate (WFPR). Top-1 accuracy was left out of the Main Text due to 10-50-90% percentile reporting failing to display meaningful bars, since Top-1 accuracy is either 0 or 1 for each row (the peak with highest intensity in predicted spectra also matches the peak with highest intensity in the target).

subsets are not randomly generated by choosing a subset of the atoms, but are instead generated via a physically-plausible "break-and-rearrange" process by which all possible bond breakages out to integer depth d are iteratively considered, followed by any possible rearrangement of hydrogens. Running this process on glucose $C_6H_{12}O_6$ outputs 164 unique subsets of the 12 heavy-atoms (6 carbon and 6 oxygen). Considering subsets of all hydrogens is also possible, but makes this process more computationally-intensive.

An example of the atom (vertex) subsets output by our subset enumeration process is shown in Fig. 5.4. All atom subsets form one connected-component due to our "physically-plausible fragment" assumption. If a bond breakage would generate two separate fragments, then both are considered as separate atom subsets.

Each atom subset maps surjectively onto the set of unique chemical subformulae of the original molecule (each atom subset corresponds to a subformulae, and there can be many subsets that map to the same subformula), and each chemical subformulae gives rise to a

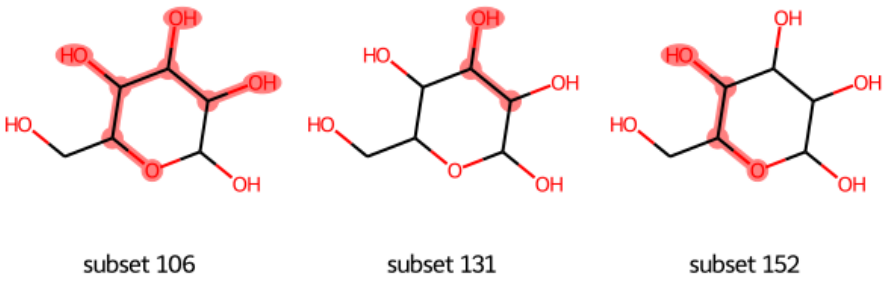

Figure 5.4: Three randomly-chosen heavy-atom (C and O only) subsets of glucose

Table 5.5: Chemical formulae and molecular weight for the three subsets depicted in Fig. 5.4

Subset idx	Chemical formula	Mol weight
106	C4O4	112.04
131	C2O1	40.02
152	C2O2	56.02

unique peak distribution. In Table 5.5, we see the chemical formula corresponding to each subset.

The peak distribution for each of the three subsets in Fig. 5.4 is shown in Fig. 5.5. Each peak is shaded according to the intensity. Note the primary peak centered at the exact weight of the molecular ion listed in Table 5.5, but also the faint echoes of peaks at higher mass, caused by the naturally-occurring isotopic variability of carbon and oxygen.

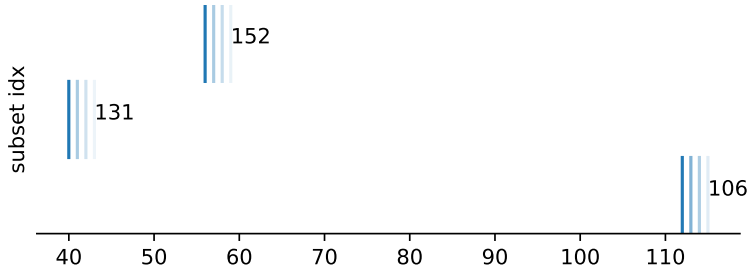

Figure 5.5: The barcode spectra corresponding to the three subsets depicted in Fig. **??**. Each peak is shaded proportionally to the intensity. The X-axis corresponds to Daltons/amu.

Our prediction models are trained to output a probability distribution over subformulae (RASSP:FN) and subsets (RASSP:SN). Once such a probability distribution is obtained, it is a simple matter of scaling the peak distribution corresponding to each subset with the

probability for that subset, and then summing all the peak distributions together to get the final output spectrum.

For sake of completion, we also provide the indicator matrix that describes all 164 heavy-atom subsets in Fig. 5.6 and the barcode spectrum for each subset in Fig. 5.7.

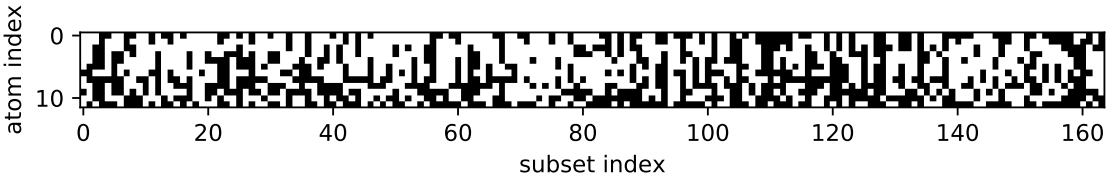

Figure 5.6: Indicator matrix depicting each of the 164 heavy-atom subsets of glucose produced by our enumeration scheme. The presence of the atom is shown in white, and the absence of the atom is shown in black. The first 6 atom indexes are carbon and the last 6 atom indexes are oxygen.

5.7.2 Toluene example

Toluene is a simple example illustrating the tradeoffs and improvements our spectral prediction process makes.

Fig. 5.8 illustrates FormulaNet's prediction of the toluene spectrum (negative values, in blue) vs the ground-truth experimental spectrum (positive values, in black). We note that our predicted spectrum captures all the important peaks attributable to fragments in the well-studied fragmentation process at [39, 51, 65, 77, 91, 92] Daltons. We have highlighted these peaks as light vertical lines in red.

We note that the 7-member ring ion featured in the fragmentation process is not a fragment or subgraph explicitly considered in our process, due to computational constraints. Rather, our subset and subformula enumeration process considers both the 91 amu 6-member ring ion with attached carbon (after a single hydrogen loss) and the 7-member ring ion as identical, due to having the same set of underlying atoms. Throwing away bond information in the subset/subformula enumeration process is critical in making our solution computationally

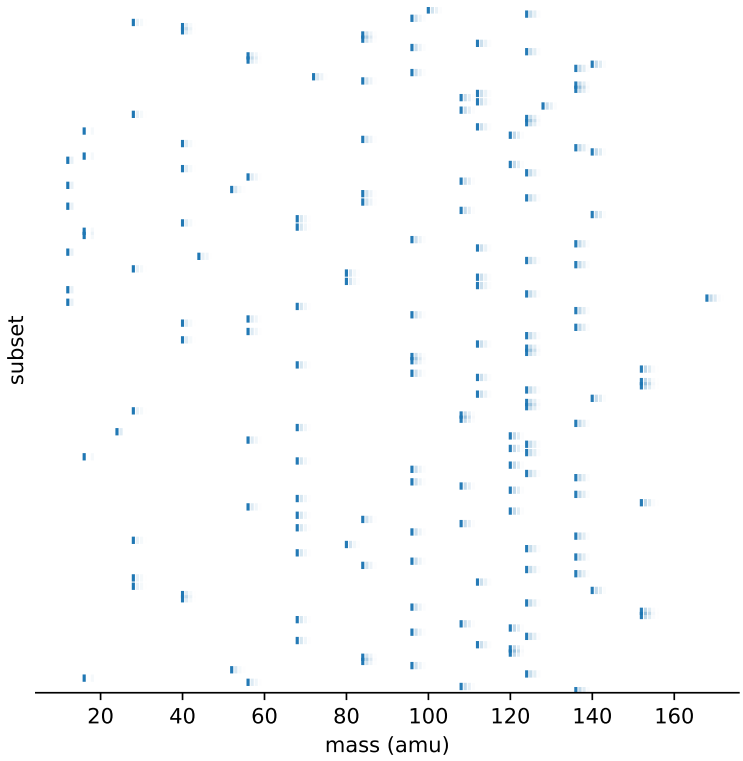

Figure 5.7: Barcode spectra for each of the 164 heavy-atom subsets of glucose produced by our enumeration scheme.

feasible, but it does result in losing the ability to separate the 6-member ring and the 7-member ring, even though they present in the mass spectrometer as the same peak.

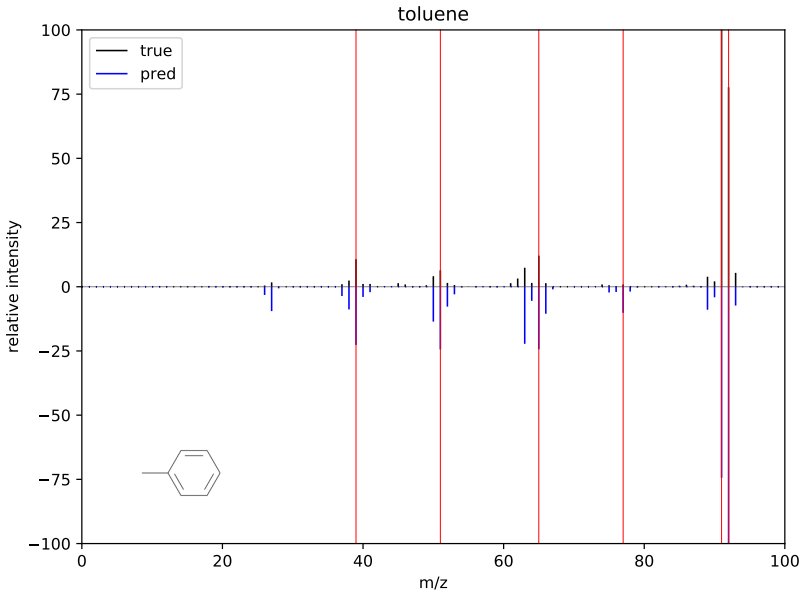

Figure 5.8: Toluene

CHAPTER 6

SUPPLEMENT TO STOCHASTIC SUM-OF-SQUARES FOR PARAMETRIC POLYNOMIAL OPTIMIZATION

6.1 Notation

Let $\mathcal{P}(X)$ and $\mathcal{P}(\Omega)$ denote the spaces of polynomials on $X \subseteq \mathbb{R}^n$ and $\Omega \subseteq \mathbb{R}^d$, respectively, where X and Ω are (not-necessarily compact) subsets of their respective ambient spaces $\mathbb{R}^n$ and $\mathbb{R}^d$. Specifically, all polynomials of the forms below belong to their respective spaces:

$$p(x) = \sum_{\alpha \in \mathbb{Z}_{\geq 0}} c_\alpha x^\alpha \in \mathcal{P}(X), \quad p(\omega) = \sum_{\alpha \in \mathbb{Z}_{\geq 0}} c_\alpha \omega^\alpha \in \mathcal{P}(\Omega)$$

where $x = (x_1, \ldots, x_n), \omega = (\omega_1, \ldots, \omega_d)$, α is a multi-index for the respective spaces, and c_α are the polynomial coefficients.

Let $\mathcal{P}^d(S)$ for some $S \in \{X, \Omega\}$ denote the subspace of $\mathcal{P}(S)$ consisting of polynomials of degree $\leq d$, i.e. polynomials where the multi-indices of the monomial terms satisfy $||\alpha||_1 \leq d$. $\mathcal{P}_{\text{SOS}}(X \times \Omega)$ refers to the space of polynomials on $X \times \Omega$ that can be expressible as a sum-of-squares in x and ω jointly. Additionally, $W \succcurlyeq 0$ for a matrix W denotes that W is symmetric positive semidefinite (PSD). Finally, $\mathbb{P}(\Omega)$ denotes the set of Lebesgue probability measures on Ω.

6.2 Related work

6.2.1 Sum-of-squares theory and practice

The theoretical justification underlying the SDP relaxations in global optimization we use here derive from the Positivstellensätz (positivity certificate) of [Putinar, 1993], a representation theorem guaranteeing that strictly positive polynomials on certain sets admit sum-of-squares

representations. Following this, [Lasserre, 2001, 2018, 2023] developed the Moment-SOS hierarchy, describing a hierarchy of primal-dual SDPs (each having fixed degree) of increasing size that provides a monotonic non-decreasing sequence of lower bounds.

There is rich theory underlying the SOS hierarchy combining disparate results from algebraic geometry [Parrilo, 2000, Lasserre, 2018, 2023], semidefinite programming [Nie, 2009, Papp and Yildiz, 2019], and complexity theory [de Klerk, 2008, O'Donnell, 2016]. The hierarchy exhibits finite convergence in particular cases where convexity and a strict local minimum are guaranteed [Nie, 2014], otherwise converging asymptotically [Bach and Rudi, 2023]. In practice, the hierarchy often does even better than these guarantees, converging exactly at c_s^* for some small s.

The SOS hierarchy has found numerous applications in wide-ranging fields, including: reproducing certain results of perturbation theory and providing useful lower-bound certifications in quantum field theory and quantum chemistry [Hastings, 2022, 2023], providing better provable guarantees in high-dimensional statistical problems [Hopkins, 2018, Hopkins and Li, 2018], useful applications in the theory and practice of sensor network localization [Nie, 2009, Sedighi et al., 2021] and in robust and stochastic optimization [Bertsimas et al., 2011].

Due to the SDP relaxation, the SOS hierarchy is quite powerful. This flexibility comes at a cost, primarily in the form of computational complexity. The SDP prominently features a PSD matrix $W \in \mathbb{R}^{a(n,d,s)\times a(n,d,s)}$ with $a(n,d,s)$ scaling as $\binom{n+d+s}{s}$ for n dimensions and maximum degree s. Without exploiting the structure of the polynomial, such as locality (coupled terms) or sparsity, solving the SDP using a standard interior point method becomes prohibitively expensive for moderate values of s or n. Work attempting to improve the scalability of the core ideas underlying the SOS hierarchy and the SDP method include [Ahmadi and Majumdar, 2019, Papp and Yildiz, 2019].

6.2.2 Stochastic sum-of-squares and parametric polynomial optimization

The S-SOS hierarchy we present in this work as a solution to parametric polynomial optimization was presented originally by [Lasserre, 2010] as a "Joint + Marginal" approach. That work provides the same hierarchy of semidefinite relaxations where the sequence of optimal solutions converges to the moment vector of a probability measure encoding all information about the globally-optimal solutions $x^*(\omega) = \text{argmin}_x f(x, \omega)$ and provides a proof that the dual problem (our primal) obtains a polynomial approximation to the optimal value function that converges almost-uniformly to $c^*(\omega)$.

6.2.3 Uncertainty quantification and polynomial chaos

Once a physical system or optimization problem is characterized, sensitivity analysis and uncertainty quantification seek to quantify how randomness or uncertainty in the inputs can affect the response. In our work, we have the parametric problem of minimizing a function $f(x, \omega)$ over x where ω parameterizes the function and is drawn from some noise distribution $\nu(\omega)$.

If only function evaluations $f(x, \omega)$ are allowed and no other information is known, Monte Carlo is often applied, where one draws $\omega_k \sim \nu(\omega)$ and solves many realizations of $\inf_x f_k(x) = f(x, \omega_k)$ to approximately solve the following stochastic program:

$$f^* = \inf_x \mathbb{E}_{\omega \sim \nu}[f(x, \omega)]$$

Standard Monte Carlo methods are ill-suited for integrating high-dimensional functions, so this method is computationally challenging in its own right. In addition, we have no guarantees on our result except that as we take the number of Monte Carlo iterates $T \to \infty$ we converge to some unbiased estimate of $\mathbb{E}_{\omega \sim \nu}[f(x, \omega)]$.

Our approach to quantifying the uncertainty in optimal function value resulting from

uncertainty in parameters ω is to find a deterministic lower-bounding $c^*(\omega)$ which guarantees $f(x,\omega) \geq c^*(\omega)$ no matter the realization of noise. This is reminiscent of the polynomial chaos expansion literature, wherein a system of some stochastic variables is expanded into a deterministic function of those stochastic variables, usually in some orthogonal polynomial basis [Sudret, 2008, Najm, 2009].

6.2.4 Building intuition for the connection between SDPs and sum-of-squares

We stated that it was obvious that a sum-of-squares polynomial admits a representation of the form $m(x)^T W m(x)$, but we didn't explicitly show why. The idea hinges on the Cholesky decomposition of a PSD matrix:

Example 6.2.1. Let $s = 2$ and $X \times \Omega = \mathbb{R} \times \mathbb{R}$. We exhibit a basis function $m_2(x,\omega) : X \times \Omega \to \mathbb{R}^6$ over the monomials:

$$m_2(x,\omega) = [1, x, \omega, x^2, x\omega, \omega^2]^T$$

Observe that

$$m_2(x,\omega)^T A m_2(x,\omega) \in \mathcal{P}^{2s}(X \times \Omega)$$

If W is PSD, it exhibits a Cholesky factorization $W = LL^T$ where L is lower-triangular, enabling us to write

$$m_2(x,\omega)^T W m_2(x,\omega) = ||L^T m_2(x,\omega))||_2^2$$

Thus, $m_2^T W m_2 \in \mathcal{P}^{2s}_{\text{SOS}}(X \times \Omega)$. □

This idea is not just a trivial one linking sum-of-squares polynomials to SDPs, this idea is also useful in practical optimization and is commonly known as the Burer-Monteiro approach [Burer and Monteiro, 2003, Boumal et al., Jiang and Khoo].

6.2.5 Theory of orthogonal polynomials

In this work, we draw noise uniformly over $[-1, 1]^d$. Numerically, it is expected we would achieve faster and more stable results in using which is a natural fit for the Legendre basis set. But depending on the noise distribution, other choices of basis may be a better fit, such as Hermite polynomials for Gaussian random variates.

The theory of orthogonal polynomials is intimately related to the character of the noise distribution $\nu(\omega)$; in fact given any choice of $\nu(\omega)$ we may find a sequence of polynomials $\{P_n(x)\}$ where each polynomial in the sequence is orthogonal relative to any other with respect to some weight function $\nu(x)$, e.g. for 1D polynomials over $[a, b]$ we require $\int_a^b P_m(x)P_n(x)\nu(x)dx = \delta_{m,n}$ where $\delta_{m,n}$ is the Kronecker delta, i.e. 1 if $m = n$ else 0. Several well-known orthogonal polynomial sequences have names and we provide their corresponding weight functions [Schmüdgen, 2017]:

- Legendre polynomials and Uniform$(-1, 1)$, i.e. $\nu(x) = 1$
- Hermite polynomials and Gaussian$(0, 1/2)$, i.e. $\nu(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$
- Chebyshev polynomials and the Wigner semicircle distribution of radius 1, i.e. $\nu(x) = \pi^{-1}(1 - x^2)^{-1/2}$

Using the appropriate sequence of orthogonal polynomials that matches the noise distribution $\nu(\omega)$ in the basis $m_s(x, \omega)$ is not exactly necessary as the results still apply without (e.g. we use the standard monomial basis and specialize to $\omega \sim$ Uniform$(-1, 1)$). However, it is anticipated that using the class of orthogonal polynomials that matches the noise distribution would make the problem more numerically stable, particularly as the problem gets large.

6.3 An example

Example 6.3.1. Let $f(x,\omega)$ be some polynomial of degree $\leq 2s$ written in the standard monomial basis, i.e.

$$\begin{aligned} f(x,\omega) &= \sum_{||\alpha||_1 \leq 2s} f_\alpha x^\alpha \\ &= \sum_{||\alpha||_1 \leq 2s} f_{(\alpha_1,\ldots,\alpha_{n+d})} \prod_{i=1}^{n} x_i^{\alpha_i} \prod_{i=1}^{d} \omega_i^{\alpha_{n+i}} \end{aligned}$$

Let $m_s(x,\omega) \in \mathbb{R}^{a(n,d,s)}$ be the basis vector representing the full set of monomials in x,ω of degree $\leq s$ with $a(n,d,s) = \binom{n+d+s}{s}$.

For all $\alpha \in \mathbb{Z}_{\geq 0}^{n+d}$ with $||\alpha||_1 \leq 2s$ and $\alpha_k = 0$ for all $k \in \{1,\ldots,n\}$ (i.e. monomial terms containing only $\omega_1,\ldots,\omega_d$) we must have:

$$\int_{X\times\Omega} \omega^\alpha \mathrm{d}\mu(x,\omega) - \int_\Omega \omega^\alpha \mathrm{d}\nu(\omega) = 0$$

Explicitly, for μ to be a valid probability distribution we must have:

$$\int_{X\times\Omega} \mathrm{d}\mu(x,\omega) - 1 = M_{0,0} - 1 = y_{(1,0,\ldots)} - 1 = 0$$

Suppose $\Omega = [-1,1], \omega \sim \text{Uniform}(-1,1)$ so that $d = 1$, $\nu(\omega) = 1/2$. We require:

$$\int_{X\times\Omega} \omega^\alpha \mathrm{d}\mu(x,\omega) = \int_{[-1,1]} \omega^\alpha \mathrm{d}\nu(\omega) = \begin{cases} 1 & \alpha = 0 \\ 0 & \alpha = 1 \\ \frac{1}{3} & \alpha = 2 \\ 0 & \alpha = 3 \\ \frac{1}{5} & \alpha = 4 \end{cases}$$

□

6.4 Strong duality

To guarantee strong duality theoretically, we need a strictly feasible point in the interior (Slater's condition). For us, this is a consequence of Putinar's Positivstellensatz, if $f(x, \omega)$ admits a decomposition as $f(x, \omega) = c(\omega) + g(x, \omega)$ where $g(x, \omega) > 0$ (i.e. is strictly positive), we have strong duality, i.e. $p^* = d^*$ and $p^*_{2s} = d^*_{2s}$ [Lasserre, 2001, Schmüdgen, 2017]. However, it is difficult to verify the conditions analytically. In practice, strong duality is observed in most cases, so in this work we refer to solving the primal and dual interchangeably, as $p^*_{2s} = d^*_{2s}$ in all cases we encounter where a SDP solver returns a feasible point.

6.5 Proofs

6.5.1 Primal-dual relationship of S-SOS

Regular SOS

Global polynomial optimization can be framed as the following lower-bound maximization problem where we need to check global non-negativity:

$$\begin{aligned} \sup_{c \in \mathbb{R}} \quad & c \\ \text{s.t.} \quad & f(x) - c \geq 0 \quad \forall x \end{aligned} \tag{6.1}$$

When we take the SOS relaxation of the non-negativity constraint in the primal, we now arrive at the SOS primal problem, where we require $f(x) - c$ to be SOS which guarantees

non-negativity but is a stronger condition than necessary:

$$\begin{aligned} \sup_{c\in\mathbb{R}} \quad & c \\ \text{s.t.} \quad & f(x) - c \in \mathcal{P}_{\text{SOS}}(X). \end{aligned} \tag{6.2}$$

The dual to Equation (6.1) is the following moment-minimization problem:

$$\begin{aligned} \inf_{\mu\in\mathbb{P}(X)} \quad & \int f(x)\mathrm{d}\mu(x) \\ \text{with} \quad & \int \mathrm{d}\mu(x) = 1. \end{aligned} \tag{6.3}$$

Taking some spanning basis $m_s(x) : \mathbb{R}^n \to \mathbb{R}^{a(n,s)}$ of monomials up to degree s, we have the moment matrix $M \in \mathbb{R}^{a(n,s)\times a(n,s)}$:

$$M_{i,j} = \int m_i(x)m_j(x)d\mu(x) = y_\alpha$$

where we introduce a moment vector y whose elements correspond to the unique moments of the matrix M. Then we may write the degree-$2s$ moment-minimization problem, which is now in a solvable numerical form:

$$\begin{aligned} \inf_{y} \quad & \sum_{\alpha} f_\alpha y_\alpha \\ \text{with} \quad & M(y)_{1,1} = 1 \\ & M(y) \succcurlyeq 0 \end{aligned} \tag{6.4}$$

where we write $M(y)$ as the matrix formed by placing the moments from y into their appropriate places and we set the first element of $m_s(x)$ to be 1, hence $M_{1,1} = \int d\nu(x) = 1$ is simply the normalization constraint. For further reading, see [Nie, 2009, Lasserre, 2001].

Stochastic SOS

Now let us lift this problem into the stochastic setting with parameters ω sampled from a given distribution ν, i.e. replacing $x \to (x, \omega)$. We need to make some choice for the objective. The expectation of the lower bound under $\nu(\omega)$ is a reasonable choice, i.e.

$$\int_\Omega c(\omega) d\nu(\omega)$$

but we could also make other choices, such as ones that encourage more robust lower bounds. In this work however, we formulate the primal S-SOS as below (same as Equation (3.1)):

$$\begin{aligned} p^* = \sup_{c \in L^1(\Omega)} \quad & \int c(\omega) \mathrm{d}\nu(\omega) \\ \text{s.t.} \quad & f(x, \omega) - c(\omega) \geq 0 \end{aligned} \tag{6.5}$$

Note that if the ansatz space for the function $c(\omega)$ is general enough, the maximization of the curve c is equivalent to a pointwise maximization, i.e. we recover the best approximation for almost all ω. Then the dual problem has a very similar form to the non-stochastic case.

Theorem 6.5.1. *The dual to Equation* (6.5) *is the following moment minimization where* $\mu(x, \omega)$ *is a probability measure on* $X \times \Omega$*:*

$$\begin{aligned} \inf_{\mu \in \mathbb{P}(X \times \Omega)} \quad & \int f(x, \omega) \mathrm{d}\mu(x, \omega) \\ \textit{with} \quad & \int_{X \times \Omega} \omega^\alpha \mathrm{d}\mu(x, \omega) = \int_\Omega \omega^\alpha \mathrm{d}\nu(\omega) \quad \textit{for all } \alpha \in \mathbb{N}^d. \end{aligned}$$

Remark 6.5.2. Notice, that the condition $\int_{X \times \Omega} \omega^\alpha \mathrm{d}\mu(x, \omega) = \int_\Omega \omega^\alpha \mathrm{d}\nu(\omega)$ implies that the first marginal of μ is the noise distribution ν. Let μ_ω denote the disintegration of μ with respect to ν, [Ambrosio et al., 2005]. Then the moment matching condition is equivalent to $\mu_\omega(X) = 1$ for almost all ω and μ being a Young measure w.r.t. ν. The idea is that $\mu_\omega(x)$ is

a minimizing density for every single configuration of ω.

Proof. We use $\mathcal{P}_{\geq 0}(X \times \Omega)$ to denote the space of non-negative polynomials on $X \times \Omega$. Given measure ν on Ω and polynomial function $p : X \times \Omega \to \mathbb{R}$ consider

$$\begin{aligned} &\sup_{\substack{\gamma \in L^1(\Omega,\nu) \\ q \in \mathcal{P}_{\geq 0}(X \times \Omega).}} \int_\Omega \gamma(\omega) \mathrm{d}\nu(\omega) \\ \text{s.t} \quad & p(x,\omega) - \gamma(\omega) = q(x,\omega) \end{aligned}$$

This is equivalent to

$$- \inf_{\substack{\gamma \in L^1(\Omega,\mu) \\ q \in \mathcal{P}_{\geq 0}(X \times \Omega)}} f(\gamma, q) + g(\gamma, q)$$

with

$$f(\gamma, q) = -\int_\Omega \gamma(\omega) \mathrm{d}\nu(\omega)$$

and

$$g(\gamma, q) = -\chi_{\{f-\gamma-q=0\}} = \begin{cases} 0 \text{ if } f - \gamma - q = 0 \\ -\infty \text{ else} \end{cases},$$

i.e. g is the characteristic function enforcing non-negativity.

Denote by h^* the Legendre dual, i.e.

$$h^*(y) = \sup_x \langle x, y \rangle - h(x).$$

Then by Rockafellar duality, [Ekeland and Temam, 1976, Rockafellar, 2015], and noting that

signed Borel measures $\mathcal{B}$ are the dual to continuous functions, the dual problem reads

$$\sup_{\Gamma\in L^\infty(\Omega,\mu),\mu\in\mathcal{B}} -f^*(\Gamma,\mu) - g^*(-(\Gamma,\nu))$$

and we would have

$$\sup_{\Gamma\in L^\infty(\Omega,\mu),\mu\in\mathcal{B}} -f^*(\Gamma,\mu) - g^*(-(\Gamma,\mu)) = -\inf_{\substack{\gamma\in L^1(\Omega,\mu)\\ q\in\mathcal{P}_{\geq 0}(X\times\Omega)}} f(\gamma,q) + g(\gamma,q).$$

The Legendre duals of f and g can be explicitly calculated as

$$f^*(\Gamma,\mu) = \begin{cases} 0 \text{ if } \Gamma = -1 \text{ and } \mu \leq 0 \\ \\ \infty \text{ else} \end{cases}$$

and

$$g^*(\Gamma,\mu) = \begin{cases} \int_{\Omega\times X} f(x,\omega)\mathrm{d}\mu(\omega,x) & \text{if } f-\gamma \in \mathcal{P}_{\geq 0}(X\times\Omega) \text{ and } \Gamma(\omega) = \mu_\omega(X) \\ \\ \infty & \text{else} \end{cases}$$

since

$$\begin{aligned} f^*(\Gamma,\mu) &= \sup_{\gamma,q}\left(\int_\Omega \gamma(\omega)\Gamma(\omega)\mathrm{d}\nu(\omega) + \int_{\Omega\times X} q(x,\omega)\mathrm{d}\mu(x,\omega) - f(\gamma,q)\right) \\ &= \sup_{\gamma,q}\int_\Omega \gamma(\omega)(\Gamma(\omega)+1)\mathrm{d}\nu(\omega) + \int_{\Omega\times X} q(x,\omega)\mathrm{d}\mu(x,\omega) \\ &= \begin{cases} 0 & \text{if } \Gamma = -1 \text{ and } \mu \leq 0 \\ \\ \infty & \text{else} \end{cases} \end{aligned}$$

and

$$
\begin{aligned}
g^*(\Gamma,\mu) &= \sup_{\gamma,q} \int_\Omega \gamma(\omega)\Gamma(\omega)\mathrm{d}\nu(\omega) + \int_{\Omega\times X} q(x,\omega)\mathrm{d}\mu(\omega,x) + \chi_{\{f-\gamma-q=0\}} \\
&= \begin{cases} \sup_\gamma \int_\Omega \gamma(\omega)\Gamma(\omega)\mathrm{d}\nu(\omega) + \int_{\Omega\times X} (f(x,\omega)-\gamma(\omega))\mathrm{d}\mu(\omega,x) & \text{if } f-\gamma =\in \mathcal{P}_{\geq 0}(X\times\Omega) \\ \infty & \text{else} \end{cases} \\
&= \begin{cases} \sup_\gamma \int_\Omega \gamma(\omega)(\Gamma(\omega)-\mu_\omega(X))\mathrm{d}\nu(\omega) + \int_{\Omega\times X} (f(x,\omega)\mathrm{d}\mu(\omega,x) & \text{if } f-\gamma \in \mathcal{P}_{\geq 0}(X\times\Omega) \\ \infty & \text{else} \end{cases} \\
&= \begin{cases} \int_{\Omega\times X} (f(x,\omega)\mathrm{d}\mu(\omega,x) & \text{if } f-\gamma \in \mathcal{P}_{\geq 0}(X\times\Omega) \text{ and } \Gamma(\omega)=\mu_\omega(X) \\ \infty & \text{else} \end{cases}
\end{aligned}
$$

Altogether, we get

$$
-f^*(\Gamma,\mu) - g^*(-\Gamma,-\mu) = \begin{cases} \int_{\Omega\times X} f(x,\omega)\mathrm{d}\mu(\omega,x) & \text{if } \mu_\omega(X)=1 \\ \infty & \text{else.} \end{cases}
$$

□

6.5.2 Convergence of S-SOS hierarchy

Lemma on approximating polynomials

Lemma 6.5.3. *Let Ω be compact and $g:\Omega\to\mathbb{R}^n$ be Lipschitz continuous. Then there is a trigonometric polynomial g_s of degree s and a constant $C>0$ depending only on Ω and n such that*

$$g \geq g_s$$

and

$$\|g - g_s\|_{L^2(\Omega)} \leq \frac{1+\ln(s)}{s} C \|g\|_{H^1(\Omega)}.$$

One cannot expect much more as the following example shows:

Example 6.5.4. Consider $g : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}$ defined by

$$g(x,p,q) = (x^2 + px + q)^2.$$

Then we have for every $(p,q) \in \mathbb{R}^2$ that

$$\inf_{x\in\mathbb{R}} g(x,p,q) = \begin{cases} 0 & \text{if } \frac{p^2}{4} \geq q \\ (\frac{p^2}{4} - q)^2 & \text{else.} \end{cases}$$

Therefore, $(p,q) \mapsto \inf_{x\in\mathbb{R}} g(x,p,q)$ is once differentiable but not twice. □

Convergence at $\ln s/s$ rate

Theorem 6.5.1 (Asymptotic convergence of S-SOS)**.** *Let $f : [0,1]^n \times \Omega \to \mathbb{R}$ be a trigonometric polynomial of degree $2r$, $c^*(\omega) = \inf_x f(x,\omega)$ the optimal lower bound as a function of ω, and ν any probability measure on compact $\Omega \subset \mathbb{R}^d$. Let $s = (s_x, s_\omega, s_c)$, referring separately to the degree of the basis in x terms, the degree of the basis in ω terms, and the degree of the lower-bounding polynomial $c(\omega)$.*

Let $c^*_{2s}(\omega)$ be the lower bounding function obtained from the primal S-SOS SDP with $m_s(x,\omega)$ a spanning basis of trigonometric monomials with degree $\leq s_x$ in x terms and of degree $\leq s_\omega$ in ω terms:

$$\begin{aligned} p^*_{2s} = \sup_{c\in\mathcal{P}^{2s_c}(\Omega), W\succcurlyeq 0} & \int c(\omega) \mathrm{d}\nu(\omega) \\ s.t. \quad & f(x,\omega) - c(\omega) = m_s(x,\omega)^T W m_s(x,\omega) \end{aligned}$$

Then there is a constant $C > 0$ depending only on Ω, d, and n such that for all $s_\omega, s_x \geq \max\{3r, 3s_c\}$ the following holds:

$$\int_\Omega [c^*(\omega) - c^*_{2s}(\omega)] \, \mathrm{d}\nu(\omega) \leq |\Omega| \epsilon(f, s)$$

$$\begin{aligned}\varepsilon(f,s) \leq & \|f - \bar{f}\|_F \left[1 - \left(1 - \frac{6r^2}{s_\omega^2}\right)^{-d} \left(1 - \frac{6r^2}{s_x^2}\right)^{-n}\right] \\ & + \|c^* - \bar{c}^*\|_F \left[1 - \left(1 - \frac{6r^2}{s_\omega^2}\right)^{-d}\right] + C\frac{(1 + \ln(2s_c))}{2s_c}.\end{aligned}$$

where $\bar{f}$ denotes the average value of the function f over $[0,1]^n$, i.e. $\bar{f} = \int_{[0,1]^n} f(x)dx$ and $||f(x)||_F = \sum_{\hat{x}} |\hat{f}(\hat{x})|$ denotes the norm of the Fourier coefficients.

$\epsilon(f, s)$ bounds the expected error, giving us asymptotic convergence as $s = \min(s_x, s_\omega, s_c) \to \infty$. Note the first two terms give a $O(\frac{1}{s^2})$ convergence rate. However, the overall error will be dominated by the degree of $c(\omega)$ (from the third term) hence our convergence rate is $O(\frac{\ln s}{s})$.

Proof. By the convergence of Fourier series [Jackson, 1930] we have the existence of a trigonometric polynomial g' of degree s with

$$\|g - g'\|_{L^1(\Omega)} \leq \frac{C'}{s} \|g\|_{H^1(\Omega)}$$

as well as

$$\|g - g'\|_\infty \leq L_g \frac{\ln(s)}{s}.$$

Then we define $g_s = g' - \|g - g'\|_\infty$ and hence $g \geq g_s$. Furthermore,

$$\|g - g_s\|_{L^1(\Omega)} \leq \frac{(C' + |\Omega| \ln s)}{s} L_g.$$

Writing $C(\Omega) = \max\{C', |\Omega|\}$ we have the desired form where $|\Omega|$ is the volume of Ω. □

Proof of Theorem 6.5.1. Let $\Omega \subset \mathbb{R}^d$ be compact and $f : R^n \times \Omega \to \mathbb{R}$ be a 1-periodic trigonometric polynomial (t.p.) of degree $\leq 2r$. We then make Ω isomorphic to $[0,1]^d$ and hereafter consider $\Omega = [0,1]^d$ and $f : [0,1]^n \times [0,1]^d \to \mathbb{R}$. Let $\varepsilon > 0$ and $b = \frac{\varepsilon}{2}$. Let the best lower bound be

$$c^*(\omega) = \inf_{x \in X} f(x, \omega).$$

Proof outline. We split the error into two parts. First, we use the fact that there is a lower-bounding t.p. c_a^* of degree s_c such that

$$\|c^* - c_a^*\| \leq C \frac{1 + \ln s_c}{s_c}$$

and

$$c^* \geq c_a^*.$$

This will provide us with a degree-s_c t.p. approximation to the lower bounding function, which in general is only known to be Lipschitz continuous.

Next, we show, that for any $b > 0$ there is a degree-$2s$ SOS t.p. $f_{\text{SOS}}(x, \omega)$ such that

$$f_{\text{SOS}} = f - (c_a^* - b).$$

We write $s = (s_x, s_\omega)$ where s_x, s_ω denotes the respective max degrees in the variables x, ω. Once we have constructed this, we can compute $f - f_{\text{SOS}} = c_a^* - \varepsilon$ and since we know that $f_{\text{SOS}} \geq 0$ everywhere and $c_a^* - \varepsilon$ is some degree-s_c t.p. we have found a degree-s_c lower-bounding t.p. The construction of this SOS t.p. adds another error term. If we can drive $\varepsilon \to 0$ as $\bar{s} = \min(s_x, s_\omega, s_c) \to \infty$ then we are done.

Proof continued. To that end, let $c_a^* : \Omega \to \mathbb{R}$ be the best degree-s_c trigonometric

approximation of c^* with respect to L^1 such that

$$c^* \geq c_a^*.$$

By [Clarke, 1975], we know that c^* is locally Lipschitz continuous with Lipschitz constant L_{c^*} and hence, by Lemma 6.5.3 we get that there is $C(\Omega) > 0$ such that

$$\|c^* - c_a^*\|_{L^1\Omega)} \leq C(\Omega)\frac{1 + \ln s_c}{s_c} L_{c^*}.$$

Next we introduce $c_{2s}^*(\omega)$ which is some degree-$2s$ t.p. After an application of the triangle inequality and Cauchy-Schwarz on the integrated error term $\int_\Omega |c^* - c_{2s}^*| d\omega$ we have

$$\int_\Omega \left| \inf_{x \in X} f(x, \omega) - c_{2s}^*(\omega) \right| \mathrm{d}\omega \leq \int_\Omega |c_a^*(\omega) - c_{2s}^*(\omega)| \mathrm{d}\omega + |\Omega| \|c^* - c_a^*\|_{L^2(\Omega)}$$

$$\int_\Omega \left| \inf_{x \in X} f(x, \omega) - c_{2s}^*(\omega) \right| \mathrm{d}\omega \leq \underbrace{\int_\Omega |c_a^*(\omega) - c_{2s}^*(\omega)| \mathrm{d}\omega}_{\text{gap between some SDP solution } c_{2s}^*(\omega) \text{ and t.p. } c_a^*(\omega)}$$

$$+ \quad \underbrace{C(\Omega)\frac{1 + \ln s_c}{s_c} L_{c^*}}_{\text{approx. error of L-contin. fn.}}$$

Now we want to show that for any $\varepsilon > 0$ we can construct a degree-$2s$ SOS trigonometric polynomial $f_{\mathrm{SOS}}(x, \omega)$ such that

$$f_{\mathrm{SOS}} = f - c_a^* + b.$$

with $b = \varepsilon/2$ and $s = (s_x, s_\omega) > r$. We can then set $f - f_{\mathrm{SOS}} = c_a^* - b = c_{2s}^*$ as the degree-$2s$ lower-bounding function. If we can drive $b = \varepsilon/2 \to 0$ as $s, s_c \to \infty$ we are done, as by

construction $|c_a^* - c_{2s}^*| = b$.

Observe that by assumption $f - c_a^* + b$ is a t.p. in (x, ω) where f is degree-$2r$ and c_a^* is degree $s_c \geq 2r$. Denote by $(f - f_*^a + b)_\omega$ its coefficients w.r.t the ω basis. Note that the coefficients are functions in x. Following the integral operator proof methodology in [Bach and Rudi, 2023], define the integral operator T to be

$$Th(x, \omega) = \int_{X \times \Omega} |q_\omega(\omega - \bar{\omega})|^2 |q_x(x - \bar{x})|^2 h(\bar{x}, \bar{\omega}) \mathrm{d}\bar{x} \mathrm{d}\bar{\omega},$$

where q_ω is a trigonometric polynomial in ω of degree $\leq s_\omega$ and q_x is a trigonometric polynomial in x of degree $\leq s_x$. The intuition is that this integral operator explicitly builds a SOS function of degrees (s_x, s_ω) out of any non-negative function h by hitting it against the kernels q_x, q_ω.

We want to find a positive function $h : X \times \Omega \to \mathbb{R}$ such that

$$Th = f - c_a^* + b.$$

In frequency space, the Fourier transform turns a convolution into pointwise multiplication so we have:

$$\widehat{Th}(\hat{x}, \hat{\omega}) = \hat{q}_\omega * \hat{q}_\omega(\hat{\omega}) \cdot \hat{q}_x * \hat{q}_x(\hat{x}) \cdot \hat{h}(\hat{x}, \hat{\omega}).$$

In the Fourier domain it is easy to write down the coefficients of $\hat{h}$:

$$\hat{h}(\hat{x}, \hat{\omega}) = \begin{cases} 0 & \text{if } \|\hat{x}, \hat{\omega}\|_\infty > \max\{2r, 2s_c\} \\ \dfrac{\hat{f}(\hat{x}, \hat{\omega}) - \hat{c}_a^*(\hat{\omega}) 1_{\hat{x}=0} + b 1_{\hat{x}=0} 1_{\hat{\omega}=0}}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega}) \cdot \hat{q}_x * \hat{q}_x(\hat{x})} & \text{otherwise.} \end{cases}$$

Computing $Th - h$ gives:

$$\begin{aligned}
&f(x,\omega) - c_a^*(\omega) + b - h(x,\omega) \\
=&\sum_{\hat{\omega},\hat{x}} \hat{f}(\hat{x},\hat{\omega}) \left(1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega}) \cdot \hat{q}_x * \hat{q}_x(\hat{x})}\right) \exp(2i\pi\hat{\omega}^T\omega) \exp(2i\pi\hat{x}^T x) \\
&+ \sum_{\hat{\omega}} (b1_{\hat{\omega}=0} - c_a^*) \left(1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega})}\right) \exp(2i\pi\hat{\omega}^T\omega)
\end{aligned}$$

and thus after requiring $\hat{q}_\omega * \hat{q}_\omega(0) = \hat{q}_x * \hat{q}_x(0) = 1$ we have:

$$\begin{aligned}
&\max_{x,\omega} |f(x,\omega) - c_a^*(\omega) + b - h(x,\omega)| \\
\leq&\|f - \bar{f}\|_F \max_{\hat{\omega}\neq 0} \max_{\hat{x}\neq 0} \left|1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega}) \cdot \hat{q}_x * \hat{q}_x(\hat{x})}\right| \\
&+ \max_{\hat{\omega}\neq 0} \|c_a^* - \bar{c}_a^*\|_F \left|1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega})}\right|.
\end{aligned}$$

As a reminder, because $c^* \geq c_a^*$ everywhere we have $f - c_a \geq f - c^* \geq 0$ or $f - c_a^* + b > 0$, since $b = \varepsilon/2 > 0$. Since $Th = f - c_a^* + b > 0$ and it is a SOS, we need to guarantee $h > 0$.

If $\max_{x,\omega} |f(x,\omega) - f_*^a(\omega) + b - h(x,\omega)| \leq b$ then

$$\max_{x,\omega} |Th - h| < b.$$

Since $Th \geq b$ and $b > 0$ we have

$$h = Th + h - Th \geq Th - \|h - Th\|_\infty \geq b - b \geq 0$$

and hence $h > 0$ if we ensure $\max_{x,\omega} |Th - h| \leq b$.

Now let us show that

$$\max_{x,\omega} |f(x,\omega) - c_a^*(\omega) + b - h(x,\omega)| \leq b$$

can be ensured if $s = (s_x, s_\omega)$ is large enough.

Using the same kernel and bounds as in [Bach and Rudi, 2023], we choose for $z \in \{x, \omega\}$ the triangular kernel such that

$$\hat{q}_z(\hat{z}) = \left(1 - \frac{6r^2}{z^2}\right)_+^d \prod_{i=1}^d \left(1 - \frac{|\hat{z}_i|}{s_{x,\omega}}\right)_+ .$$

Note that $(x)_+ = \max(x, 0)$. Then we have

$$\begin{aligned}
&\max_x |f(x,\omega) - c_a^*(\omega) + b - h(x,\omega)| \\
\leq &\|f - \bar{f}\|_F \max_{\hat{\omega},\hat{x}} \left|1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega}) \cdot \hat{q}_x * \hat{q}_x(\hat{x})}\right| + \|c_a^* - \bar{c}_a^*\|_F \max_{\hat{\omega}} \left|1 - \frac{1}{\hat{q}_\omega * \hat{q}_\omega(\hat{\omega})}\right| \\
\leq &\|f - \bar{f}\|_F \left|1 - \left(1 - \frac{6r^2}{s_\omega^2}\right)^{-d} \left(1 - \frac{6r^2}{s_x^2}\right)^{-n} \right| + \|c_a^* - \bar{c}_a^*\|_F \left|1 - \left(1 - \frac{6^2}{s_\omega^2}\right)^{-d} \right|
\end{aligned}$$

Therefore, by choosing s_ω and s_x large enough such that

$$\|f - \bar{f}\|_F \left|1 - \left(1 - \frac{6r^2}{s_\omega^2}\right)^{-d} \left(1 - \frac{6r^2}{s_x^2}\right)^{-n} \right| + \|c_a^* - \bar{c}_a^*\|_F \left|1 - \left(1 - \frac{6^2}{s_\omega^2}\right)^{-d} \right| \leq b = \frac{\varepsilon}{2}$$

we have

$$h \geq 0$$

and thus Th is SOS. By design we have

$$c_a^* - c_{2s}^* \leq b$$

and thus

$$\int_\Omega |c_a^* - c_{2s}^*| \mathrm{d}\omega \leq \frac{\varepsilon}{2}.$$

Recalling

$$\int_\Omega \left| \inf_{x\in X} f(x,\omega) - c^*_{2s}(\omega)\right| \mathrm{d}\omega \leq \underbrace{\int_\Omega |c^*_a(\omega) - c^*_{2s}(\omega)|\mathrm{d}\omega}_{\text{gap between some SDP solution } c^*_{2s}(\omega) \text{ and t.p. } c^*_a(\omega)}$$

$$+ \underbrace{C(\Omega)\frac{1+\ln s_c}{s_c} L_{c^*}}_{\text{approx. error of L-contin. fn.}}$$

we can additionally choose s_c large enough to guarantee

$$C(\Omega)\frac{1+\ln s_c}{s_c} L_{c^*} \leq \frac{\varepsilon}{2}$$

and then we are done.

Setting $s_x, s_\omega, s_c = s$ and sending $s \to \infty$ we have asymptotic behavior of the final error expression:

$$\boxed{\int_\Omega \left| \inf_{x\in X} f(x,\omega) - c^*_{2s}(\omega)\right| \mathrm{d}\omega \leq C_1\frac{1}{s^2} + C_2\frac{1}{s} + C_3\frac{\ln s}{s} = \mathcal{O}\left(\frac{\ln s}{s}\right)}$$

with the constants C_1, C_2, C_3 depending on $r, n, d, \|f - \bar{f}\|_F, \|c_a - \bar{c}^*_a\|_F, \Omega$ and L_{c^*}. □

Convergence at $1/s$ rate

Proof of Proposition 3.2.2. Let $c^*_a(\omega)$ be a piecewise-constant approximation of $c^*(\omega) = \inf_x f(x,\omega)$ on equidistant grid-points. Then $\|c^* - c^*_a\|_{L^1\Omega} \leq C\frac{1}{s_p}$ where s_p is the number of grid points ω_i. Let

$$c^*_s(\omega) = \sum c^*_s(\omega_i) 1_{[\omega_i, \omega_{i+1}]}$$

where $c_s^*(\omega_i)$ is the best lower bound (resulting from regular SOS) of degree s of $x \mapsto f(x, \omega_i)$. Then we have $c_a^*(\omega_i) - c_s^*(\omega_i)$ can be bounded by

$$\max_{\omega_i} \|f(\omega_i, \cdot) - \bar{f}(\omega_i, \cdot)\|_F \left(1 - \left(1 - \frac{6r^2}{s^2}\right)^{-n}\right)$$

by [Bach and Rudi, 2023]. Then

$$\int_\Omega c^*(\omega) - c_s^*(\omega) \mathrm{d}\omega \leq \sum_i |c_a^*(\omega_i) - c_s^*(\omega_i)||\Delta(\omega_i)| + \|c^* - c_a^*\|_{L^1(\Omega)}.$$

Using the same bound we get for the first term from the proof of Theorem 6.5.1, we can reduce the first term to a $O(1/s^2)$ dependence and we use the theorem on the L^1 convergence of piecewise-constant approximation to 1-periodic trigonometric polynomials from [Jackson, 1930] for the second:

$$\int_\Omega c^*(\omega) - c_s^*(\omega) \mathrm{d}\omega \leq \max_{\omega_i} \|f(\omega_i, \cdot) - \bar{f}(\omega_i, \cdot)\|_F \left(1 - \left(1 - \frac{6r^2}{s^2}\right)^{-n}\right) |\Omega| + \frac{C}{s_p}$$

Note that the $1/s_p$ term dominates in the resulting expression and thus we have the desired result. □

6.6 S-SOS for a simple quadratic potential

We provide a simple application of S-SOS to a simple quadratic potential that admits a closed-form solution so as to demonstrate its usage and limitations.

6.6.1 *Analytic solution for the lower bounding function $c^*(\omega)$ with $\omega \sim$ Uniform$(-1, 1)$*

Let $x \in \mathbb{R}$ and $\omega \sim \text{Uniform}(-1, 1)$. Suppose that we have

$$f(x, \omega) = (x - \omega)^2 + (\omega x)^2$$

In this case we may explicitly evaluate the exact minimum function $c^*(\omega) = \inf_x f(x; \omega)$. Note that

$$f(x; \omega) = x^2 - 2\omega x + \omega^2 + \omega^2 x^2$$

Explicitly evaluating the zeros of the first derivative we have

$$\partial_x f(x; \omega) = 2x^* - 2\omega + 2\omega^2 x^* = 0$$

$$x^*(1 + \omega^2) = \omega$$

$$x^* = \frac{\omega}{1 + \omega^2}$$

and, thus,

$$c^*(\omega) = \inf_x f(x; \omega) = \frac{\omega^4}{1 + \omega^2}.$$

Note that despite $f(x, \omega)$ being a simple degree-2 SOS polynomial, the tightest lower-bound $c^*(\omega) = \inf_x f(x, \omega)$ is explicitly not polynomial. However, it is algebraic, as it is defined implicitly as the root of the polynomial equation

$$c^*(\omega)(1 + \omega^2) - \omega^4 = 0$$

6.6.2 Degree-2s S-SOS to find a polynomial lower-bounding function $c^*_{2s}(\omega)$

Observe that the tightest lower-bounding function $c^*(\omega)$ is not polynomial even in this simple setting. However, we can relax the problem to trying to find $c_{2s} \in \mathcal{P}^{2s}(\Omega)$ to obtain a weaker bound with $\inf_x f(x,\omega) = c^*(\omega) \geq c_{2s}(\omega)$.

We now proceed with formulating and solving the degree-2s primal S-SOS SDP (Equation (3.2)). We assume that $c_{2s}(\omega)$ is parameterized by a polynomial of degree $\leq 2s$ in ω. Observe that this class of functions is not large enough to contain the true function $c^*(\omega)$.

We choose $s \in \{2, 4\}$ and use the standard monomial basis in x, ω, we have the feature maps $m_2(x,\omega) : \mathbb{R}^2 \to \mathbb{R}^6$ and $m_4(x,\omega) : \mathbb{R}^2 \to \mathbb{R}^{15}$, since there are $\binom{n+s}{s}$ unique monomials of up to degree-s in n variables. These assumptions together enable us to explicitly write a SOS SDP in terms of coefficient matching. Note that we must assume some noise distribution $\nu(\omega)$. For this section, we present results assuming $\omega \sim \text{Uniform}(-1, 1)$. We solve the resulting SDP in CVXPY using Legendre quadrature with $k = 5$ zeroes on $[-1, 1]$ to evaluate the objective $\int c(\omega) d\nu(\omega)$. In fact, k sample points suffice to exactly integrate polynomials of degree $\leq 2k - 1$.

We solve the SDP for two different levels of the hierarchy, $s = 2$ and $s = 4$ (producing lower-bound polynomials of degree 4 and 8 respectively), and plot the lower bound functions $c_{2s}(\omega)$ vs the true lower bound $c^*(\omega) = \omega^4/(1 + \omega^2)$ as well as the optimality gap to the true lower bound in Fig.6.1.

6.6.3 Convergence of lower bound as degree s increases

To solve the S-SOS SDP in practice, we must choose a maximum degree $2s$ for the SOS function $m_2(x,\omega)^T W m_2(x,\omega)$ and the lower-bounding function $c(\omega)$, which are both restricted to be polynomials. Indeed, a larger s not only increases the dimension of our basis function $m_s(x,\omega)$ but also the complexity of the resulting SDP. We would expect that $d^*_{2s} \to d^*$ as $s \to \infty$, i.e. the optimal value of the degree-2s S-SOS SDP (Equation (3.4)) converges to

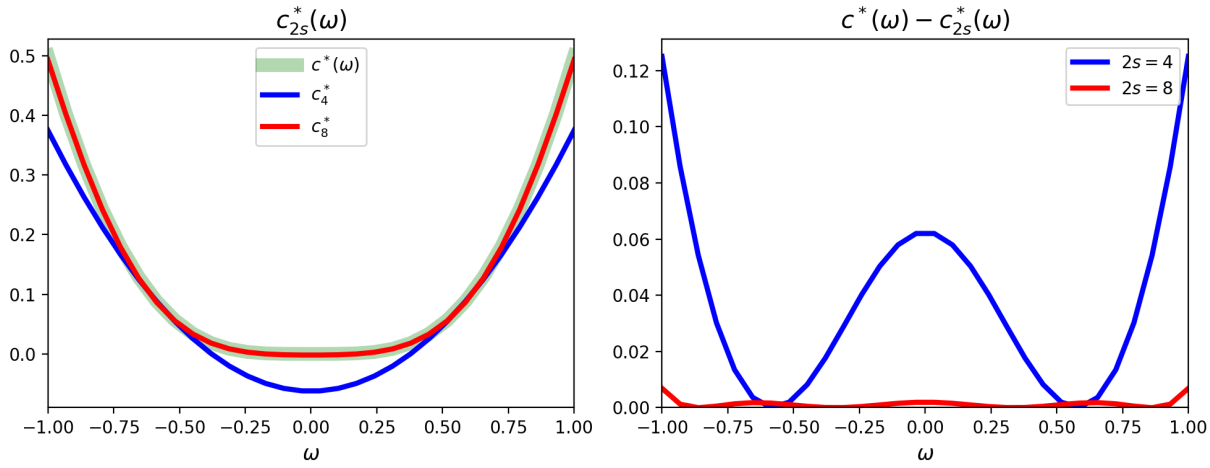

Figure 6.1: Lower bound functions for basis function degree $d = 2, 4$ (left) and the optimality gap to the true lower bound $c^*(\omega) - c^*_{2s}(\omega)$ (right)

that of the "minimizing distribution" optimization problem (Equation (3.3)).

In particular, note that in the standard SOS hierarchy we typically find finite convergence (exact agreement at some degree $2s^* < \infty$). However, in S-SOS, we thus far have only a guarantee of asymptotic convergence, as each finite-degree S-SOS SDP solves for a polynomial approximation to the optimal lower bound $c^*(\omega) = \inf_{x \in X} f(x, \omega)$. In Figure 3.1, we illustrate the primal S-SOS SDP objective values

$$p^*_{2s} = \sup_{c \in \mathcal{P}^{2s}(\Omega)} \int c(\omega) d\nu(\omega) \quad \text{with} \quad f(x, \omega) - c(\omega) \in \mathcal{P}^{2s}_{\text{SOS}}(X \times \Omega)$$

for a given level of the hierarchy (a chosen degree s for the basis $m_s(x, \omega)$) and their convergence towards the optimal objective value

$$\int c^*(\omega) d\nu(\omega) = \frac{\pi}{4} - \frac{2}{3} \approx 0.1187$$

for the simple quadratic potential, assuming $\nu(\omega) = \frac{1}{2}$ with $\omega \sim \text{Uniform}(-1, 1)$. We note that in the log-linear plot (right) we have a "hinge"-type curve, with a linear decay (in

logspace) and then flattening completely. This suggests perhaps that in realistic scenarios the degree needed to achieve a close approximation is very low, lower than suggested by our bounds. The flattening that occurs here is likely due to the numerical tolerance used in our solver (CVXPY/MOSEK), as increasing the tolerance also increases the asymptotic gap and decreases the degree at which the gap flattens out.

6.6.4 Effect of different noise distributions

In the previous two sections, we assumed that $\omega \sim \text{Uniform}(-1, 1)$. This enabled us to solve the primal exactly using Legendre quadrature of polynomials. Note that in Figure 3.1 we see that the lower-bounding $c_2^*(\omega), c_4^*(\omega)$ for $\omega \sim \text{Uniform}(-1, 1)$ is a smooth polynomial that has curvature (i.e. sign matching that of the true minimum). This is actually not guaranteed, as we will see shortly.

In Figure 6.2, we present the lower-bounding functions $c_4^*(\omega)$ achieved by degree-4 S-SOS by solving the dual for $\omega \sim \text{Normal}(0, \sigma^2)$ for varying widths σ. We can see that for small $\sigma \ll 1$, the primal solution only cares about the lower-bound accuracy within a small region of $\omega = 0$, and the lower-bounding curve fails to "generalize" effectively outside the region of consideration.

6.7 S-SOS for sensor network localization

The following is a self-contained exposition of sensor network localization. Our notation and framing tend to follow that of [Nie, 2009]. Certain parts of this section have been used in the main chapter as important background.

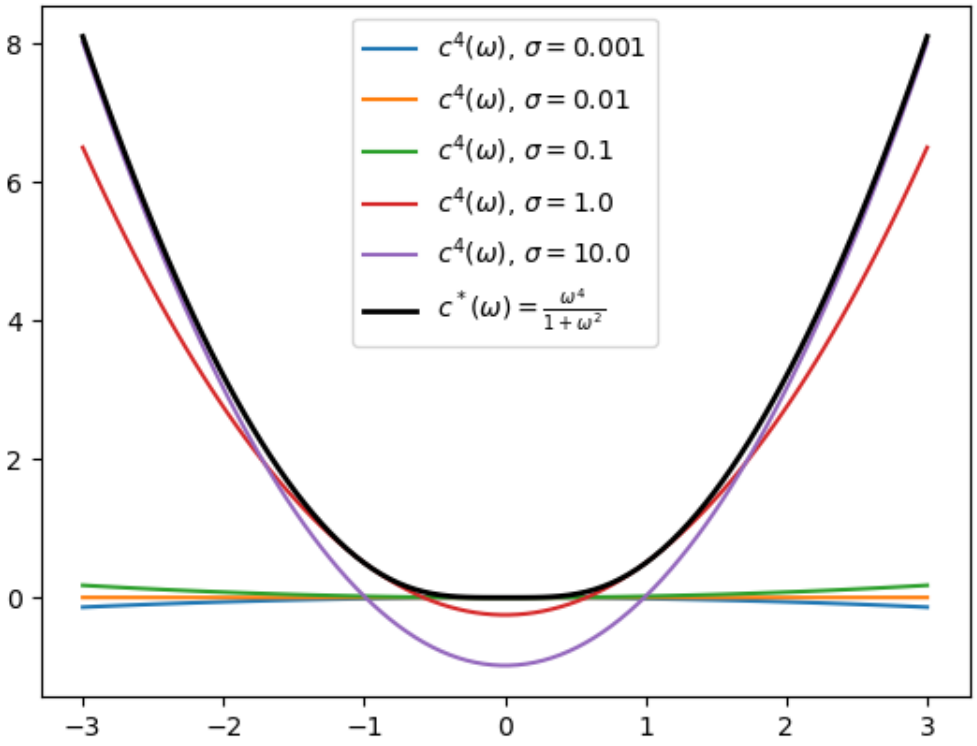

Figure 6.2: Different lower-bounding functions for degree-4 S-SOS done on the simple quadratic potential $f(x, \omega) = (x - \omega)^2 + (\omega x)^2$. The true lower-bounding function $c^*(\omega)$ is plotted in black.

6.7.1 SDP formulation

Recall the form of $f(x, \omega)$:

$$f(x, \omega; X, A, r) = \underbrace{\sum_{d_{ij} \in \mathcal{D}_{ss}(r)} (||x_i - x_j||_2^2 - d_{ij}(\omega)^2)^2}_{\text{sensor-sensor interactions}} + \underbrace{\sum_{d_{ik} \in \mathcal{D}_{sa}(r)} (||x_i - a_k||_2^2 - d_{ik}(\omega)^2)^2}_{\text{sensor-anchor interactions}}$$

Note that the function $f(x, \omega)$ is exactly a degree-4 SOS polynomial, so it suffices to choose the degree-2 monomial basis containing $a = \binom{N\ell+d+2}{2}$ elements as $m_2(x, \omega) : \mathbb{R}^{N\ell+d} \to \mathbb{R}^a$. That is, we have N sensor positions in ℓ spatial dimensions and d parameters for a total of $N\ell + d$ variables.

Let the moment matrix be $M \in \mathbb{R}^{a \times a}$ with elements defined as

$$M_{i,j} := \int m_2^{(i)}(x, \omega) m_2^{(j)}(x, \omega) d\mu(x, \omega)$$

for $i, j \in \{1, \ldots, a\}$, which fully specifies the minimizing distribution $\mu(x, \omega)$ as in Equation (3.4).

Our SDP is then of the form

$$\begin{aligned} d_4^* = \inf_y & \sum_\alpha f_\alpha y_\alpha \\ \text{s.t. } & M(y) \succcurlyeq 0 \\ & y_\alpha = m_\alpha \; \forall \; (\alpha, m_\alpha) \in \mathcal{M}_\nu \\ & y_\alpha = y_\alpha^* \; \forall \; (\alpha, y_\alpha^*) \in \mathcal{H} \end{aligned}$$

where $y_\alpha = m_\alpha$ corresponds to the moment-matching constraints of Equation (3.4) and $y_\alpha = y_\alpha^*$ correspond to any possible hard equality constraints required to set the exact position (and uncertainty) of a sensor $\mathbb{E}[x_i] = x_i^*, \mathbb{E}[x_i^2] - \mathbb{E}[x_i]^2 = 0$ for all ω. $\mathcal{M}_\nu$ represents the

$\binom{d+2s}{2s}$ moment-matching constraints necessary for all moments w.r.t. ω and $\mathcal{H}$ represents the $2\ell n$ constraints needed to set the exact positions of n known sensor positions in $\mathbb{R}^\ell$ (i.e. 1 constraint per sensor and dimension, 2 each for mean and variance).

6.7.2 Noise types

In this work we focus on the linear uniform noise case, as it is a more accurate reflection of measurement noise in true SNL problems. Special robust estimation approaches may be needed to properly handle the outlier noise case.

- **Linear uniform noise**: for a subset of edges we write $d_{ij,k}(\omega) = d^*_{ij} + \epsilon\omega_k$, $\omega_k \sim$ Uniform$(-1, 1)$, and $\epsilon \geq 0$ some noise scale we set. The same random variate ω_k may perturb any number of edges. Otherwise the observed distances are the true distances.

- **Outlier uniform noise**: for a subset of edges we ignore any information in the actual measurement $d_{ij,k} = \omega_k$, $\omega_k \sim$ Uniform$(0, 2\sqrt{\ell})$ where ℓ is the physical dimension of the problem, i.e. $x_i \in \mathbb{R}^\ell$.

6.7.3 Algorithms: S-SOS and MCPO

Here we explicitly formulate MCPO and S-SOS as algorithms. Let $X = \mathbb{R}^n, \Omega = \mathbb{R}^d$ and use the standard monomial basis. We write $z = [x_1, \ldots, x_n, \omega_1, \ldots, \omega_d]$. Our objective is to approximate $c^*(\omega) = \inf_x f(x, \omega)$ for all ω, with a view towards maximizing $\int c^*(\omega) d\nu(\omega)$ for ω sampled from some probability density $\nu(\omega)$.

MCPO (Algorithm 2) simply samples ω_t and finds a set of tuples $(x^*(\omega_t), \omega_t)$ where the optimal minimizer $(x^*(\omega_t, \omega_t)$ is computed using a local optimization scheme (we use BFGS).

S-SOS (Algorithm 3) via solving the dual (Equation (3.4)) is also detailed below.

Algorithm 2 Monte Carlo Point Optimization (MCPO)

Input: Function $f(x;\omega)$, sampler for distribution $\nu(\omega)$, number of samples T
Output: Approximate integral $\hat{I}$, empirical distribution $p_{\mathcal{D}}(x)$, empirical mean μ, empirical covariance Σ
for $t = 1$ to T **do**
    Sample $\omega_t \sim \nu(\omega)$
    Find minimizer $x_t = \min_x f(x;\omega_t)$ using BFGS
end for
Estimate integral $\hat{I} \approx \frac{1}{T}\sum_{t=1}^{T} f(x_t;\omega_t)$
Construct empirical distribution

$$p_{\mathcal{D}}(x) = \frac{1}{T}\sum_{t=1}^{T} \delta(x - x_t)$$

Calculate empirical mean $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} x_t$ and covariance $\hat{\Sigma} = \frac{1}{T-1}\sum_{t=1}^{T}(x_t-\hat{\mu})(x_t-\hat{\mu})^T$.

6.7.4 Cluster basis hierarchy

Recall from Section 3.2.2 that we defined the cluster basis hierarchy using body order b and maximum degree per variable t. In this section, we review the additional modifications needed to scale S-SOS for SNL.

In SNL, $f(x,\omega)$ is by design a degree $s = 4$ polynomial in $z = [x,\omega]$, with interactions of body order $b = 2$ (due to the (x_i, x_j) interactions) and maximum individual variable degree $t = 4$. Written this way, we want to only consider monomial terms $[x,\omega]^\alpha$ with $||\alpha||_1 \leq s$, $||\alpha||_\infty \leq 4$, and $||\alpha||_0 \leq 2$.

To sparsify our problem, we start with some k-clustering (k clusters, mutually-exclusive) of the sensor set $\mathcal{C} = \{C_1, \ldots, C_k\}$. This clustering can be considered as leveraging some kind of "coarse" information about which sensors are close to each other. For example, just looking at the polynomial $f(x,\omega)$ enables us to see which sensors (i,j) must be interacting.

Assume that there is some *a priori* clustering given to us. We denote $x^{(i)}$ as the subset of the variables restricted to the cluster C_i, i.e. $x^{(i)} = \{x_j : j \in C_i\}$. Moreover, let $G = (V,E)$ be a graph where the vertices $V = \{1,\ldots,k\}$ correspond to the k clusters and the edges

Algorithm 3 Stochastic Sum-of-squares (S-SOS), Dual formulation

Input: Maximum basis function degree $s \in \mathbb{Z}_{>0}$, complete basis function $m_s(x, \omega) : \mathbb{R}^{n+d} \to \mathbb{R}^{\binom{n+d+s}{s}}$, function $f(x; \omega) : \mathbb{R}^{n+d} \to \mathbb{R}$ represented as a dictionary mapping multi-index $\alpha \in \mathbb{Z}^{n+d}_{\geq 0} \to$ coefficient f_α, probability density function for $\nu(\omega)$ with known moments $\int \omega^\alpha d\nu(\omega) < \infty \; \forall \; ||\alpha||_1 \leq 2s$, any hard equality constraints where we want to set $x_k = x_k^*$ for some $k \in \mathcal{K}$.

Let $i1, i2, i4$ be the lexicographically-ordered arrays

$$\mathbb{Z}^{(n+d+1)\times(n+d)}_{\geq 0}, \mathbb{Z}^{\binom{n+d+s}{s}\times(n+d)}_{\geq 0}, \mathbb{Z}^{\binom{n+d+2s}{2s}\times(n+d)}_{\geq 0}$$

which correspond to the arrays of multi-indices for all degree-1, degree-s, and degree-$2s$ monomials in the variables z.

Create $M \in \mathbb{R}^{\binom{n+d+s}{s}\times\binom{n+d+s}{s}}$ as a matrix of variables to be estimated.

Create $y \in \mathbb{R}^{\binom{n+d+2s}{2s}}$ as a vector of variables to be estimated, corresponding to the vector of independent moments that fully specifies M.

Add $M \succcurlyeq 0$ constraint.

for i in length($i2$) **do**

  for j in length($i2$) **do** ▷ Require M to be formed from the elements of y.

    Compute $\alpha_{ij} = i2[i] + i2[j]$ as the multi-index corresponding to the sum of the multi-indices $i2[i], i2[j]$.

    Add constraint $M_{i,j} = y_{\alpha_{ij}}$.

  end for

end for

for each row α in $i4$ **do** ▷ Require y_α moments to equal the known moments of ω^α.

  if $\sum_{i=1}^{n} \alpha_i = 0$ **then**

    Add constraint $y_\alpha = \int z^\alpha d\nu(\omega) = \int \omega^{\alpha[-d:]} d\nu(\omega)$.

  end if

end for

for k in $\mathcal{K}$ **do** ▷ Handle any hard equality constraints in our variables x.

  Form multi-index $\alpha_1 \in \mathbb{Z}^{n+d}_{\geq 0}$ where the entry for x_k is set to 1 and everything else is zero.

  Form multi-index $\alpha_2 \in \mathbb{Z}^{n+d}_{\geq 0}$ where the entry for x_k^2 is set to 1 and everything else is zero.

  Add constraint $y_{\alpha_1} = x_k^*$. ▷ $\mathbb{E}[x_k] = x_k^*$.

  Add constraint $y_{\alpha_2} = (x_k^*)^2$. ▷ $\text{Var}[x_k] = \mathbb{E}[x_k^2] - \mathbb{E}[x_k]^2 = 0$.

end for

Form the objective to be minimized: $F = \int f(x, \omega) d\mu(x, \omega) = \sum_{\alpha \in i4} f_\alpha y_\alpha$.

Solve SDP where we compute $\inf F$ subject to above constraints.

Output: If the problem is feasible (i.e. there exists a degree-$2s$ decomposition of f into f_{SOS} and $c_{2s}^*(\omega)$), return moment matrix $M \in \mathbb{R}^{\binom{n+d+s}{s}\times\binom{n+d+s}{s}}$, dual objective value d_{2s}^*. Otherwise, terminate and return failed/infeasible SDP solve.

$E = \{(i,j) : i,j \in V\}$ correspond to known cluster-cluster interactions.

The SOS part of the function $f(x)$ may then be approximated as the sum of dense intra-cluster interactions and sparse inter-cluster interactions, where the cluster-cluster interactions are given exactly by edges in the graph G:

$$m_s(x)^T W m_s(x) \approx \sum_{i \in V} m_s(x^{(i)})^T W^{(i)} m_s(x^{(i)}) + \sum_{(i,j) \in E} m_s(x^{(i)})^T W^{(i,j)} m_s(x^{(j)})$$

where $W^{(k)}$ are symmetric PSD matrices and $W^{(i,j)}$ are rectangular matrices where we require $W^{(i,j)} = (W^{(j,i)})^T$. $m_s(x)$ for $x \in \mathbb{R}^n$ here behaves as before and denotes the basis function generated by all $\binom{n+s}{s}$ combinations of monomials with degree $\leq s$. Notice that this is a strict reduction from the standard Lasserre hierarchy at the same degree s, since in general the standard basis $m_s(x)$ on the full variable set will contain terms that mix variables from two different clusters that may not have an edge connecting them.

Efficiency gains in the SDP solve occur when we constrain certain of the off-diagonal $W^{(i,j)}$ blocks to be zero, i.e. the graph G is sparse in cluster-cluster interactions. As we can see from the block decomposition written above, this resembles block sparsity on the matrix W. We may interpret the above scheme as having a hierarchical structure out to depth 2, where we have dense interactions at the lowest level and sparse interactions aggregating them. In full generality, the resulting hierarchical sparsity in W may be interpreted as generating a chordal W, which is known to admit certain speed-ups in SDP solvers [Vandenberghe, 2017].

When attempting to solve an SNL problem in the cluster basis instead of the full basis, we need to throw away terms in the potential $f(x,\omega)$ that correspond to cross-terms that are "ignored" by the particular cluster basis we chose. The resulting polynomial $\bar{f}(x,\omega)$ has fewer terms and produces a cluster basis SDP that is easier to solve, but generally less accurate due to the sparser connectivity.

In particular, for the rows in Table 3.1 that have $N_C > 1$, we do a N_C-means clustering

of the ground-truth sensor positions and use those sensor labels to create our partitioning of the sensors. We connect every cluster using plus-one c_i, c_{i+1} (including the wrap-around one) connections, so that the cluster-cluster connectivity graph has N_C edges. We then use this information to throw out observed distances from the set $\mathcal{D}_{ss}$ and from the full basis function $m_2(x, \omega)$. See our code for complete details.

6.7.5 Hard equality constraints

The sensor-anchor terms in Equation (3.7) are added to make the problem easier, because by adding them now each sensor no longer needs to rely only on a local neighborhood of sensors to localize itself, but can also use its position relative to some known anchor. When we remove them entirely, we need to incorporate hard equality constraints between certain sensors and known "anchor" positions. This fixes certain known sensors but lets every other sensor be unrooted, defined only relative to other sensors (and potentially an anchor if it is within the sensing radius).

To deal with the equality constraints where we set the exact position of a sensor $x_i = x_i^*$, we solve the dual Equation (3.4) and implement them as equality constraints on the moment matrix, i.e. for the basis element $m_2(x, \omega)_i = x_i$ we may set $\mathbb{E}[x_i] - x_i^* = M_{0,i} - x_i^* = 0$. Note that we also need to set $\mathrm{Var}(x_i) = 0$ so for $m_2(x, \omega)_j = x_i^2$ we add the equality constraint $\mathrm{Var}(x_i) = \mathbb{E}[x_i^2] - \mathbb{E}[x_i]^2 = M_{0,j} - M_{0,i}^2 = 0$.

6.7.6 Solution extraction

Once the dual SDP has been solved, we extract the moment matrix M and can easily recover the point and uncertainty estimates for the sensor positions $\mathbb{E}[x], \mathrm{Var}[x]$ by inspecting the appropriate entries $M_{0,i}$ corresponding to $m_2(x, \omega)_i = x_i$ and $M_{0,j}$ corresponding to $m_2(x, \omega)_j = x_i^2$.

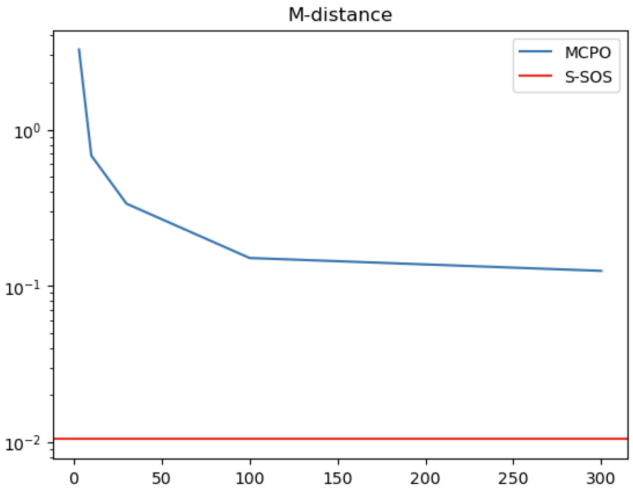

Figure 6.3: Comparison of the performance of MCPO and S-SOS (degree-4) for sensor recovery accuracy in 1D SNL with varying number of samples T used in the estimate of empirical $\hat{\mu}, \hat{\Sigma}$. M-distance is δ_M, our metric for sensor recovery accuracy per Equation (3.8). The problem type here is a $N = 5$ sensor, $\ell = 1$ spatial dimension, $|\Omega| = 2$ noise variables, $\epsilon = 0.1$ noise scale, $r = 3$ sensing radius problem. The full basis is used here for the S-SOS SDP.

6.7.7 Impact of using MCPO with varying numbers of samples T

In Figure 6.3 we can see how δ_M varies as we scale the number of samples T used in the MCPO estimate of the empirical mean/covariance of the recovered solutions. In this particular example, the runtime of the S-SOS estimate was 0.3 seconds, comparing to 30 seconds for the $T = 300$ MCPO point. Despite taking 100x longer, the MCPO solution recovery still dramatically underperforms S-SOS in δ_M. This reflects the poor performance of local optimization methods vs. a global optimization method (when it is available).

6.7.8 Scalability

The largest 2D SNL experiment we could run had $N = 15$ sensors, $N_C = 9$ clusters, and $d = 9$ noise parameters. This generated $N\ell + d = 39$ variables and 820 basis elements in the naive $m_2(x, \omega)$ construction, which was reduced to 317 after our application of the cluster basis, giving us $W, M \in \mathbb{R}^{317\times317}$. A single solve in CVXPY (MOSEK) took 30 minutes on

our workstation (2x Intel Xeon 6130 Gold and 256GB of RAM). We attempted a run with $N = 20$ sensors and $N_C = 9$ clusters and $d = 9$ noise parameters, but the process failed due to OOM constraints. Thus, we report the largest experiment that succeeded.

CHAPTER 7

CONCLUSION

In this dissertation, we have seen three very different perspectives on computational science today. In Chapter 2, we described a method to accurately and rapidly predict the EI-MS mass spectra for small molecules. Our method achieves state-of-the-art performance and speed with the limited data available, combining the data efficiency of modern graph neural networks with the chemical prior knowledge of plausible substructure enumeration. Our method, RASSP, compares favorably to first-principles simulation of mass spectra, which is too slow to generate large-scale predictions, and to other data-driven approaches, which are either end-to-end deep learning [Wei et al., 2019] or utilize more traditional ideas from cheminformatics [Allen et al., 2014, 2016, Djoumbou-Feunang et al., 2019].

However, EI-MS is only the start. EI-MS is primarily limited by the resolution of the observed spectra, which has m/z peaks observed at integer amu/Daltons. Utilizing higher-resolution data, such as that output from tandem MS/MS machines, can improve on resolution by several orders-of-magnitude and outputting significantly more data per spectrum. Our approach is distinguished from other data-driven approaches by the fusion of deep learning with substructure enumeration, enabling us to easily train and predict spectra at arbitrary resolution. Our early results in this direction indicate that many more possibilities may lie in this direction.

In Chapter 3, we reviewed an approach to parametric polynomial optimization that takes a sum-of-squares relaxation. The hierarchy thus generated resembles the Lasserre hierarchy. We proved convergence of this hierarchy and obtained a convergence rate that improves on previous results [Lasserre, 2010]. Notably, we provide illustrations of how such a hierarchy performs in practice, focusing on the sensor network localization setting.

Though sum-of-squares polynomials map neatly onto the cone of semidefinite matrices, the SOS hierarchy may be replaced by even more restrictive cones, such as more tractable

DSOS and SDSOS optimization alternatives [Ahmadi and Majumdar, 2019]. There are fundamental algorithmic limits on how efficient semidefinite solvers can be, so any way we can avoid the semidefinite cone is welcome.

Even if we do keep the semidefinite cone, there are still more specialized ways to scale our method, particularly when we leverage sparsity or other structure in the solution matrix W. Our work outlined the cluster basis hierarchy, a variant that seeks to match the polynomial with basis functions of limited "body" order, i.e. the number of unique variables a given monomial has. Other ways of reducing the matrix size we need to consider should also be considered.

Finally, in Chapter 4, we took a look at generative diffusion processes. We looked at diffusion processes where the score function $s(x,t) = \nabla_x \log p_t(x)$ could be explicitly computed, and we found that the ill-conditioning of score near $t = 0$ and even multi-modality posed little hurdle for the current variants of diffusion models. It would be interesting to generalize these results to much higher dimensions, as behavior may be very different in those regimes. Ultimately, though we know for certain that diffusion models "just work" in practice, we seek to understand the "why" and "how".

BIBLIOGRAPHY

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Amir Ali Ahmadi and Anirudha Majumdar. DSOS and SDSOS Optimization: More Tractable Alternatives to Sum of Squares and Semidefinite Optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, January 2019. ISSN 2470-6566. doi:10.1137/18M118935X. URL `https://epubs.siam.org/doi/10.1137/18M118935X`.

Felicity Allen, Allison Pon, Michael Wilson, Russ Greiner, and David Wishart. CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. page 6, May 2014.

Felicity Allen, Russ Greiner, and David Wishart. Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification. *Metabolomics*, (11):98–110, 2015. doi:10.1007/s11306-014-0676-4.

Felicity Allen, Allison Pon, Russ Greiner, and David Wishart. Computational Prediction of Electron Ionization Mass Spectra to Assist in GC/MS Compound Identification. *Anal. Chem.*, page 9, July 2016.

Luigi Ambrosio, Nicola Gigli, and Savare. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser, second edition, 2005.

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, May 1982. ISSN 03044149. doi:10.1016/0304-4149(82)90051-5. URL `https://linkinghub.elsevier.com/retrieve/pii/0304414982900515`.

T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, Hoboken, NJ, 3rd edition, 2003.

MOSEK ApS. *The MOSEK optimization toolbox for Python manual. Version 10.0.*, 2023. URL https://docs.mosek.com/10.0.44/pythonapi/index.html.

Francis Bach and Alessandro Rudi. Exponential convergence of sum-of-squares hierarchies for trigonometric polynomials, April 2023. URL http://arxiv.org/abs/2211.04889. arXiv:2211.04889 [math].

Christoph Alexander Bauer and Stefan Grimme. How to Compute Electron Ionization Mass Spectra from First Principles. *J. Phys. Chem. A*, page 12, 2016.

Dimitris Bertsimas, Dan Andrei Iancu, and Pablo A. Parrilo. A Hierarchy of Near-Optimal Policies for Multistage Adaptive Optimization. *IEEE Transactions on Automatic Control*, 56(12):2809–2824, December 2011. ISSN 0018-9286. doi:10.1109/TAC.2011.2162878. URL http://ieeexplore.ieee.org/document/5986692/.

Nicolas Boumal, Vladislav Voroninski, and Afonso S Bandeira. The non-convex Burer–Monteiro approach works on smooth semidefinite programs.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 1 edition, March 2004. ISBN 978-0-521-83378-3 978-0-511-80444-1. doi:10.1017/CBO9780511804441. URL https://www.cambridge.org/core/product/identifier/9780511804441/type/book.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, New York, 1 edition, May 2011. ISBN 978-0-429-13850-8. doi:10.1201/b10905. URL https://www.taylorfrancis.com/books/9780429138508.

John Buckingham. *Dictionary of Natural Products, Supplement 2*. Routledge, Boca Raton, 1 edition, February 2023. ISBN 978-1-315-14116-9. doi:10.1201/9781315141169. URL https://www.taylorfrancis.com/books/9781315141169.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, February 2003. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-002-0352-8. URL http://link.springer.com/10.1007/s10107-002-0352-8.

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. A Survey on Generative Diffusion Model, December 2023. URL http://arxiv.org/abs/2209.02646. arXiv:2209.02646 [cs].

Yian Chen, Yuehaw Khoo, and Lek-Heng Lim. Convex Relaxation for Fokker-Planck, June 2023. URL http://arxiv.org/abs/2306.03292. arXiv:2306.03292 [cs, math].

Frank H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975. URL https://api.semanticscholar.org/CorpusID:120174258.

Etienne de Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, June 2008. ISSN 1613-9178. doi:10.1007/s10100-007-0052-9. URL https://doi.org/10.1007/s10100-007-0052-9.

Thomas De Vijlder, Dirk Valkenborg, Filip Lemière, Edwin P. Romijn, Kris Laukens, and Filip Cuyckens. A tutorial in small molecule identification via electrospray ionization-mass spectrometry: The practical art of structural elucidation. *Mass spectrometry reviews*, 37 (5):607–629, September 2018. ISSN 1098-2787 0277-7037. doi:10.1002/mas.21551.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Sander Dieleman. Diffusion is spectral autoregression, September 2024. URL https://sander.ai/2024/09/02/spectral-autoregression.html.

Yannick Djoumbou-Feunang, Allison Pon, Naama Karu, Jiamin Zheng, Carin Li, David Arndt, Maheswor Gautam, Felicity Allen, and David S Wishart. CFM-ID 3.0: Significantly Improved ESI-MS/MS Prediction and Compound Identification. page 23, 2019.

Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast, January 2018. URL https://arxiv.org/abs/1801.02309v4.

Ivar Ekeland and Roger Temam. Convex analysis and variational problems, 1976.

Kun Fang and Hamza Fawzi. The sum-of-squares hierarchy on the sphere, and applications in quantum information theory. *Mathematical Programming*, 190(1-2):331–360, November 2021. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-020-01537-7. URL http://arxiv.org/abs/1908.05155. arXiv:1908.05155 [quant-ph].

William Feller. On the Theory of Stochastic Processes, with Particular Reference to Applications. In René L. Schilling, Zoran Vondraček, and Wojbor A. Woyczyński, editors, *Selected Papers I*, pages 769–798. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16858-6 978-3-319-16859-3. doi:10.1007/978-3-319-16859-3_42. URL http://link.springer.com/10.1007/978-3-319-16859-3_42.

Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. A series of books in the mathematical sciences. Freeman, New York [u.a], 27. print edition, 2009. ISBN 978-0-7167-1044-8 978-0-7167-1045-5.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs]*, June 2017. URL http://arxiv.org/abs/1704.01212. arXiv: 1704.01212.

Clark R. Givens and Rae Michael Shortt. A class of Wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2), January 1984. ISSN 0026-2285.

doi:10.1307/mmj/1029003026. URL https://projecteuclid.org/journals/michigan-mathematical-journal/volume-31/issue-2/A-class-of-Wasserstein-metrics-for-probability-distributions/10.1307/mmj/1029003026.full.

Yanfei Guan, S. V. Shree Sowndarya, Liliana C. Gallegos, Peter C. St. John, and Robert S. Paton. Real-time prediction of 1H and 13C chemical shifts with DFT accuracy using a 3D graph neural network. *Chemical Science*, 12(36):12012–12026, 2021. ISSN 2041-6520, 2041-6539. doi:10.1039/D1SC03343C. URL http://xlink.rsc.org/?DOI=D1SC03343C.

Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat. Wavelet Score-Based Generative Modeling, August 2022. URL http://arxiv.org/abs/2208.05003. arXiv:2208.05003 [cs, stat].

M. B. Hastings. Field Theory and The Sum-of-Squares for Quantum Systems, February 2023. URL http://arxiv.org/abs/2302.14006. arXiv:2302.14006 [quant-ph].

Matthew B. Hastings. Perturbation Theory and the Sum of Squares, June 2022. URL http://arxiv.org/abs/2205.12325. arXiv:2205.12325 [cond-mat, physics:hep-th, physics:quant-ph].

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, June 2020. URL https://arxiv.org/abs/2006.11239v2.

Samuel B. Hopkins. *Statistical Inference and the Sum of Squares Method*. PhD Thesis, Cornell University, August 2018. URL https://www.samuelbhopkins.com/thesis.pdf.

Samuel B. Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034, Los Angeles CA USA, June 2018. ACM. ISBN 978-1-4503-5559-9. doi:10.1145/3188745.3188748. URL https://dl.acm.org/doi/10.1145/3188745.3188748.

Aapo Hyvarinen. Estimation of Non-Normalized Statistical Models by Score Matching. 2005.

D. Jackson. *The Theory of Approximation*. Colloquium Publications. American Mathematical Society, 1930. ISBN 9780821838921. URL https://books.google.de/books?id=e6GPCwAAQBAJ.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, Montreal Quebec Canada, June 2009. ACM. ISBN 978-1-60558-516-1. doi:10.1145/1553374.1553431. URL https://dl.acm.org/doi/10.1145/1553374.1553431.

Hongchao Ji, Hanzi Deng, Hongmei Lu, and Zhimin Zhang. Predicting a Molecular Fingerprint from an Electron Ionization Mass Spectrum with Deep Neural Networks. *Analytical Chemistry*, 92(13):8649–8653, July 2020. ISSN 0003-2700, 1520-6882. doi:10.1021/acs.analchem.0c01450. URL https://pubs.acs.org/doi/10.1021/acs.analchem.0c01450.

Hanyang Jiang and Yuehaw Khoo. Learning to solve semidefinite programs on graphs.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2020. URL https://doi.org/10.1093/nar/gkaa971. ISBN: 0305-1048 Type: 10.1093/nar/gkaa971.

Frederic Koehler, Alexander Heckett, and Andrej Risteski. Statistical Efficiency of Score Matching: The View from Isoperimetry, December 2022. URL http://arxiv.org/abs/2210.00726. arXiv:2210.00726 [cs, math, stat].

Jeroen Koopman and Stefan Grimme. Calculation of Electron Ionization Mass Spectra with Semiempirical GFNn-xTB Methods. *ACS Omega*, page 14, 2019.

Greg Landrum. RDKit: Open-Source Cheminformatics Software. URL https://www.rdkit.org/.

Jean Lasserre. The Moment-SOS hierarchy, August 2018. URL http://arxiv.org/abs/1808.03446. arXiv:1808.03446 [math].

Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001. ISSN 1052-6234. doi:10.1137/S1052623400366802. URL https://epubs.siam.org/doi/10.1137/S1052623400366802. Publisher: Society for Industrial and Applied Mathematics.

Jean B. Lasserre. A "Joint+Marginal" Approach to Parametric Polynomial Optimization. *SIAM Journal on Optimization*, 20(4):1995–2022, January 2010. ISSN 1052-6234. doi:10.1137/090759240. URL https://epubs.siam.org/doi/10.1137/090759240. Publisher: Society for Industrial and Applied Mathematics.

Jean-Bernard Lasserre. The Moment-SOS hierarchy: Applications and related topics. *To appear in Acta Numerica (2024)*, September 2023. URL https://laas.hal.science/hal-04201167.

Monique Laurent. Sums of Squares, Moment Matrices and Optimization Over Polynomials. In Mihai Putinar and Seth Sullivant, editors, *Emerging Applications of Algebraic Geometry*, The IMA Volumes in Mathematics and its Applications, pages 157–270. Springer, New York, NY, 2009. ISBN 978-0-387-09686-5. doi:10.1007/978-0-387-09686-5_7. URL https://doi.org/10.1007/978-0-387-09686-5_7.

Robert K. Lindsay, Bruce G. Buchanan, Edward A. Feigenbaum, and Joshua Lederberg.

Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project. McGraw-Hill Companies, 1980. ISBN 0-07-037895-9.

PC Mahalanobis. On the generalized distance in statistics. In *Proceedings National Institute of Science of India*, volume 49, pages 234–256, 1936. Issue: 2.

Lei Mao. Group Lasso, February 2020. URL `https://leimao.github.io/blog/Group-Lasso/`.

Dmitriy D. Matyushin, Anastasia Yu. Sholokhova, and Aleksey K. Buryak. Deep Learning Driven GC-MS Library Search and Its Application for Metabolomics. *Analytical Chemistry*, 92(17):11818–11825, September 2020. ISSN 0003-2700, 1520-6882. doi:10.1021/acs.analchem.0c02082. URL `https://pubs.acs.org/doi/10.1021/acs.analchem.0c02082`.

Fred W. McLafferty and Frantisek Turecek. *Interpretation of Mass Spectra*. 4 edition, 1994.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. Technical Report AECU-2435, LADC-1359, 4390578, March 1953. URL `http://www.osti.gov/servlets/purl/4390578/`.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3: e103, January 2017. ISSN 2376-5992. doi:10.7717/peerj-cs.103. URL `https://doi.org/10.7717/peerj-cs.103`.

Kevin P. Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press, Cambridge, MA, 2012. ISBN 978-0-262-01802-9.

Habib N. Najm. Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 41(1):35–52, 2009. doi:10.1146/annurev.fluid.010908.165248. URL https://doi.org/10.1146/annurev.fluid.010908.165248. _eprint: https://doi.org/10.1146/annurev.fluid.010908.165248.

Yurii Nesterov. Squared Functional Systems and Optimization Problems. In Hans Frenk, Kees Roos, Tamás Terlaky, and Shuzhong Zhang, editors, *High Performance Optimization*, Applied Optimization, pages 405–440. Springer US, Boston, MA, 2000. ISBN 978-1-4757-3216-0. doi:10.1007/978-1-4757-3216-0_17. URL https://doi.org/10.1007/978-1-4757-3216-0_17.

Andrew Ng and Michael Jordan. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://papers.nips.cc/paper_files/paper/2001/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html.

Jiawang Nie. Sum of squares method for sensor network localization. *Computational Optimization and Applications*, 43(2):151–179, June 2009. ISSN 1573-2894. doi:10.1007/s10589-007-9131-z. URL https://doi.org/10.1007/s10589-007-9131-z.

Jiawang Nie. Optimality conditions and finite convergence of Lasserre's hierarchy. *Mathematical Programming*, 146(1-2):97–121, August 2014. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-013-0680-x. URL http://link.springer.com/10.1007/s10107-013-0680-x.

Shen Nie, Hanzhong Allan Guo, Cheng Lu, Yuhao Zhou, Chenyu Zheng, and Chongxuan Li.

The Blessing of Randomness: SDE Beats ODE in General Diffusion-based Image Editing, February 2024. URL http://arxiv.org/abs/2311.01410. arXiv:2311.01410 [cs].

NIST. NIST Standard Reference Database 1A. Data version v17, software version 2.XX. URL https://www.nist.gov/srd/nist-standard-reference-database-1a.

Bernt Oksendal. *Stochastic differential equations (3rd ed.): an introduction with applications*. Springer-Verlag, Berlin, Heidelberg, 1992. ISBN 3387533354.

Ryan O'Donnell. SOS is not obviously automatizable, even approximately. *R. O*, 2016.

Dávid Papp and Sercan Yildiz. Sum-of-Squares Optimization without Semidefinite Programming. *SIAM Journal on Optimization*, 29(1):822–851, January 2019. ISSN 1052-6234, 1095-7189. doi:10.1137/17M1160124. URL https://epubs.siam.org/doi/10.1137/17M1160124.

P.A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, Pasadena, CA, 2000. URL https://thesis.library.caltech.edu/1647/1/Parrilo-Thesis.pdf.

Mihai Putinar. Positive Polynomials on Compact Semi-algebraic Sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993. ISSN 0022-2518. URL https://www.jstor.org/stable/24897130. Publisher: Indiana University Mathematics Department.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative Modelling With Inverse Heat Dissipation, April 2023. URL http://arxiv.org/abs/2206.13397. arXiv:2206.13397 [cs, stat].

Gareth O. Roberts and Richard L. Tweedie. Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):341, December 1996. ISSN 13507265. doi:10.2307/3318418. URL https://www.jstor.org/stable/3318418?origin=crossref.

Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, Princeton, NJ, 2015. ISBN 0691015864.

Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A Gentle Introduction to Graph Neural Networks. Technical report, Google Research, Distill, September 2021. URL 10.23915/distill.00033.

Konrad Schmüdgen. *The Moment Problem*, volume 277 of *Graduate Texts in Mathematics*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-64545-2 978-3-319-64546-9. doi:10.1007/978-3-319-64546-9. URL http://link.springer.com/10.1007/978-3-319-64546-9.

Saeid Sedighi, Kumar Vijay Mishra, M. R. Bhavani Shankar, and Bjorn Ottersten. Localization With One-Bit Passive Radars in Narrowband Internet-of-Things Using Multivariate Polynomial Optimization. *IEEE Transactions on Signal Processing*, 69: 2525–2540, January 2021. ISSN 1053-587X. doi:10.1109/TSP.2021.3072834. URL https://ui.adsabs.harvard.edu/abs/2021ITSP...69.2525S. ADS Bibcode: 2021ITSP...69.2525S.

Lucas Slot. Sum-of-squares hierarchies for polynomial optimization and the Christoffel-Darboux kernel, February 2023. URL http://arxiv.org/abs/2111.04610. arXiv:2111.04610 [math].

Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for Sensor Network Localization. *Mathematical Programming*, 109(2-3):367–384, January 2007. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-006-0040-1. URL http://link.springer.com/10.1007/s10107-006-0040-1.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep

Unsupervised Learning using Nonequilibrium Thermodynamics, November 2015. URL http://arxiv.org/abs/1503.03585. arXiv:1503.03585 [cond-mat, q-bio, stat].

Qun Song, Chaojie Gu, and Rui Tan. Deep Room Recognition Using Inaudible Echos. *arXiv:1809.00531 [cs, eess]*, September 2018. URL http://arxiv.org/abs/1809.00531. arXiv: 1809.00531.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution, October 2020. URL http://arxiv.org/abs/1907.05600. arXiv:1907.05600 [cs, stat].

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced Score Matching: A Scalable Approach to Density and Score Estimation, June 2019. URL http://arxiv.org/abs/1905.07088. arXiv:1905.07088 [cs, stat].

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. URL http://arxiv.org/abs/2011.13456. arXiv:2011.13456 [cs, stat].

Stephen E. Stein and Donald R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *Journal of the American Society for Mass Spectrometry*, 5(9):859–866, September 1994. ISSN 1044-0305. doi:10.1016/1044-0305(94)87009-8. URL https://pubs.acs.org/doi/10.1016/1044-0305%2894%2987009-8.

Bruno Sudret. Global sensitivity analysis using polynomial chaos expansion. *Reliability Engineering & System Safety*, 93:964–979, July 2008. doi:10.1016/j.ress.2007.04.002.

Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996. ISSN 00359246.

doi:10.1111/j.2517-6161.1996.tb02080.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1996.tb02080.x.

Lieven Vandenberghe. Chordal Graphs and Sparse Semidefinite Optimization, 2017.

Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, July 2011. ISSN 0899-7667, 1530-888X. doi:10.1162/NECO_a_00142. URL https://direct.mit.edu/neco/article/23/7/1661-1674/7677.

Lilapati Waikhom and Ripon Patgiri. Graph Neural Networks: Methods, Applications, and Opportunities. *arXiv:2108.10733 [cs]*, August 2021. URL http://arxiv.org/abs/2108.10733. arXiv: 2108.10733.

Shunyang Wang, Tobias Kind, Dean J. Tantillo, and Oliver Fiehn. Predicting in silico electron ionization mass spectra using quantum chemistry. *Journal of Cheminformatics*, 12(1):63, Oct 2020. ISSN 1758-2946. doi:10.1186/s13321-020-00470-3. URL https://doi.org/10.1186/s13321-020-00470-3.

Jennifer N Wei, David Belanger, Ryan P Adams, and D Sculley. Rapid Prediction of Electron–Ionization Mass Spectrometry Using Neural Networks. *ACS Central Science*, page 9, 2019.

Ming Yuan and Yi Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, February 2006. ISSN 1369-7412, 1467-9868. doi:10.1111/j.1467-9868.2005.00532.x. URL https://academic.oup.com/jrsssb/article/68/1/49/7110631.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph Neural Networks: A Review of Methods and Applications.

arXiv:1812.08434 [cs, stat], July 2019. URL http://arxiv.org/abs/1812.08434. arXiv: 1812.08434.

Hao Zhu, Liping Liu, and Soha Hassoun. Using Graph Neural Networks for Mass Spectrum Prediction. 2020.

Richard L. Zhu, Mathias Oster, and Yuehaw Khoo. S-SOS: Stochastic Sum-Of-Squares for Parametric Polynomial Optimization, June 2024. URL http://arxiv.org/abs/2406.08954. arXiv:2406.08954 [math].

Richard Licheng Zhu and Eric Jonas. Rapid Approximate Subset-Based Spectra Prediction for Electron Ionization–Mass Spectrometry. *Analytical Chemistry*, 95(5):2653–2663, February 2023. ISSN 0003-2700, 1520-6882. doi:10.1021/acs.analchem.2c02093. URL https://pubs.acs.org/doi/10.1021/acs.analchem.2c02093.