THE UNIVERSITY OF CHICAGO

SELF-PLAY METHODS IN REINFORCEMENT LEARNING FOR LANGUAGE MODELS

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

DEPARTMENT OF COMPUTER SCIENCE

BY

ZIYU YE

CHICAGO, ILLINOIS
DECEMBER 2025

To my family and Q

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this thesis we develop a series of practical algorithms for language model to self-train, by actively and strategically creating and controlling learning experiences themselves, enabling better model generalizability and training efficiency compared to traditional approaches.

In Chapter 1, we discuss the background and motivation of the work, providing necessary technical preliminaries on reinforcement learning and language model training, and lay out key contributions with a high-level principled view.

In Chapter 2, we introduce the Evolving Alignment (eva) framework, which replaces the classical static training scheme with an adaptive one, where a learnable creator policy generates and schedules training contexts for a solver policy. Plainly, new prompts are adaptively generated (by the language models themselves) and stored in a prioritized buffer, where their sampling frequency is determined by reward signals.

In Chapters 3 and 4, we discuss practical techniques to further improve the training of the solver policy. We first present the Reasoning in Reasoning (RiR) framework, which introduces a planner policy that generates subgoals for the solver policy to condition on. We then extend this idea by learning an abstract world model for lookahead planning. The proposed frameworks enhance both training and inference efficiency on theorem proving and real-world tool-use tasks.

Together, we use the phrase self play to summarize these studies, yet in essence they are nothing more than practical techniques for *synthesizing and scheduling learning experiences*. We anticipate future research in this direction to further move beyond the traditional static training paradigm toward adaptive continual self-training for language models, with stronger theoretical foundations and guarantees.

The complete package is to be updated on: link.

CHAPTER 1

INTRODUCTION

1.1 Background

Artificial Intelligence and Machine Learning. Intelligence measures an agent's ability to achieve goals in a wide range of environments [Legg et al., 2007]. Developing artificial intelligence can be formalized as a *learning* problem, where Mitchell [1997] has a succinct definition for learning:

> "*A computer program is said to learn from* **experience** *E with respect to some class of* **tasks** *T and* **performance measure** *P, if its performance at task T as measured by P improves with experience E.*"

Deep Learning. In deep learning [Goodfellow et al., 2016], the learning agent is a neural network, and improving the performance measure is to find parameters of the neural network that improves an objective function over a data-generating process of interest. While we may not have access to the true process, people often can optimize the objective over a curated dataset as the experience for training.

Reinforcement Learning. Reinforcement learning (RL) refers to the learning problems where the *experience E* comes from the agent's interaction with an *environment*. More specifically, a *policy* defines the agent's way of interaction and the environment sends a *reward signal* according to actions taken by the agent; the agent's objective is often to maximize the total reward it receives in the long run by updating its policy through trial-and-error learning [Sutton et al., 1998a].

Large Language Models. Large language models (LLMs) are deep generative models (often transformer-based [Vaswani et al., 2017]) which are trained on massive text data [Radford

et al., 2018], and can be viewed as agents experiencing in a textual environment. Recent works have explored applying RL to train large language models, where environment feedback can be human preferences [Christiano et al., 2017], program verification [Polu and Sutskever, 2020], etc. In this thesis, we focus especially on tasks solvable by language models, which is widely regarded as central to the current progress of artificial intelligence [Team, 2025].

Thesis Statement. Reinforcement learning has long emphasized *learning to act*, with little principled understanding of *learning what to experience*. This thesis introduces **self-play experience shaping** as an auxiliary objective jointly optimized with reward maximization, and presents practical self-play algorithms that bring concrete gains on standard benchmarks of language models.

1.2 Formal Statement

Setup. Let's use the contextual bandit objective [Abbasi-Yadkori et al., 2011] for a simplified illustration of RL, which is conveniently used in the context of training LLMs [Rafailov et al., 2023]:

$$J_{\mathsf{train}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}\sim q_{\mathsf{train}}(\cdot),\ \boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s})}\Big[R(\boldsymbol{s},\boldsymbol{a})\Big]. \tag{1.1}$$

where

- $\boldsymbol{s}$: the environment state/context, *i.e.*, the prompt for language models;
- $\boldsymbol{a}$: the action taken by the agent, *i.e.*, the response from language models;
- $R(\cdot,\cdot)$: the reward function, *e.g.*, the rating of the response from language models;
- $\pi_{\boldsymbol{\theta}}(\cdot|\cdot)$: the action policy and $\boldsymbol{\theta}$ is the parameter of the network being trained;

- $q_{\mathsf{train}}(\cdot)$: a prescribed training state distribution, determined by environment and externally specified sampling mechanisms of training, *e.g.*, a pre-curated prompt dataset [Cui et al., 2023].

The performance of the learning algorithm is evaluated on the target true data generating process:

$$J_{\mathsf{true}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}\sim q_{\mathsf{true}}(\cdot),\ \boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s})}\Big[R_{\mathsf{true}}(\boldsymbol{s},\boldsymbol{a})\Big]. \tag{1.2}$$

Problem. The issue of the default RL objective is that it assumes an exogenous experience distribution. To illustrate, while the learning agent drives action trajectories, the sampling of training experiences (*e.g.*, state prioritization and scheduling) is imposed exogenously of the agent's policy. The exogeneity of experience distribution can hurt the learning agent's **generalizability** (*e.g.*, due to distribution mismatch) and **training efficiency** (*e.g.*, due to high-variance update). Existing works try to solve this often use ad-hoc techniques to select and schedule the data. However, they are mostly heuristics-based, and lacks of a learnable, generative *data policy* as opposed to *action policy*, as we summarized in Table 1.1.

Curriculum Learning [Bengio et al., 2009]	Bias sampling towards easier examples first
Prioritized Replay Buffer [Schaul et al., 2015]	Bias sampling towards high TD-error transitions
Ordered SGD [Kawaguchi and Lu, 2020]	Re-order training examples by current loss
Curriculum RL [Jiang et al., 2021b]	Adapt environment difficulty over training

Table 1.1: Representative heuristics for experience sampling in training deep neural networks.

Research Question. We propose the following research question:

> *Can we design a principled approach for shaping the experience during RL (for large language models) to improve the generalizability and training efficiency?*

Our contribution. We propose a new family of RL training algorithms, where the learning agent is incentivized to adaptively generate and shaping its experiences while learning to act, enabling better learning generalizbility and training efficiency. To achieve this, we introduce a **learnable experience policy** π_ϕ jointly optimized with $\boldsymbol{\theta}$ to *sample*, *schedule*, and *generate* experience data.

$$\boldsymbol{\theta}^\star(\phi) = \arg\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\boldsymbol{s}\sim\pi_\phi(\cdot),\ \boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s})}\Big[R(\boldsymbol{s},\boldsymbol{a})\Big], \tag{1.3}$$

$$\text{where } \pi_\phi \text{ is an experience policy subject to its own objective.} \tag{1.4}$$

In the next, we present concrete objectives for the experience policy with practical algorithms.

CHAPTER 2

SCALABLE REINFORCEMENT LEARNING FROM HUMAN FEEDBACK VIA CREATOR-SOLVER GAMES

2.1 Introduction

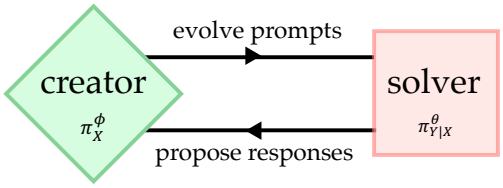

Figure 2.1: **RL for LLMs as Asymmetric Self-Play.** Conventional training methods are restricted to pre-curated prompt sets with fixed training schedules. We show that it is crucial to adaptively control the prompt distribution by creating and prioritizing more useful prompts. This is implemented through a simple self-play scheme (Algorithm 1), where the **creator** is the newly introduced prompt policy π_X and the **solver** is the response policy $\pi_{Y|X}$; both interact with objectives driven by reward signals (section 2.3).

Long-lived artificial intelligence must deal with an ever-evolving, open-ended world, however the current training paradigm is restricted to being fairly short-lived and static.

This paper concerns the post-training paradigm of large language models (LLMs), which is typically done in two stages, imitation (*i.e.*, supervised fine-tuning, SFT) and reinforcement learning (*i.e.*, RL post-training). In particular, we focus on the latter, which has led to remarkable success in enhancing LLM capabilities [Team et al., 2023, 2024a, 2025]. However, there is a fundamental issue in existing practices: they restrict themselves within a *pre-curated*, *static* prompt distribution during post-training.

This is sub-optimal and bottlenecks scaling properties *w.r.t.*: (i) **training efficiency**: the existing paradigm treats all prompts equally, despite their utility depending on the changing states of LLMs in training; as not all prompts contribute equally, relying on a static set with a fixed training schedule is inefficient. (ii) **model generalizability**: once the LLM saturates on the static prompt set, learning stops, preventing the model acquiring new skills

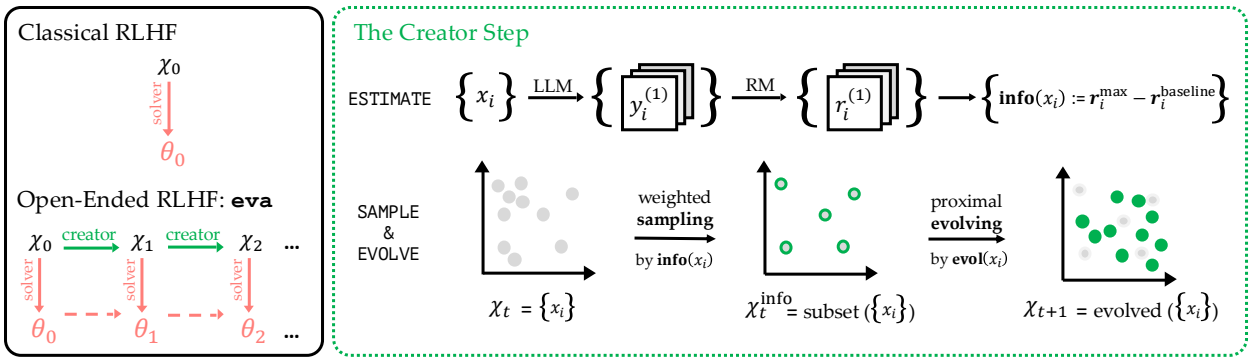

Figure 2.2: (Left) **eva** generalizes classical RL with open-ended RL for LLMs, by introducing a creator policy in training. (Right) The **creator** strategically controls prompt distributions by generating new prompts after each mini-batch gradient update (for online RL) or a complete dataset iteration (for offline RL), with a simple *estimate, sample then evolve* procedure, where the usefulness of each prompt is estimated from reward signals. The **solver** is then trained on a mixture of the original and evolved prompts. See the **minimax-regret** objective driving the game design and practical implementations in § 2.3.

or knowledge beyond the predefined distribution.[1]

We thereby investigate the two questions:

1. (Learning Signal) *Which prompts should be prioritized during RL training?*

2. (Algorithm) *How can we generate more useful prompts, and use them to keep LLMs self-improving in RL?*

To address them, we design **eva** (Evolving *via* Asymmetric Self-Play), as in Figure 2.1, 2.2. Central to **eva** is an infinite game with minimax-regret objectives, achieved by alternating optimization in creating prompts and solving them. We summarize our original contributions as:

1. We find **reward advantage** as an effective signal to identify useful prompts in RL post-training.

2. We design **eva**, the first method that enables LLMs to **prioritize** *and* **adaptively create** useful prompts, for continual RL training and self-improving.

1. Literature has considered prompt evolving, but is restricted to SFT [Xu et al., 2023a], and/or evolves uniformly [Yuan et al., 2024b]. We claim **adaptive training** *w.r.t.* **reward signals** are crucial.

The rest of technical details are organized as follows: in Problem 1, we formalize the problem of RL post-training as a bilevel optimization over a creator policy and a solver policy, which corresponds to the Stackelberg game [von Stackelberg, 1934] in game theory. In section 2.3.2, we define the objectives for the two players of the game by regret, for which the solver minimizes and the creator maximizes, and use its approximation by reward advantage for creator optimization. In section 2.3.3, we discuss simple practical algorithms of `eva` in both online and offline settings. In § 2.4, we run extensive experiments showing `eva` universally improves the performance of both online RL (*e.g.*, RLOO, OAIF) and offline RL (*e.g.*, DPO, SPPO, SimPO, ORPO) for LLMs and is SOTA on various challenging real-world benchmarks.

As we enter the new epoch where compute is moving from training to data synthesis, the question on "*how to put more compute in generating better data*" [Nishihara, 2025] is more critical than ever. While existing works [Zelikman et al., 2022b, Gulcehre et al., 2023, Chow et al., 2024] primarily focus on the **exploration in** $\mathcal{Y} \mid \mathcal{X}$, we perform the first systematic study in **exploration in** $(\mathcal{X}, \mathcal{Y})$ in RL for LLMs, guided by carefully designed *reward signals*.

2.2 Preliminaries

Classical RL post-training [Ouyang et al., 2022] optimizes over a **static prompt dataset** $\mathcal{D}$:

$$\max_{\pi_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[r(\mathbf{x},\mathbf{y})\Big] - \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\Big[\beta\cdot\mathbb{D}_{\mathsf{KL}}\Big(\pi_{\boldsymbol{\theta}}(\cdot\mid\mathbf{x})\,\|\,\pi_{\mathsf{base}}(\cdot\mid\mathbf{x})\Big)\Big]. \tag{2.1}$$

Here, $\pi_{\mathsf{base}}(\cdot|\mathbf{x})$ is a base policy, $\mathbf{x}$ and $\mathbf{y}$ are prompts and responses, $\mathbb{D}$ is a divergence measure. To approximate the expectation over $\mathbf{y}$, we employ Monte Carlo sampling by generating k responses per prompt, $\boldsymbol{y}_{(1)},\ldots,\boldsymbol{y}_{(k)} \sim \pi_{\boldsymbol{\theta}}(\cdot\mid\mathbf{x})$ [Sutton et al., 1998b, Ahmadian

et al., 2024]. In this paper, $r(\cdot)$ is the reward assumed to be from an oracle $r^\star(\cdot)$ and is fixed during post-training [Team et al., 2024a]; we use the conventional term RLHF (reinforcement learning from human feedback) interchangeably with RL (post-)training, and focus on human preference optimization, *i.e.*, AI alignment [Leike et al., 2018], in our experiments. Nevertheless, the pipeline is compatible with any other reward types.

Conventionally, prompts $\mathbf{x}$ are scheduled in training either sequentially or *via i.i.d.* uniform sampling. Prior works explored active selection [Kawaguchi and Lu, 2020, Muldrew et al., 2024] or prioritized sampling [Schaul et al., 2015, Lee et al., 2022], but focus solely on existing data. In contrast, `eva` introduces new prompt creation, enabling training with improved coverage and complexity beyond the initial static prompt set. Depending on how $\mathbf{y}$ is generated and labeled, RL post-training methods can be described as: (i) **online** on-policy, where $\mathbf{y}$ are generated on-policy [Team et al., 2024a]; and (ii) **offline**, where $\mathbf{y}$ are pre-generated by existing model checkpoints [Xiong et al., 2024, Pang et al., 2024a]. In this paper, for online RLHF, we evolve prompts at each mini-batch, while for offline RLHF, we construct a new prompt set after each full iteration over the previous set.

2.3 Method

2.3.1 The Problem: Open-Ended RLHF

Classical RLHF, as in section 2.2, samples prompts from a *static* set $\mathcal{D}$, which can have limited prompt coverage and complexity, and may diverge from true scenarios in the *open-ended world* [Dennis et al., 2020, Parker-Holder et al., 2022]. In Problem 1, we introduce a prompt generation policy $\pi_{\boldsymbol{\phi}}(\cdot)$ to be optimized together with the response generation policy $\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$. An optimizable $\pi_{\boldsymbol{\phi}}(\cdot)$ brings a few benefits: (i) during training, it allows dynamic adjustment of training prompts on-the-fly, making it possible to prioritize prompts that are more informative to the current $\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$, improving learning efficiency; (ii) at convergence, it

Problem 1 (Open-Ended RLHF)**.** *We define the problem of* Open-Ended RLHF *as the bilevel optimization on both a prompt policy (the creator $\pi_{\phi}(\mathbf{x})$) and a response policy (the solver $\pi_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$):*

$$\boldsymbol{\phi}^{\star} \in \arg\max_{\boldsymbol{\phi}} \; \mathcal{R}\Big(\pi_{\boldsymbol{\phi}}(\cdot), \pi_{\mathsf{true}}(\cdot); \mathcal{D}, \boldsymbol{\theta}^{\star}(\boldsymbol{\phi})\Big), \tag{2.2}$$

$$\text{s.t.} \quad \boldsymbol{\theta}^{\star}(\boldsymbol{\phi}) \in \arg\max_{\boldsymbol{\theta}} \; \mathbb{E}_{\mathbf{x}\sim\pi_{\boldsymbol{\phi}}(\cdot)}\Big[\, \mathbb{E}_{\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[\, r(\mathbf{x},\mathbf{y}) \,\Big] - \beta \cdot \mathbb{D}_{\mathsf{KL}}\Big[\pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x}) \,\|\, \pi_{\mathsf{base}}(\cdot \mid \mathbf{x})\Big]\Big]. \tag{2.3}$$

where π_{true} is the (potentially unknown) true target prompt distribution, $\mathcal{D}$ is an optional artifact parameter (e.g., the seed prompt distribution), and $R(\cdot)$ is a regularization function on the creator policy π_{ϕ}, which we discuss in detail at section 2.3. The problem generalizes classical RLHF, and captures the dual objective that (i) **response alignment***: the solver should perform well on the training prompt distribution while staying close to π_{base}, and (ii)* **prompt generation***: the creator should generate training prompts allowing the solver to perform robustly on target prompt distributions.*

brings a new prompt distribution beyond the initial static set, making it possible for $\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ to learn knowledge beyond $\mathcal{D}$ and perform more robustly on the target distribution. The creator objective $\mathcal{R}(\cdot)$ characterizes the optimization of $\pi_{\boldsymbol{\phi}}(\cdot)$, preventing it from collapsing in trivial prompts and guiding it towards true target prompts, which we discuss a specific implementation by regret maximization in the next.

2.3.2 The Game: Minimax Regret Games

Problem 1 can be cast as a sequential game [von Stackelberg, 1934] by two strategic players optimizing each's utility:

- **Solver** $\pi_{\boldsymbol{\theta}}(\mathbf{y} \mid \mathbf{x})$, who generates responses that optimize alignment given training prompts.
- **Creator** $\pi_{\phi}(\mathbf{x})$, who generates training prompts for the solver to perform well in the real world, knowing the solver will optimize over the generated.

A natural objective for the creator is to improve the solver's transfer performance [Ben-

gio, 2012] on the true target prompt distribution π_{true}: the closer π_ϕ is to π_{true}, the better the solver performance is expected, thereby the higher the utility the creator may receive. If π_{true} is known, $\mathcal{R}(\cdot)$ can be instantiated by a f-divergence measure for distribution matching with π_{true}. This work considers the case when π_{true} is unknown a priori; the optimization then falls into a standard *decision under ignorance problem* [Savage, 1951, Gustafsson, 2022]. Several decision rules can be considered, *e.g.*, randomization, that chooses training prompt distribution uniformly [Jiang, 2023]; in this paper, we study *Minimax Regret Rule*, which finds a training distribution that minimizes solver's worst-case regret over all possible distributions (see also section 2.4.2). We abuse the notation r to denote the reward with KL penalty, and **Regret** is defined as reward differences of $\pi_{\boldsymbol{\theta}}$ and optimal policy $\pi^\star_{\boldsymbol{\theta}}$:

$$\begin{aligned}\mathrm{Regret}(\pi_{\boldsymbol{\phi}}, \pi_{\boldsymbol{\theta}}) = \mathbb{E}_{\mathbf{x}\sim\pi_{\boldsymbol{\phi}}(\cdot)}\Big[&\mathbb{E}_{\mathbf{y}\sim\pi^\star_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[\ r(\mathbf{x},\mathbf{y})\ \Big] \\ &-\mathbb{E}_{\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[\ r(\mathbf{x},\mathbf{y})\ \Big]\Big].\end{aligned}$$

Problem 1 is then formulated as:

$$\boldsymbol{\phi}^\star \in \arg\max_{\boldsymbol{\phi}}\ \mathrm{Regret}(\pi_\phi, \pi_{\boldsymbol{\theta}^\star}), \tag{2.4}$$

$$s.t.\quad \boldsymbol{\theta}^\star(\boldsymbol{\phi}) \in \arg\min_{\boldsymbol{\theta}}\ \mathrm{Regret}(\pi_\phi, \pi_{\boldsymbol{\theta}}). \tag{2.5}$$

where

$$\mathrm{Regret}(\pi_{\boldsymbol{\phi}}, \pi_{\boldsymbol{\theta}}) = \ \mathbb{E}_{\mathbf{x}\sim\pi_{\boldsymbol{\phi}}(\cdot)}\Big[\mathbb{E}_{\mathbf{y}\sim\pi^\star_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[\ r(\mathbf{x},\mathbf{y})\ \Big]\mathbb{E}_{\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\Big[\ r(\mathbf{x},\mathbf{y})\ \Big]\Big].$$

Note that (i) for solver optimization, Eq. 2.5 is equivalent to Eq. 2.3 by definition, and (ii) for creator optimization, Eq. 2.4 approximates Eq. 2.2 with a worst-case optimal guarantee for the solver's policy, when π_{true} is unknown.

Remark 1. *Under mild assumptions, the (local)* Nash equilibrium *is a (local)* minimax point *[Jin et al., 2020] for the above optimization; here, at the (local) Nash equilibrium,*

the solver follows a (local) minimax regret policy [Jiang, 2023], i.e., the solver's regret is worst-case optimal.

The equilibrium finding of the game can be solved by alternating optimization [Zhang et al., 2022]. Intuitively, this allows for the creation of evolving prompt distributions that challenge the agent progressively for better generalization; the regret objective ensures *robustness* on such evolving curricula by *incentivizing agents to perform well in all cases*, providing a worst-case guarantee. In optimization, this brings a sweet spot where the creator can create challenging yet solvable prompts (*i.e.*, neither too hard nor too easy) for the solver.

Regret Minimization for the Solver. Any preference optimization algorithms can be used as a plug-in for the regret minimization for the solver's step in Algorithm 1.

Regret Maximization for the Creator. When it is direct for the solver to minimize the regret by policy optimization, the true optimal policy remains unknown during optimization, and we must approximate it when using it as the utility to incentivize the creator. Similar to heuristics in prior works [Jiang et al., 2021b,a, Parker-Holder et al., 2022], we use the advantage-based estimate for each $\mathbf{x}$:

$$\hat{\text{Regret}}(\mathbf{x}, \pi_{\boldsymbol{\theta}}) \leftarrow r(\mathbf{x}, \mathbf{y}_+) - r(\mathbf{x}, \mathbf{y}_{\mathsf{baseline}}), \tag{2.6}$$

where

$$\mathbf{y}_+ := \arg\max_{\mathbf{y}_i} r(\mathbf{x}, \mathbf{y}),$$
$$\mathbf{y}_{\mathsf{baseline}} := \text{avg}_{\mathbf{y}_i}\ r(\mathbf{x}, \mathbf{y}) \text{ or } \arg\min_{\mathbf{y}_i} r(\mathbf{x}, \mathbf{y}),$$

and $\{\mathbf{y}_i\}_{i=1}$ is a set of responses sampled from $\pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x})$ and $r(\cdot, \cdot)$ is the reward oracle. We choose $\arg\text{avg}_{\mathbf{y}_i} r(\mathbf{x}, \mathbf{y})$ for online RLHF, and $\arg\min_{\mathbf{y}_i} r(\mathbf{x}, \mathbf{y})$ for offline RLHF, based

on consistent strong empirical gains observed across extensive experiments. As the policy optimizes, the proxy will approximate the true regret better[2].

We denote the regret estimate (*i.e.*, negated reward advantage) as the informativeness value for a prompt $\mathbf{x}$ *w.r.t.* $\boldsymbol{\theta}$,

$$\mathsf{info}_{\boldsymbol{\theta}}(\mathbf{x}) := \hat{\mathrm{Regret}}(\mathbf{x}, \pi_{\boldsymbol{\theta}}). \tag{2.7}$$

Directly doing gradient ascent on regret could lead to training instability [Zhang, 2023]. In this work, we *approximate new prompt distributions* that maximize regret by 3 steps:

1. **Estimate informativeness** for each prompt in the set.
2. **Sampling a subset of high-regret prompts**.
3. **Generating new prompts** by making variations on those high-regret prompts.

The `eva` way of scalable regret maximization can relate to curriculum RL [Parker-Holder et al., 2022], which finds environments with high-regret levels, then edits within some distance, or *evolution strategies* [Schwefel, 1977] which find the most promising species, then mutate and crossover.

2.3.3 The Practical Algorithm

Algo 1 is an overview of `eva` where creator creates new training prompts after each iteration of solver.

The Solver Step. This step is the classical preference optimization [Rafailov et al., 2023]. Take DPO as an example, for every prompt, we sample n responses and annotate rewards,

2. This approximates the expectation over $\pi^{\star}_{\boldsymbol{\theta}}$ by best observed responses, which introduces bias quantifiable by $\mathbb{D}_{\mathsf{KL}}[\pi_{\boldsymbol{\theta}} \| \pi^{\star}]$. Also, single-sample estimation for $\pi_{\boldsymbol{\phi}}$ introduces variance. See Liu et al. [2023c] for discussions on sampling effects.

then take the responses with the maximal and the minimal reward to construct preference pairs and optimize upon.

The Creator Step Plainly, the creator finds most useful prompts and generate variants of them to approximate regret maximization.

- **Step 1: `info`**$(\cdot)$ – *estimate the informativeness.* For each $\mathbf{x}$ in the prompt set $\mathcal{X}_t$, we generate responses, annotate rewards and estimate the informativeness of $\mathbf{x}$ by Eq. 2.7.
- **Step 2: `sample`**$(\cdot)$ – *weighted sampling for an informative subset.* By using the informativeness metric as the weight, we sample an informative subset $\mathcal{X}_t^{\mathsf{info}}$ to be evolved.
- **Step 3: `evolve`**$(\cdot)$ – *evolving for high-regret prompts.* `eva` is agnostic to and does not rely on any specific evolving method. We take Xu et al. [2023a] as a default baseline for prompt evolving.

Prioritized Generative Buffer. While `eva` can operate on the full $\mathcal{D}$ at once and iteratively train LLMs (*i.e.*, offline `eva`), the informativeness can become off-policy. Inspired by Schaul et al. [2015], we design a simple *prioritized generative buffer* $\mathcal{B}$ that extends Algorithm 1 to be on-policy and evolves per mini-batch (*i.e.*, online `eva`), with:

1. **Warm-up phase**: we start with $\mathbf{x}$ from $\mathcal{D}$ and populate $\mathcal{B}$ with evolved prompts until it reaches size B.
2. **Mix-up phase**: training continues using a balanced mix of samples from $\mathcal{D}$ and $\mathcal{B}$ (prioritized by informativeness) per mini-batch. New prompts are evolved and added to $\mathcal{B}$, while older ones are removed.
3. **Bootstrap phase**: Once $\mathcal{D}$ is exhausted, training relies on $\mathcal{B}$, continuing evolving and replacing prompts.

`eva` is easy to use and flexible to extend. We present our adaptive reinforcement training algorithm below.

Algorithm 1 A Practical Implementation of `eva`

Input: a prompt set $\mathcal{X}_0$, solver policy $\pi_{\boldsymbol{\theta}}$, no. of rollout per prompt N, chosen RLHF algorithm Φ, reward function $r(\cdot)$

for iteration $t = 0, 1, \ldots$ **do**

▽ /* creator step */

  for $\mathbf{x} \in \mathcal{X}_t$ **do**

    $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \overset{\text{i.i.d.}}{\sim} \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{x})$

    $\mathsf{weight}(\mathbf{x}) \leftarrow \max_i r(\mathbf{x}, \boldsymbol{y}_i) - \frac{1}{N} \sum_{i=1}^{N} r(\mathbf{x}, \boldsymbol{y}_i)$

  end for

  $\mathcal{X}_t^{\text{seed}} \leftarrow$ sample M_1 prompts from $\mathcal{X}_t$ w.p. $\propto \mathsf{weight}(\mathbf{x})$

  $\mathcal{X}_t^{\text{unif}} \leftarrow$ sample M_2 prompts from $\mathcal{X}_t$ uniformly

  $\mathcal{X}_{t+1} \leftarrow \mathsf{evolve}(\mathcal{X}_t^{\text{seed}}) \cup \mathcal{X}_t^{\text{unif}}$

▽ /* solver step */

  optimize π_θ *using algorithm* Φ *and prompt set* $\mathcal{X}_{t+1}$

end for

2.4 Experiments

Datasets and models. We use **UltraFeedback** [Cui et al., 2023] as the training dataset, which contains diverse high-quality prompts that are primarily human-generated. We use the instruction-finetuned GEMMA-2-9B [Team et al., 2024b] as the base $(\boldsymbol{\theta}_0)$[3], which is a

3. Unless stated otherwise, each iteration uses 10K prompts (the initial prompt set), referred to as 1×. In offline RLHF, we denote $\boldsymbol{\theta}_{t \to t+1}$ as the one trained with new human prompts from the t-th checkpoint. $\boldsymbol{\theta}_{t \to \tilde{t}}$ denotes the one trained with evolved prompts from the t-th checkpoint without any new human prompts. In online RLHF, training is a continual iteration and $\boldsymbol{\theta}_{0 \to \tilde{1}}$ (n×) denotes training with 10n K prompts in total, mixed and evolved from the initial.

strong baseline for models of its size. Note that we directly apply RL training without SFT, as the base model is sufficiently capable.

Evaluation settings. We use: (i) **AlpacaEval 2.0** [Dubois et al., 2024a], which assesses general instruction following with 805 questions; (ii) **MT-Bench** [Zheng et al., 2023e], which evaluates multi-turn instruction following with 80 hard questions in 8 categories; (iii) **Arena-Hard** [Li et al., 2024b], which is derived from 200K user queries on Chatbot Arena with 500 challenging prompts across 250 topics.

Optimization algorithms. We evaluate `eva` across a wide range of six representative RLHF algorithms:

- **Online RLHF**: RLOO [Ahmadian et al., 2024], OAIF (*i.e.*, online DPO) [Guo et al., 2024].
- **Offline RLHF**: (*with reference*) DPO [Rafailov et al., 2023], SPPO [Wu et al., 2024b]; (*without reference*) SimPO [Meng et al., 2024], ORPO [Hong et al., 2024].

Oracle reward models. We take ArmoRM-8B [Wang et al., 2024b] to be the default reward model for human-preference proxy, with the below for ablation studies:

- **Pointwise**: ArmoRM-8B [Wang et al., 2024b], SkyworkRM-27B [Liu and Zeng, 2024].
- **Pairwise**: PairRM-0.4B [Jiang et al., 2023], PairRM-8B [Dong et al., 2024a].

2.4.1 Main Results

`eva` consistenly achieves strong self-improvement. As in in Table 2.1 and 2.2, `eva` yields notable performance improvement across different optimization algorithms, especially on the more challenging and robust Arena-Hard benchmark [Li et al., 2024b]. For example, `eva` brings 10.6% gain with DPO in the offline setting, and 9.8% gain with RLOO in the on-

line setting, surpassing **`claude-3-opus-240229`** as reported by AH leaderboard and matching **`gemini-1.5-pro`**, while using fully adaptive self-automated joint prompt-response generation. This demonstrates the superior empirical performance of **`eva`**.

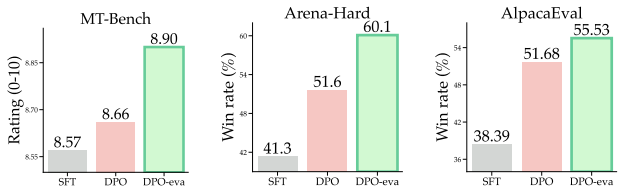

Figure 2.3: Illustration of gains with one round **`eva`** by DPO.

Table 2.1: **Online `eva` results.** **`eva`** has notable gains and is comparable to default training with even 6x human prompts (gray). Note **`eva` only uses 1x human prompts** and continuously evolves (nx denotes total prompt size).

Optimization Method (→)	Online RLHF				
Benchmark (→)	**Arena-Hard**	**MT-Bench**			**AE 2.0**
Method (↓) / Metric (→)	**WR** (%)	**avg.**	turn 1	turn 2	**LC-WR** (%)
$\boldsymbol{\theta}_0$: Base Model	41.3	8.57	8.81	8.32	47.11
$\boldsymbol{\theta}_{0\to1}$: RLOO (1x)	52.6	8.68	9.02	8.34	54.23
$\boldsymbol{\theta}_{0\to\tilde{1}}$: RLOO-`eva` (1x)	**57.3**	8.87	9.03	8.71	**55.02**
$\boldsymbol{\theta}_{0\to\tilde{1}}$: RLOO-`eva` (2x)	**60.5**	8.96	9.12	8.80	**57.10**
$\boldsymbol{\theta}_{0\to\tilde{1}}$: RLOO-`eva` (3x)	**62.4**	9.09	9.23	8.94	**61.04**
$\theta_{0\to1}$: RLOO (6x)	62.7	9.07	9.24	8.90	62.91
$\boldsymbol{\theta}_{0\to1}$: OAIF (1x)	52.1	8.66	8.97	8.35	55.15
$\boldsymbol{\theta}_{0\to\tilde{1}}$: OAIF-`eva` (1x)	**55.0**	8.85	9.04	8.66	**55.43**
$\boldsymbol{\theta}_{0\to\tilde{1}}$: OAIF-`eva` (2x)	**60.4**	8.93	9.06	8.79	**56.49**
$\boldsymbol{\theta}_{0\to\tilde{1}}$: OAIF-`eva` (3x)	**61.7**	9.01	9.19	8.82	**59.09**

Table 2.2: **Offline `eva` results.** We apply `eva` after 1 iteration of offline RLHF. It brings strong gains and can surpass training with human prompts. See more iterations in § 2.4.2.

Optimization Method (→)	Offline RLHF				
Benchmark (→)	**Arena-Hard**	**MT-Bench**			**AE 2.0**
Method (↓) / Metric (→)	**WR** (%)	**avg.**	turn 1	turn 2	**LC-WR** (%)
$\boldsymbol{\theta}_0$: Base Model	41.3	8.57	8.81	8.32	47.11
$\boldsymbol{\theta}_{0\to1}$: DPO	51.6	8.66	9.01	8.32	55.01
$\boldsymbol{\theta}_{1\to\tilde{1}}$: + `eva`	**60.1**	**8.90**	**9.04**	**8.75**	55.35
$\theta_{1\to2}$: + new human prompts	59.8	8.64	8.88	8.39	**55.74**
$\boldsymbol{\theta}_{0\to1}$: SPPO	55.7	8.62	9.03	8.21	51.58
$\boldsymbol{\theta}_{1\to\tilde{1}}$: + `eva`	**58.9**	**8.78**	**9.11**	**8.45**	**51.86**
$\theta_{1\to2}$: + new human prompts	57.7	8.64	8.90	8.39	51.78
$\boldsymbol{\theta}_{0\to1}$: SimPO	52.3	8.69	9.03	8.35	54.29
$\boldsymbol{\theta}_{1\to\tilde{1}}$: + `eva`	**60.7**	**8.92**	**9.08**	**8.77**	**55.85**
$\theta_{1\to2}$: + new human prompts	54.6	8.76	9.00	8.52	54.40
$\boldsymbol{\theta}_{0\to1}$: ORPO	54.8	8.67	9.04	8.30	52.17
$\boldsymbol{\theta}_{1\to\tilde{1}}$: + `eva`	**60.3**	**8.89**	**9.07**	**8.71**	**54.39**
$\theta_{1\to2}$: + new human prompts	57.2	8.74	9.01	8.47	54.00

`eva` curricula can surpass human-crafted prompts. We further show that `eva` models can match and even outperform those trained on additional new prompts from UltraFeedback (denoted as new human prompts as they are primarily sourced from humans [Cui et al., 2023]), while being much more efficient. Interestingly, on MT-Bench, training with new human prompts typically show decreased performance in the 1^{st} turn and only modest gains in the 2^{nd} turn, whereas `eva` notably enhances 2^{nd} gains. We hypothesize that `eva` adaptively evolves novel, learnable prompts that include features of second-turn questions, reflecting generalized skills like handling follow-up interactions.

2.4.2 Ablation Studies

Taking offline DPO as a representative case, we conduct extensive ablation studies on **`eva`**:

- § 2.4.2 - **informativeness metric**: our *regret*-based metric outperforms other alternatives.
- § 2.4.2 - **adaptive evolving procedure**: our method outperforms active selection without evolving.
- § 2.4.2 - **scaling with reward models**: the alignment gain of `eva` scales with reward models.
- § 2.4.2 - **continual training** : our method has monotonic gain with incremental training.
- § 2.4.2 - **curriculum effect**: our method creates meaningful curriculum over iterations.

The Choice of Informativeness Metrics

Metric	**info**$(\mathbf{x})$	**Related Approximation**
$A^\star_{\text{min}}$: worst-case optimal advantage	$\lvert \min_{\mathbf{y}} r(\mathbf{x},\mathbf{y}) - \max_{\mathbf{y}'} r(\mathbf{x},\mathbf{y}') \rvert$	minimax regret [Savage, 1951]
$A^\star_{\text{avg}}$: average optimal advantage	$\lvert \frac{1}{N}\sum_{\mathbf{y}} r(\mathbf{x},\mathbf{y}) - \max_{\mathbf{y}'} r(\mathbf{x},\mathbf{y}') \rvert$	Bayesian regret [Banos, 1968]
$A^\star_{\text{dts}}$: dueling optimal advantage	$\lvert \max_{\mathbf{y}\neq\mathbf{y}^\star} r(\mathbf{x},\mathbf{y}) - \max_{\mathbf{y}'} r(\mathbf{x},\mathbf{y}') \rvert$	min-margin regret [Wu and Liu, 2016]

Table 2.3: The reward-advantage-based metrics that serve as the informativeness proxies for prompts.

Benchmark (→)	Arena-Hard	MT-Bench			AE 2.0
Method (↓) / Metric (→)	WR (%)	avg.	turn 1	turn 2	LC-WR (%)
$\boldsymbol{\theta}_{0\to 1}$: DPO	51.6	8.66	9.01	8.32	55.01
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** (uniform)	57.5	8.71	9.02	8.40	53.43
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** (var($\boldsymbol{r}$))	54.8	8.66	9.13	8.20	54.58
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** (avg($\boldsymbol{r}$))	58.5	8.76	9.13	8.40	55.01
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** ($1/$avg($\boldsymbol{r}$))	56.7	8.79	9.13	8.45	55.04
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** ($1/A^{\star}_{\text{min}}$)	52.3	8.64	8.96	8.31	53.84
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** ($A^{\star}_{\text{avg}}$) (our variant)	60.0	8.85	9.08	8.61	**56.01**
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** ($A^{\star}_{\text{dts}}$) (our variant)	60.0	8.86	**9.18**	8.52	55.96
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + **eva** ($A^{\star}_{\text{min}}$) (our default)	**60.1** (+8.5)	**8.90**	9.04	**8.75** (+0.43)	55.35

Table 2.4: **Choice of informativeness metric matters.** Our adaptive metric by *reward advantage* achieves the best performances.

Reward advantage as the informativeness metric outperforms baselines. As in Table 2.4, **eva** offers an effective curriculum by the advantage-based proxy as the informativeness metric (bottom row):

- *Comparing with uniform evolving* (brown): Existing baselines generate prompts in a uniform manner [Yuan et al., 2024b] (*cf.*, the principle of insufficient reason [Keynes, 1921, Tobin et al., 2017]). **eva** concretely outperforms, corroborating Das et al. [2024] that uniform learners can suffer from sub-optimality gaps.

- *Comparing with other heuristics* (gray): Prior practices [Team et al., 2023] tried heuristics like prioritizing prompts with the most variance in its rewards or with the lowest/highest average. We find our advantage based methods (red) outperforms those heuristics.

- *Comparing with the inverse advantage* (purple): Contrary to curriculum learning, a line of works conjecture that examples with higher losses may be prioritized [Jiang et al., 2019], which can be done by inverting our metric. We find it significantly *hurt* the alignment gain, corroborating Mindermann et al. [2022] that those examples can

be unlearnable or irrelevant, meaning our curriculum is effective and practical.

- *Among our advantage variants* (green): We designed variants of our default advantage-based metric, as in Table 2.3; the default $A^{\star}_{\min}$ remains competitive among its peers. Together, the advantage-based principle provides a robust guideline for sampling and evolving.

The lesson is that we must be selective about which are the promising to evolve, otherwise unlearnable, noisy or naïve prompts may hinder learning. Our regret-inspired metric represents a solid baseline.

The Effect of Evolving

Benchmark (→)	**Arena-Hard**	**MT-Bench**			**AlpacaEval 2.0**	
Method (↓) / **Metric** (→)	**WR** (%)	**avg.**	turn 1	turn 2	**LC-WR** (%)	**WR** (%)
$\boldsymbol{\theta}_{0\to 1}$: DPO	51.6	8.66	9.01	8.32	55.01	51.68
$\boldsymbol{\theta}_{1\to \tilde{1}}$: `[no evolve]-greedy`	56.1	8.68	8.98	8.38	54.11	53.66
$\boldsymbol{\theta}_{1\to \tilde{1}}$: `[no evolve]-sample`	55.3	8.69	9.00	8.38	54.22	54.16
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + `eva-greedy` (our variant)	59.5	8.72	9.06	8.36	54.52	55.22
$\boldsymbol{\theta}_{1\to \tilde{1}}$: + `eva-sample` (our default)	**60.1**	**8.90**	9.04	**8.75**	**55.35**	**55.53**

Table 2.5: **Effect of evolving.** The blue are those training with only the informative subset and without evolving); we denote `-sample` for the default weighted sampling procedure in Algo 1, while using `-greedy` for the variant from the classical active data selection procedure (*cf.*, a recent work [Muldrew et al., 2024] and a pre-LLM work [Kawaguchi and Lu, 2020]), which selects data by a high-to-low ranking via the metric greedily. We show evolving brings a remarkable alignment gain (green v.s. blue); and as we evolve, sampling is more robust than being greedy.

The design of `evolve(·)` is effective. As in Table 2.5:

- Removing the `evolve`$(\cdot)$ step: if we only do subset sampling or ordered selection, we still have gain, but not as much as with evolving (*e.g.*, `eva` brings 4.8% additional wins on Arena Hard).

- Altering the `sample`$(\cdot)$ step: if we greedily select prompts by the metric instead of using them as weights for importance sampling, the performance will be weaker as we evolve.

The lesson is that simply adaptive training within a fixed prompt distribution is not enough; our open-ended RLHF with *generative* prompt exploration gives a substantial headroom for self-improvement. In other words, the RL post-training process should be both *adaptive* and *generative* in terms of prompt distribution.

Scaling `eva` with Reward Models

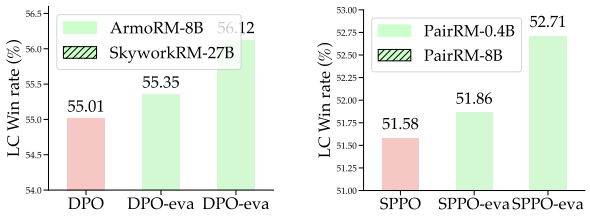

Figure 2.4: **`eva` scales with quality of reward models**, under pointwise RMs with DPO (*left*) and pairwise RMs with SPPO (*right*). Note SPPO handles general preferences thus requires pairwise RMs, and DPO relies on the Bradley-Terry assumption, for which pointwise RMs are suitable.

Figure 2.4 presents the length-controlled win rate of `eva` on AlpacaEval using pointwise and pairwise reward models of varying scales. As the quality of reward models improve, `eva` brings higher alignment gain. The scaling observation shows the effectiveness of `eva` in exploiting more accurate reward signals to choose informative prompts for better alignment. One takeaway is interaction with the external world is essential for intelligence. The more accurate reward signals observed, the better the agent incentivize themself to improve (*cf.*, Silver et al. [2021a]).

`eva` Improves Efficiency & Generalization

We run the default *incremental training* (*i.e.*, trainining from the last checkpoint with the evolved set in each iteration), as in Fig 2.5, `eva` presents *monotonic gains*.

The solutions found by `eva` *cannot* be recovered by training longer by a fixed set (the dashed), nor by naïvely sourcing new prompts without examining informativeness (the gray

dotted), thus our generative data schedule is effective.

We conjecture that behaviors of the dashed/dotted lines relate to *loss of plasticity* [Ash and Adams, 2019, Dohare et al., 2023, Abbas et al., 2023b, Xue et al., 2024]. Classical works resolve it by the *optimization* view (*e.g.*, weight perturbing), whereas **eva** offers a new *data* view, potentially mimicing an **implicit regularizer for better generalization**.

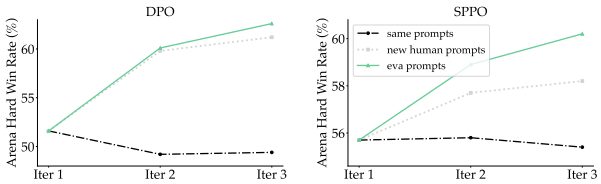

Figure 2.5: **eva** stays robust with more iterations.

In Table 2.6, we ablate **eva** in *scratch training*, *i.e.*, training with the full original *and* evolved set. **eva** is competitive in incremental training, *learning more effectively* with *less data* – a nice bonus by minimax regret [Jiang et al., 2021a].

Table 2.6: **Ablation on incremental v.s. scratch training.**

Benchmark (→)	**Arena-Hard**	**MT-Bench**	**AE 2.0**
Method (↓) / **Metric** (→)	**WR** (%)	**avg. score**	**LC-WR** (%)
$\boldsymbol{\theta}_0$: SFT	41.3	8.57	47.11
$\boldsymbol{\theta}_{0\to 1}$: DPO	51.6	8.66	55.01
$\boldsymbol{\theta}_{0\to \tilde{1}}$: **eva** (scratch)	59.8	8.88	54.59
$\boldsymbol{\theta}_{1\to \tilde{1}}$: **eva** (incremental)	**60.1**	**8.90**	**55.35**

`eva` Creates Meaningful Curriculum

Table 2.7: `eva` improves prompt quality and complexity.

Prompt Set (↓) / Metric (→)	**Complexity** (1-5)	**Quality** (1-5)
UltraFeedback (seed)	2.90	3.18
UltraFeedback-`eva`-Iter-1	3.84	3.59
UltraFeedback-`eva`-Iter-2	3.92	3.63
UltraFeedback-`eva`-Iter-3	**3.98**	**3.73**

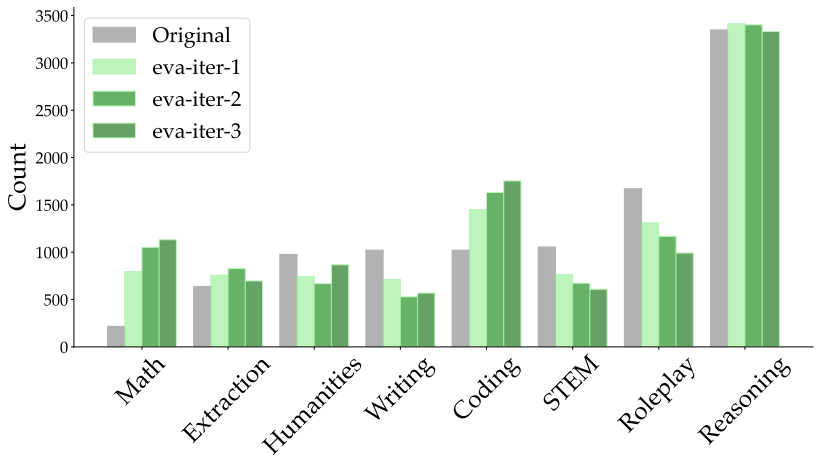

Figure 2.6: **Curriculum effect in training distributions.** `eva` creates a curriculum that prioritizes math / coding prompts over iterations.

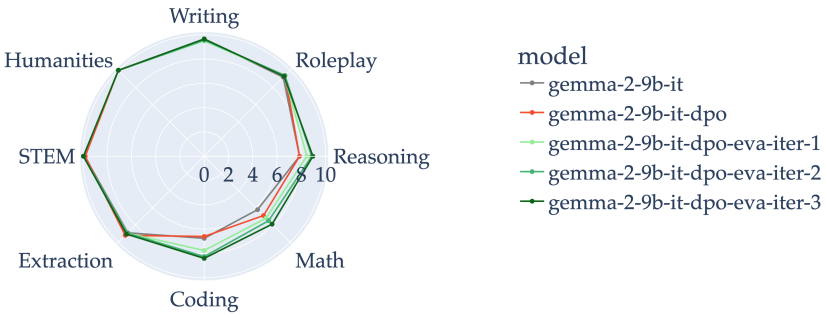

Figure 2.7: **Curriculum effect in benchmark performance.** The radar figure for ratings on MT-Bench. `eva` prioritizes and gradually improves on coding, math and reasoning over iterations, implicitly reflecting a learned curriculum.

In Table 2.7, we show that there is a gradual improvement in prompt complexity and quality[4] over iterations with `eva`. In Figure 2.6 and 2.7, we show that `eva` brings auto-curricula and the creator is incentivized to create new prompts that are informative *w.r.t.* the current solver policy.

Together, those evidences supports the importance of *adaptively evolving* prompts jointly with responses, which we believe to be crucial in scaling up next-gen RL post-training.

2.5 Related Works

With the history of machine learning, it is not new that self-play and data exploration brings intelligence (*e.g.*, Schmidhuber [1991]). We believe, to our knowledge, `eva` is among the first empirical works that systematically studied adaptive prompt evolving in RL post-training on large-scale LLM benchmarks. Below, viewing from many different angles, we list `eva`'s distinct impact and contribution.

Self-improving algorithms and iterative optimization. This line of work focuses on iteratively generating samples from the response policy and continuously re-training the policy by selected self-generated samples. Major works include ReST [Gulcehre et al., 2023, Singh et al., 2023b], STaR [Zelikman et al., 2022b], RFT [Yuan et al., 2023], RAFT [Dong et al., 2023], self-improving LLMs [Huang et al., 2022a, Yuan et al., 2024b]; in the context of preference optimization, iterative DPO [Tran et al., 2023, Xiong et al., 2024, Pang et al., 2024b] has proven effective. Most works focus on self-training by improving in $\mathcal{Y} \mid \mathcal{X}$, while we **jointly optimize** both responses and prompts via generative exploration in $(\mathcal{X}, \mathcal{Y})$.

Prompt synthesis for language models. Major works include Self-Instruct [Wang et al., 2022b], WizardLM [Xu et al., 2023a, Luo et al., 2023], Self-Align [Sun et al., 2024b], Evo-

4. We use `gemini-1.5` as the generative scorer.

Prompt [Guo et al., 2023], Magpie [Xu et al., 2024b], etc. `eva` is orthogonal to them since any such method can be plugged in as the `evolve`$(\cdot)$ for the creator. We focus on the **RL post-training** phase, while those works are primarily SFT. Importantly, our work proposes **an adaptive metric** to sample and evolve prompts – a secret sauce is that prompts shall be prioritized *w.r.t.* the *informativeness*, whereas prior works mostly evolve *uniformly* [Yuan et al., 2024b]. Furthermore, most works evolves in an *offline* manner, while `eva`, to our knowledge, is the first framework that supports **online/on-policy evolving** in general RL post-training.

Active and curriculum learning. This line of works re-order training examples for efficiency [Bengio et al., 2009, Mindermann et al., 2022, Kawaguchi and Lu, 2020], with recent LLM-related works [Muldrew et al., 2024, Das et al., 2024] (note those are done at a much smaller scale compared to `eva`). In contrast, `eva` breaks free from the static paradigm, not only re-orders data but also **generatively creates** new data, yielding significant gains.

Self-play and curriculum RL. Agents trained on a fixed data distribution are often brittle and may struggle to adapt to the real world [Hughes et al., 2024a]. Self-play [Samuel, 1959, Goodfellow et al., 2014a, Silver et al., 2016a] addresses this by having the agent learn through self-interaction, thus creating more diverse experiences and automatic curricula. In asymmetric self-play, the paradigm centers on "*Alice proposing a task, and Bob doing it*" [Sukhbaatar et al., 2017, Samvelyan et al., 2023, Beukman et al., 2024a, Dennis et al., 2020]. We revive the classical asymmetric self-play [Sutton et al., 2011] in optimizing language models. Unlike traditional curriculum RL [Parker-Holder et al., 2022], which renders environments by specifying levels [Dennis et al., 2020], our approach is *generative* by nature, as we directly **generate states** from powerful generative models.

Self-play in RLHF. A growing line of research frames RLHF as a *symmetric* self-play game, where both players are response players [Munos et al., 2023b, Wu et al., 2024b, Choi

et al., 2024a, Rosset et al., 2024]. However, these methods still rely on a fixed prompt distribution thus is sub-optimal. In contrast, we solve this by **asymmetric** self-play, enabling evolving prompt distributions; this step is structurely resembles adversarial training [Goodfellow et al., 2014b, Ho and Ermon, 2016]. Zheng et al. [2024] is a concurrent work with similar asymmetric setup, however it applies to safety tasks instead of general instruction-following tasks and is incompatible with direct preference optimization.

2.6 Limitations and Conclusions

Limitations. While `eva` introduces a new paradigm for RL post-training, several limitations remain. (i) The method relies on a fixed oracle reward model (RM), as is standard practice in industry [Team et al., 2024a]. However, since policy updates under `eva` can generate out-of-distribution prompts, this static assumption may lead to reward misspecification without continual RM training [Makar-Limanov et al., 2024]. (ii) The creator policy is non-differentiable in our formulation, which limits the range of optimization techniques that can be applied. (iii) The framework has so far been evaluated primarily in standard post-training tasks, and its applicability to more complex reasoning problems remains untested [Poesia et al., 2024]. (iv) Our current design is restricted to text-based, two-player interactions; extending to richer modalities [Bruce et al., 2024] or additional roles (*e.g.*, rewriters [Kumar et al., 2024]) would require substantial modifications.

Conclusions. This empirical work presents `eva`, a new, simple and scalable framework for RL post-training of LLMs, that **adaptively generates** prompt distributions during training. `eva` is simple – it can be easily plugged into any existing pipeline, and highly effective – it reaches new SOTA on challenging alignment benchmarks. The primary takeaway may be: (i) self-evolving joint training distributions $(\mathcal{X}, \mathcal{Y})$ brings significant gain, and (ii) reward advantage acts as an effective metric informing the collection and creation of prompts in

RLHF. Philosophically, **eva** presents a new view of post-training as an infinite game [Abel et al., 2024a]; **eva** incentivizes agents to create problems rather than to simply solve, which is key to intelligence, yet LLM trainers may neglect.

CHAPTER 3

HIERARCHICAL SELF-PLAY FOR NEURAL THEOREM PROVING

3.1 Introduction

The broad question we aim to address in this work is: what is an effective learning mechanism for language models to solve complex reasoning problems, such as mathematical theorem proving?

Recent progress in language models have shown promises in generating intermediate steps by next-token prediction [Wei et al., 2022], yet the performance often deteriorates when facing long trajectories or vast spaces. This challenge is particularly evident in automated theorem proving, a task that has been at the core of artificial intelligence research since the field's early days [Simon, 1969]. The process of crafting a proof is a classical example of reasoning [Wang, 1961]: just as a learning agent needs generalize from a limited set of examples to the broader set of all possible worlds, a prover must navigate from a given set of known theorems to the vast space of provable statements. An effective strategy from human mathematicians is the decomposition of problems with a sequence of target goals. This provides a more informative direction for subsequent reasoning steps, potentially reducing the effective search space.

We hereby introduce **Reasoning in Reasoning** (**RiR**), a fundamental hierarchical framework unifying *decomposing* and *search*. In the context of theorem proving, our framework consists of an offline co-training stage followed by an online goal-driven bi-level planning stage. Our contributions are:

- **Framework**: We develop Reasoning in Reasoning (RiR), a new and general reasoning framework, that is practically implemented with goal-driven hierarchical learning for neural theorem proving.

- **Experiments**: We show that RiR achieves both state-of-the-art performance and efficiency on popular benchmarks of LeanDojo [Yang et al., 2023] and miniF2F [Zheng et al., 2021].

3.2 Preliminaries: Classical Neural Theorem Proving with Language Models

We here introduce classical methods. The glossary used throughout the paper is as follows.

theorem statement ($\mathbf{q}$)	a mathematical statement.
goal ($\mathbf{s}$)	a statement in the context of a proofsearch, denoted as $\mathbf{s}$.
state	a representation containing contexts (*e.g.*, hypotheses) and goals for the proof; for simplicity, we also use this term interchangeably with *goal*.
proofstep / tactic ($\mathbf{y}$)	a reasoning step that uses established assumptions etc to achieve the goal.

Setup. We frame formal theorem proving as a Markov Decision Process. Starting with a to-prove statement $\mathbf{q}$ whose initial state is $\mathbf{s}_0$, we sequentially apply tactics $\mathbf{y}_t$ to prove it. Each tactic applied will make the current state $\mathbf{s}_t$ transit to the next state $\boldsymbol{s}_{t+1}$.

Each state is associated with a scalar reward, $r(\mathbf{s}_t)$, provided by the environment. Below we show an example in Lean4.

```
theorem (p q: Prop): p ∨ q → q ∨ p := by -- goal s₀: (p q: Prop) p ∨ q → q ∨ p
  intro h                               -- goal s₁: (p q: Prop)(h: p ∨ q) → q ∨ p
  cases h with                          -- goal s₂: (p q: Prop)(hp: p) → q ∨ p
  inl hp => apply Or.inr; exact hp      -- goal s₃: (p q: Prop)(hq: q) → q ∨ p
  inr hq => apply Or.inl; exact hq      -- goal s₄: None
```

Neural theorem proving. A neural network parameterized by $\boldsymbol{\theta}$ can act as a policy that samples single tactic $\mathbf{y}_{t+1} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s}_t)$ at step t. The objective is to find the optimal trajectory which leads to `solved` for each statement $\mathbf{q}$, that is to find a sequence of tactics

$\mathbf{y}_1, \ldots, \mathbf{y}_T$ such that:

$$\mathbf{s}_0 \xrightarrow{\mathbf{y}_1} \mathbf{s}_1 \xrightarrow{\mathbf{y}_2} \mathbf{s}_2 \xrightarrow{\mathbf{y}_3} \ldots \xrightarrow{\mathbf{y}_T} \mathbf{s}_T.$$

The problem of automated theorem proving is often solved via a two-stage framework as follows.

Stage 1: learning for proofstep generation. Classical approaches [Han et al., 2021, Welleck et al., 2022a, Yang et al., 2023, Li et al., 2024c] fine-tune a model $p_{\boldsymbol{\theta}}(\mathbf{y}^\star \mid \mathbf{s})$ to sample the next proofstep $\mathbf{y}$ conditional **only on current goal** $\mathbf{s}$.

The classical prompt format is:

`> Input:` {$current_goal s} `> Output:` {$proofstep $\mathbf{y}^\star$}

Stage 2: search for complete proof. Classically, given a statement $\boldsymbol{q}$, a full proof $\bar{\boldsymbol{y}}_{1:T}$ is found by constructing a tree [Yang et al., 2023, Li et al., 2024c] with **only low-level tactic search**. A common choice is *best-first search*, where there is a priority queue $\mathcal{Q}$ of partial proofs, ordered by some value function $v(\cdot)$. At step t, we pop one partial proof $\bar{\boldsymbol{y}}_{1:t}$ (each associated with its current state $\boldsymbol{s}_t$) with the highest value.

We then expand $\bar{\boldsymbol{y}}_{1:t}$ by generating M candidate proofsteps, and each resulting partial proof $\bar{\boldsymbol{y}}_{1:t+1} \in \mathcal{S}_{t+1}(\bar{\boldsymbol{y}}_{1:t})$ is inserted into the queue $\mathcal{Q}_{\bar{\boldsymbol{y}}}$ prioritized by the value.

The search continues until a full proof $\bar{\boldsymbol{y}}_{1:T}$ is found, or termination criteria is reached.

3.3 Method

3.3.1 Learning Stage: Goal-Driven Co-Training

Unlike classical approaches which learn to minimize the loss with regard to the conditional distribution $p(\mathbf{y}^\star \mid \mathbf{s})$, we propose to learn the joint distribution $p(\mathbf{s}^\star_{t+1}, \mathbf{y}^\star_{t+1} \mid \mathbf{s}_t)$, where $\mathbf{s}^\star_{t+1}$

is the target goal state achieved by applying $\mathbf{y}^\star_{t+1}$. In the next, we also use $\mathbf{g}^\star$ to represent the target goal state. We consider two variants of our proposed method:

- **The Imitation Learning Objective**:

$$\boldsymbol{\theta}^\star, \phi^\star \in \arg\min_{\boldsymbol{\theta},\phi} \; -\mathbb{E}_{(\mathbf{s},\mathbf{g}^\star,\mathbf{y}^\star)\sim\mathcal{D}} \left[\log \underbrace{\pi_\phi(\mathbf{g}^\star \mid \mathbf{s})}_{\text{planner}} + \log \underbrace{\pi_{\boldsymbol{\theta}}(\mathbf{y}^\star \mid \mathbf{s}, \mathbf{g}^\star)}_{\text{goal-conditioned actor}} \right] \tag{3.1}$$

- **The Reinforcement Learning Objective**:

$$\boldsymbol{\theta}^\star, \phi^\star \in \arg\max_{\boldsymbol{\theta},\phi} \; \sum_t \mathbb{E}_{\underbrace{\mathbf{g}_t \sim \pi_\phi(\cdot|\mathbf{s}_t)}_{\text{planner}}} \mathbb{E}_{\underbrace{\mathbf{y}_t \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t, \mathbf{g}_t)}_{\text{goal-conditioned actor}}} \gamma^t \cdot r(\mathbf{s}_t, \mathbf{y}_t) \tag{3.2}$$

We use the following input-output prompt format in training for the theorem proving task:

Planner (Target Goal Generation):

> **Input:** [CURRENT GOAL] {$current_goal s} [TARGET GOAL]
> **Output:** {$target_goal $s^\star$}

Actor (Goal-Driven Tactic Generation):

> **Input:** [CURRENT GOAL] {$current_goal s} [TARGET GOAL] {$target_goal $s^\star$}
> [PROOFSTEP]
> **Output:** {$tactic $y^\star$}

By decomposing the decision making process into goal state generation and goal-driven proofstep generation, RiR naturally captures the hierarchical structure of the reasoning.

3.3.2 Planning Stage: Goal-Driven Hierarchical Search

Algorithm 2 is a general design for RiR during the planning phase, where we can plug in various practical tree search policies. The high-level search explores promising target goals, while the low-level search finds the tactics to achieve each target goal, similar to the classical

leader-follower game. A key feature is the joint update of both trees. A concrete algorithm with best-first search (BFS) that we currently deploy for experiments is in Algorithm 3.

Algorithm 2 Hierarchical Training of `RiR`

Input: problem statement $\boldsymbol{q}$, a language model w/ parameter $\boldsymbol{\theta}$

$\overline{\texttt{tree}} \leftarrow \overline{\texttt{Tree}}(\boldsymbol{\theta}, \boldsymbol{q})$
repeat
  $\boldsymbol{s}_l^\star \leftarrow \overline{\texttt{tree}}.\texttt{policy}()$ ▷ /* planner */
  $\underline{\texttt{tree}} \leftarrow \underline{\texttt{Tree}}(\boldsymbol{\theta}, \boldsymbol{s}_l^\star)$
  repeat
    $\boldsymbol{y}_{t_l} \leftarrow \underline{\texttt{tree}}.\texttt{policy}()$ ▷ /* goal-driven actor */
  until STOP_LOW
  $\{\overline{\texttt{tree}}, \underline{\texttt{tree}}\}.\texttt{update}()$ ▷ /* joint update */
until STOP_HIGH
return $\underline{\texttt{tree}}.\texttt{solution}$

Algorithm 3 Hierarchical Inference of of `RiR` by Best-First Search

Input: problem statement $\boldsymbol{q}$, a language model with parameter $\boldsymbol{\theta}$

$\mathcal{Q} \leftarrow \textsc{Queue}(\boldsymbol{q})$
while $\mathcal{Q} \neq \emptyset$ and not $\textsc{BudgetExhausted}()$ **do**
  $\tau = (\underline{\boldsymbol{s}_0, (\hat{\boldsymbol{s}}_1^\star, \hat{\boldsymbol{y}}_1)}, \ldots, \underline{\boldsymbol{s}_{t-1}, (\hat{\boldsymbol{s}}_t^\star, \hat{\boldsymbol{y}}_t)}, \boldsymbol{s}_t) \leftarrow \mathcal{Q}.\textsc{Pop}()$
  if $\boldsymbol{s}_t$ is $\textsc{ProofFinished}$ **then return** τ
  end if
  $\mathcal{G}_t \leftarrow \textsc{SampleTargetGoals}(\boldsymbol{s}_t, \boldsymbol{\theta})$ ▷ /* high-level search */
  for $\hat{\boldsymbol{s}}_t^{\star(i)} \in \mathcal{G}_t$ **do**
    $\mathcal{Y}_t^{(i)} \leftarrow \textsc{SampleTactics}(\hat{\boldsymbol{s}}_{t+1}^{\star(i)}, \boldsymbol{s}_t, \boldsymbol{\theta})$ ▷ /* low-level search */
    for $\hat{\boldsymbol{y}}_{t+1}^{(i,j)} \in \mathcal{Y}_t^{(i)}$ **do**
      $\boldsymbol{s}_{t+1}^{(i,j)} \leftarrow \textsc{ApplyTactic}(\hat{\boldsymbol{y}}_{t+1}^{(i,j)}, \boldsymbol{s}_t)$
      $\tau' \leftarrow (\underline{\boldsymbol{s}_0, (\hat{\boldsymbol{s}}_1^\star, \hat{\boldsymbol{y}}_1)}, \ldots, \underline{\boldsymbol{s}_t, (\hat{\boldsymbol{s}}_{t+1}^\star, \hat{\boldsymbol{y}}_{t+1})}, \boldsymbol{s}_{t+1}^{(i,j)})$
      $\mathcal{Q}.\textsc{Push}(\tau', -\log p(\tau'))$ ▷ /* joint update */
    end for
  end for
end while

3.4 Experiments

Setups: Datasets and Models. We use the random split of LeanDojo Benchmark 4 [Yang et al., 2023] as the training dataset. We use ByT5-0.3B [Xue et al., 2021] as our base model, which is a pretrained byte-level encoder-decoder Transformer model, and was adopted in Yang et al. [2023] with the state-of-the-art performance in theorem proving.

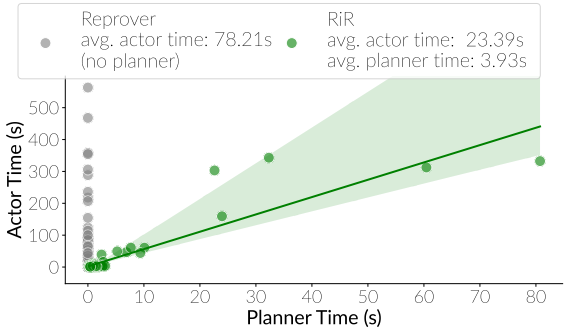

Figure 3.1: **Efficiency**. The scatter plot for actor and planner time spent for proved theorems on miniF2F. RiR significantly reduces the actor time via the goal guidance from the planner.

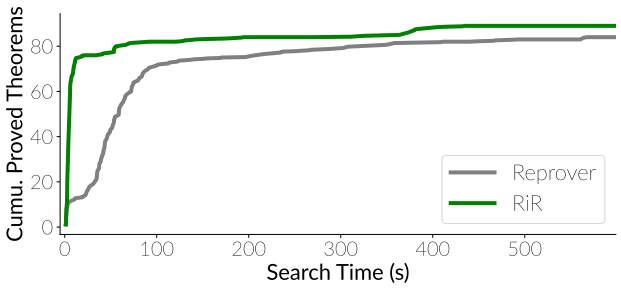

Figure 3.2: **Efficiency**. The CDF plot for search time spent for proved theorems on miniF2F Benchmark. RiR is significantly faster (nearly **3x**) than the existing state-of-the-art baseline.

We refer to this trained checkpoint of Reprover (w/o retrieval) as our baseline, and evaluate it *with the same setting* as RiR. We train the above model for 500K steps, with the learning rate as 5.0×10^{-4} and batch size as 8. For evaluation, we use both LeanDojo Benchmark 4 and miniF2F [Zheng et al., 2021]. We use the *Pass@1* metric with 10-min timeout limit for evaluation. The search width for the high-level and the low-level is 5 and 64.

Results: Performance Gain. We present the performance comparison of RiR with existing baselines in Table 3.1. RiR also proved 1 more AIME and 2 more AMC problems compared to the current state-of-the-art Reprover [Yang et al., 2023].

Dataset (→)	**miniF2F-test**[1]	**LeanDojo-test**
Method (↓) / Model (→)	ByT5-0.3B	ByT5-0.3B
Reprover-SFT (BFS)	34.43%	50.16%
RiR-SFT (BFS)	**36.89%**	53.73%
RiR-RL (BFS)	36.73%	**54.01%**

Table 3.1: **Performance with BFS.** *Pass@1* rate on LeanDojo and miniF2F.

2. Additional data points on miniF2F-test for readers' reference:
- Azerbayev et al. [2023] Llemma-7B achieves 26.2%;
- Welleck and Saha [2023] LLMStep-1.8B achieves 27.9%;
- Polu and Sutskever [2020] GPT-f achieves 36.6%.

Dataset (→)	miniF2F-test[2]	LeanDojo-test
Method (↓) / Model (→)	ByT5-0.3B	ByT5-0.3B
Reprover-SFT (MCTS)	36.51%	50.24%
RiR-SFT (MCTS)	37.83%	53.92%
RiR-RL (MCTS)	**37.95%**	**54.24%**

Table 3.2: **Performance with MCTS.** *Pass@1* rate on LeanDojo and miniF2F.

Results: Efficiency Gain. RiR is significantly faster in searching for the optimal reasoning trajectories via a more compact and information-directed search space with the goal-driven planner, as illustrated in Figure 3.2 on miniF2F benchmark. RiR is more time efficient in the sense that we achieve better results in small computational budget. Specifically, as shown in Figure 3.1, while the classical Reprover has an average actor time (*i.e.*, time spent for low-level proofstep search) of 78.21s, RiR reduces this to only **23.39**s, with additional 3.93s for planner time (*i.e.*, time spent for high-level goal search) on average, setting the new efficiency benchmark for neural theorem proving.

Remarks. We present logs showing how RiR found hard proofs fast while classical approaches fail in Appendix; take `Finset.union_subset_left` for example, while the classical method expanded more than 8914 nodes yet still failed after 10 minutes, RiR proved the theorem within 5 seconds and only searched 1 node. We believe the significant improvement in efficiency and effectiveness comes from RiR's ability to generalize better and to explore better in the more compact and informative search space.

3.5 Related Works

Reasoning with language models. In language modeling, reasoning typically refers to generating intermediate steps within the language space to reach a final solution to a problem [Wei et al., 2022]. Solving complex or novel reasoning problems remains as an open challenge.One promising direction is **reasoning by searching**, *e.g.*, expanding the reasoning

space by tree search for intermediate steps [Yao et al., 2024, Feng et al., 2023, Liu et al., 2023a, Yuan et al., 2024a]. Another research direction is **reasoning by decomposition**, *i.e.*, generating higher-level goals that trigger a single or a sequence of lower-level steps [Zhou et al., 2022, Liu et al., 2023e, Zheng et al., 2023d, Liang et al., 2024, Dalal et al., 2024, Huang et al., 2022b, Hu et al., 2024b]. The most similar line of literature to ours is subgoal search [Wilkins, 1980, Czechowski et al., 2021, Zawalski et al., 2022, Parascandolo et al., 2020, Paul et al., 2019], while we put specific focus on the theorem proving benchmarks, and present initial theoretical conjectures, and formally unifies *search* and *decomposing* in a hierarchical framework for large language model training and inference.

Automatic Theorem Proving with language models. As a representative reasoning task, automatic theorem proving (ATP) is often characterized as a *tree search* problem, *i.e.*, constructing a (tactic-based) proof tree and traversing it to find the correct proof [Li et al., 2024c]. In the context of language modeling, *proofstep generation* forms the edges of the proof tree; the common standard in prior works is to generate single proof steps with the input format similar to `[GOAL]${goal}[PROOFSTEP]`, *i.e.*, conditional on the current goal, generating the next tactic [Polu and Sutskever, 2020, Yang et al., 2023, Azerbayev et al., 2023, Lample et al., 2022]. For the proof search stage, while people have been using simple heuristics like breadth-first search [Bansal et al., 2019], or MCTS-like search guided by learned value functions [Lample et al., 2022, Polu et al., 2022], designing better search algorithms remains an active area [Li et al., 2024c]. The key challenge is that the tactic-based proof space is combinatorially large. Distinguished from prior works, RiR introduces the goal-driven co-training for proofstep generation with a bi-level search framework for generalization and efficiency advantage.

Hierarchical and goal-conditioned RL. Planning and learning is hard when the decision-making space scales up [Bakker et al., 2004]. Hierarhical RL intends to address this issue

by learning a hierarchy of policies operating on different levels of abstraction (*e.g.*, subgoals over the state space). This mitigates the scaling issues by improving exploration for the environment [Ghosh et al., 2020, Chitnis et al., 2022, Kumar et al., 2023a, Silver et al., 2023a, Le et al., 2018]. There is another line of research termed as goal-conditioned RL [Ghosh et al., 2019b, Wang et al., 2023b, Ghugare et al., 2024a], which trains offline RL policy in a supervised manner conditioning on goal or return. Unlike most prior works that rely on a predefined goal structure, we train models to *learn to generate goals* in the language space, and refine the goal planning via low-level tree search and joint update.

CHAPTER 4

LOOKAHEAD WITH ABSTRACT WORLD MODELS FOR LONG-HORIZON AGENTIC TASKS

The complete package will be updated on: link.

4.1 Introduction

Backgrounds. Training agents that can use tools to solve complex real-world tasks is an important step toward artificial general intelligence [Mialon et al., 2023]. Many real-world problems cannot be solved with a single action or tool call. Instead, they require multiple steps that may involve using a bash terminal, a Python interpreter, a web browser, dataset retrieval utilities, image processors, or external APIs [Jiang et al., 2025a, Li et al., 2023a].

Challenges. However, scaling such long-horizon trajectory training remains difficult due to the low data efficiency of current model-free approaches [Goldie et al., 2025, Yu et al., 2024a, Li et al., 2025c]. Model-based methods have traditionally improved data efficiency in classical reinforcement learning tasks [Murphy, 2024], but directly learning a model of the real world with large language models is often impractical. Tool use typically involves real-time interaction (*e.g.*, web search), where the environment changes as new information appears. A model trained only on past trajectories aiming to simulate full details of the world can therefore become outdated and hallucinating, leading to poor policy updates.

Our contributions. In this work, we propose to learn an *abstract world model* that captures the high-level structure of tool-using behavior without modeling the full external world. During decision time, the agent performs abstract lookahead with this model for action selection and guide policy learning towards high-reward trajectories, while continually updating the model itself with new feedback from the environment to remain adaptive and grounded.

4.2 Related Works

Model-based RL. On a high-level, this line of research is about learning a model of the world (on state transitioning and rewarding), and using the learned model to guide policy learning, which has shown to be more sample efficient compared to purely model-free methods [Alver and Precup, 2022, Murphy, 2024]. There are typically two types of model-based RL, decision-time planning and background planning. In (online) decision-time planning [Kaelbling and Lozano-Pérez, 2011, Silver et al., 2016a, 2017a, Schrittwieser et al., 2020, Ye et al., 2021a, Wang et al., 2024d, Macfarlane et al., 2024], we use the model rollout to guide the action *selection* during policy rollout; in (offline) background planning [Sutton, 1990, Rajeswaran et al., 2020a, Hafner et al., 2023], we use the full model rollout trajectories to train the policy. Our work integrates both approaches with an *abstract* world model (which is helpful in both accelerating planning and reducing hallucination); different from past works like MuZero [Schrittwieser et al., 2020], the abstraction in our case is achieved semantically by calling language models.

RL for language agents with tool use. Beyond classical supervised fine-tuning with offline trajectories with tool use [Schick et al., 2023, Qin et al., 2023, Hu et al., 2025], recent literature explores using RL for training language agents, effectively improving the training data coverage through online rollouts. However, those works are mainly restricted to *model-free* setting with either trajectory-level RL where tool calling results are masked [Singh et al., 2025], or stepwise RL [Yu et al., 2024a, Goldie et al., 2025, Li et al., 2025c, Chen et al., 2025, Jiang et al., 2025a, Jin et al., 2025, Qian et al., 2025]; both of which are sample inefficient. Differently, we first introduce a *model-based* RL method to improve the tool calling ability of LLMs.

LLMs as world models. Existing works have considered using LLMs as the world models to improve test-time accuracy of textual reasoning [Hao et al., 2023]. Recently, there has been works on using LLMs as world models for code tasks, which an LLM is employed to model program execution feedback [Tang et al., 2024a, Carbonneaux et al., 2025, Lehrach et al., 2025]. Differently, in our work, we introduce *abstraction* in the world modeling, which is more sample efficient, more robust to hallucinations when tool calling is involved, and thus more generalizable to a broader task space.

4.3 Method

4.3.1 Problem Setup

A typical trajectory for an LLM $\boldsymbol{\theta}$ to complete a task with multi-step tool interaction can be written as:

$$\overbrace{(\underbrace{\underbrace{\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{e}_0}_{\boldsymbol{s}_1}, \boldsymbol{a}_1, \boldsymbol{e}_1}^{\boldsymbol{s}_2}, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{T-1}, \boldsymbol{e}_{T-1}}^{\boldsymbol{s}_T}, \boldsymbol{a}_T)} \tag{4.1}$$

Here, $\boldsymbol{s}_0$ is the initial task prompt, $\boldsymbol{a}_t$ is the model action at step t (which includes reasoning texts and potential tool invocation), $\boldsymbol{e}_t$ is the optional feedback from the environment w.r.t. the tool interaction, and the state $\boldsymbol{s}_t$ is defined as the full interaction history *up to* that step.

Existing works apply RL with a stepwise policy $\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{s}_t)$, with the following objective, optimized by REINFORCE-style *model-free* methods [Goldie et al., 2025, Qian et al., 2025, Yu et al., 2024a]:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}_0 \sim \rho,\ \boldsymbol{a}_t \sim \pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t),\ \boldsymbol{s}_{t+1} \sim \mathcal{T}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)} \sum_{t=0}^{T-1} r_t\left(s_t, a_t, s_{t+1}\right). \tag{4.2}$$

However, since such model-free approaches learn only from sampled trajectories, they require large amounts of interaction data and provide no mechanism for reasoning about unseen outcomes. This makes long-horizon learning and generalization across diverse tool-use sequences especially inefficient.

While model-based reinforcement learning [Alver and Precup, 2022] can, in principle, improve data efficiency by enabling planning towards higher-rewarding trajectories, directly modeling the real-world environment of tool-using agents may be infeasible and overkill. Specifically, the external world evolves with new information (for example, through web search or API responses), and attempting to mimic full details of such dynamics from static trajectories may lead to hallucination and poor generalization. Instead, we propose to learn an abstract world model that captures the high-level semantic structure of interaction. We describe the details of our algorithm in the next.

4.3.2 Algorithms

To explain, we first learn an *abstract world model* and use it to perform multiple lookahead rollouts at each decision step, guiding the policy toward high-reward trajectories. Given access to a warmup dataset $\mathcal{D}$ containing prior trajectories, a frozen LLM is used as an abstractor Φ to convert raw tool feedback into compact semantic summaries; the same abstractor can also act as a judge, providing stepwise reward signals as in Goldie et al. [2025]. We then train a model $\mathcal{M}_{\boldsymbol{\theta}}^{\tilde{\mathbf{z}}}$ to learn abstract transitions, immediate reward r and value Q of actions, and action dynamics in this latent space. At decision time, the agent performs lookahead rollouts within this abstract model to evaluate candidate actions and select those with the highest predicted return – conceptually similar to MuZero [Schrittwieser et al., 2020] with *hypothetical* latent rollouts, but planning in an abstract semantic space.

Algorithm 4 Ours: lookahead with abstract models

Input: a initial state distribution ρ, a base LLM model $\boldsymbol{\theta}$, an offline warmup dataset $\mathcal{D}$, a fixed abstractor Φ

▽ /* (offline warmup) abstract world model + policy learning */

$\mathcal{D}^{\boldsymbol{z}} \leftarrow \Phi(\mathcal{D})$ ▷ /* convert each real $\boldsymbol{e}$ to an abstract $\boldsymbol{z}$ */

$\mathcal{M}^{\boldsymbol{z}}_{\boldsymbol{\theta}} \leftarrow \mathsf{update\text{-}model}(\boldsymbol{\theta}, \mathcal{D}^{\boldsymbol{z}})$ ▷ /* learn abstract model */

$\pi_{\boldsymbol{\theta}} \leftarrow \mathsf{update\text{-}policy}(\boldsymbol{\theta}, \mathcal{D} \cup \mathcal{D}^{\boldsymbol{z}})$ ▷ /* learn policy */

▽ /* (online) update policy with abstract model lookahead */

for $\boldsymbol{s}_0 \sim \rho$ **do**

  repeat

    $\{\boldsymbol{a}_t^{(k)}\} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{s}_t)$ ▷ /* sample candidate actions */

    $\{\hat{Q}_t^{(k)}\} \leftarrow \mathsf{lookahead}(\boldsymbol{s}_t, \{\boldsymbol{a}_t^{(k)}\}, \mathcal{M}^{\boldsymbol{z}}_{\boldsymbol{\theta}})$ ▷ /* value abstract rollout */

    $\boldsymbol{a}_t^{\star} \leftarrow \mathsf{choose}(\{\boldsymbol{a}_t^{(k)}, \hat{Q}_t^{(k)}\})$ ▷ /* select one greedy action */

    $(\boldsymbol{e}_t, r_t) \leftarrow \mathsf{env\text{-}step}(\boldsymbol{a}_t^{\star})$ ▷ /* commit the action in real env */

  until stop

  update $\mathcal{M}^{\boldsymbol{z}}_{\boldsymbol{\theta}}$ and $\pi_{\boldsymbol{\theta}}$ with new data

end for

For comparison, the baseline algorithm [Yu et al., 2024a, Goldie et al., 2025, Jiang et al., 2025a] follows the same training and interaction procedure but omits the abstract model components. Specifically, the policy $\pi_{\boldsymbol{\theta}}$ is first initialized through supervised fine-tuning (SFT) on the warmup dataset $\mathcal{D}$, and then refined by online rollouts under real environment feedback. Unlike our approach, it selects actions directly from $\pi_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{s}_t)$ without performing lookahead or receiving guidance from the abstract model $\mathcal{M}^{\boldsymbol{z}}_{\boldsymbol{\theta}}$.

4.4 Experiments

4.4.1 Setup

Evaluation benchmarks. We evaluate the method on (i) search-related benchmark, which evaluates the LLM's ability to conduct web search and retrieval, including HotPotQA [Yang et al., 2018], MuSiQue [Trivedi et al., 2022], and 2wiki [Ho et al., 2020]; and (ii) math-related benchmark, which evaluates the LLM's ability to use Python interpreter with Sympy toolkit [Meurer et al., 2017], including MATH500 [Lightman et al., 2023], AIME [Veeraboina, 2023], and Olympiad Bench [He et al., 2024].

Datasets, models, and training specification. Following existing works [Yu et al., 2024a, Goldie et al., 2025, Li et al., 2025c, Chen et al., 2025, Jiang et al., 2025a, Jin et al., 2025, Qian et al., 2025], we sample a mix of subsets from the following datasets with the same filtering procedure as in [Hu et al., 2025]: for search-related tools, 1K from HotPotQA [Yang et al., 2018], 1K from WikiTableQuestions [Pasupat and Liang, 2015], 1K from ToolACE [Liu et al., 2024]; for math-related tools, 1K from GSM8K [Cobbe et al., 2021], 2.5K from Numina Math [LI et al., 2024], 2.5K from MATH [Hendrycks et al., 2021b], all with synthetic sympy trajectories [Li et al., 2023a]. We use Qwen-2.5-7B-Instruct [Team, 2024] as the baseline model for search-related training and evaluation, while using its math version for math-related ones. During training, we set the lookahead horizon to be until termination, and the number of candidate actions sampled each step to be 6. We use GPT-4o [Achiam et al., 2023] as the abstractor and the generative reward model. We use standard negative log-likelihood loss for the abstract model learning, and group-relative policy optimization for the policy update, following standard hyperparameter setting as in [Jiang et al., 2025a].

4.4.2 Results

Table 4.1: Results on search-related benchmark.

	HotPotQA	**MuSiQue**	**2wiki**
Direct Inference	18.3	3.1	25.0
SFT	21.7	6.6	25.9
RL (model-free) [Jin et al., 2025]	32.6	12.5	29.7
Ours (abstract model-based RL)	44.9	32.1	36.7

Table 4.2: Results on math-related benchmark with sympy tools.

	MATH500	**AIME24**	**Olympiad Bench**
Direct Inference	82.4	16.2	41.0
SFT	78.2	18.4	43.1
RL (model-free) [Jiang et al., 2025a]	83.2	43.3	50.5
Ours (abstract model-based RL)	84.5	39.4	51.8

4.5 Conclusions and Future Works

By performing lookahead in the abstract space, the agent can reason about future outcomes and guide policy learning more efficiently than model-free approaches. In future work, we plan to explore more flexible formulations of the latent abstraction built-in the transformer models and develop more efficient strategies for sampling abstract trajectories [Wang et al., 2024d], aiming to further improve the scalability and stability of abstract model-based training.

CHAPTER 5

CONCLUSION

5.1 Conclusions and Dissertation Contributions

This dissertation presents a unified research agenda on *self-play methods for reinforcement learning of language models*, advancing our understanding of how agents can not only learn to act but also learn what to experience.

The major contributions can be summarized as follows:

- **Reward-Guided Prompt Evolving (Chapter 2):** We proposed Evolving via Asymmetric Self-Play (eva) [Ye et al., 2024b], a practical framework that enables large language models to generate and prioritize informative prompts during reinforcement learning. By optimizing a minimax-regret objective between a creator and solver policy, eva improves alignment efficiency and generalization, establishing new state-of-the-art results across standard RLHF benchmarks.

- **Hierarchical Self-Play for Reasoning (Chapter 3):** We introduced Reasoning-in-Reasoning (RiR) [Ye et al., 2024c], a hierarchical self-play framework for neural theorem proving. RiR allows the model to decompose complex reasoning problems into goal-driven subproblems, combining offline co-training with online hierarchical planning. Experiments on LeanDojo and miniF2F show substantial gains in both accuracy and computational efficiency.

- **Abstract World Model Lookahead (Chapter 4):** We developed a model-based reinforcement learning framework where an *abstract world model* enables fast lookahead planning for long-horizon tool-use tasks. By integrating latent model rollouts with policy optimization, the method improves sample efficiency and generalization on multi-step tool use tasks.

Together, these chapters demonstrate that *self-play experience shaping*provides a useful foundation for training general-purpose reasoning agents.

5.2 Future Research Directions

Several promising directions emerge from this dissertation. One direction is to develop continual and interactive reward modeling, where reward models co-evolve with policies to mitigate reward misspecification in open-ended environments. Another is to extend the creator into a differentiable, trainable policy, which may enable gradient-based meta-learning and richer multi-role interactions beyond the two-player paradigm. Future work may also explore abstract model–based agents by designing better latent abstractions through architectural innovation, such as combining diffusion world models [Ding et al., 2024b], and by developing more efficient sampling and rollout strategies [Macfarlane et al., 2024, Wang et al., 2024d]. Finally, establishing the theoretical foundations of experience shaping remains an important open challenge, particularly in formally characterizing the underlying learning dynamics.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on Lifelong Learning Agents*, pages 620–636. PMLR, 2023a.

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on Lifelong Learning Agents*, pages 620–636. PMLR, 2023b.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado P van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024a.

David Abel, Mark K Ho, and Anna Harutyunyan. Three dogmas of reinforcement learning. *arXiv preprint arXiv:2407.10583*, 2024b.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340B Technical Report. *arXiv preprint arXiv:2406.11704*, 2024.

Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. The cringe loss: Learning what language not to model. *arXiv preprint arXiv:2211.05826*, 2022.

Constructions Aeronautiques, Adele Howe, Craig Knoblock, ISI Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, David Wilkins SRI, Anthony Barrett, Dave Christianson, et al. Pddl| the planning domain definition language. *Technical Report, Tech. Rep.*, 1998.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Safa Alver and Doina Precup. A look at value-based decision-time vs. background planning methods across different settings. *arXiv preprint arXiv:2206.08442*, 2022.

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct Preference Optimization with an Offset. *arXiv preprint arXiv:2402.10571*, 2024.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *International conference on machine learning*, pages 166–175. PMLR, 2017.

Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger Grosse. Learning to give checkable answers with prover-verifier games. *arXiv preprint arXiv:2108.12099*, 2021.

Richard Antonello, Nicole Beckage, Javier Turek, and Alexander Huth. Selecting informative contexts improves language model finetuning. *arXiv preprint arXiv:2005.00175*, 2020.

Hannah Arendt. The Origins of Totalitarianism. *The American Historical Review*, 1958.

Kushal Arora, Timothy J O'Donnell, Doina Precup, Jason Weston, and Jackie CK Cheung. The Stable Entropy Hypothesis and Entropy-Aware Decoding: An Analysis and Algorithm for Robust Natural Language Generation. *arXiv preprint arXiv:2302.06784*, 2023.

Kenneth J Arrow. *Social choice and individual values*, volume 12. Yale university press, 1952.

Dilip Arumugam and Benjamin Van Roy. Deciding what to model: Value-equivalent sampling for reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 9024–9044, 2022.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Jordan T Ash and Ryan P Adams. On the difficulty of warm-starting neural network training. *arXiv preprint arXiv:1910.08475*, 2019.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems*, 33:2159–2170, 2020.

Bram Bakker, Jürgen Schmidhuber, et al. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, pages 438–445. Citeseer, 2004.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

Alfredo Banos. On pseudo-games. *The Annals of Mathematical Statistics*, 39(6):1932–1945, 1968.

Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. Holist: An environment for machine learning of higher order logic theorem proving. In *International Conference on Machine Learning*, pages 454–463. PMLR, 2019.

Adrien Baranes and Pierre-Yves Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36. JMLR Workshop and Conference Proceedings, 2012.

Yoshua Bengio and Nikolay Malkin. Machine learning and information theory concepts towards an AI Mathematician. *Bulletin of the American Mathematical Society*, 61(3): 457–469, 2024a.

Yoshua Bengio and Nikolay Malkin. Machine learning and information theory concepts towards an AI Mathematician. *arXiv preprint arXiv:2403.04571*, 2024b.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Michael Beukman, Samuel Coward, Michael Matthews, Mattie Fellows, Minqi Jiang, Michael Dennis, and Jakob Foerster. Refining Minimax Regret for Unsupervised Environment Design. *arXiv preprint arXiv:2402.12284*, 2024a.

Michael Beukman, Samuel Coward, Michael Matthews, Mattie Fellows, Minqi Jiang, Michael Dennis, and Jakob Foerster. Refining Minimax Regret for Unsupervised Environment Design. *arXiv preprint arXiv:2402.12284*, 2024b.

Lasse Blaauwbroek, David Cerna, Thibault Gauthier, Jan Jakubuv, Cezary Kaliszyk, Martin Suda, and Josef Urban. Learning Guided Automated Reasoning: A Brief Survey. *arXiv preprint arXiv:2403.04017*, 2024.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

David Brandfonbrener, Sibi Raja, Tarun Prasad, Chloe Loughridge, Jianang Yang, Simon Henniger, William E Byrd, Robert Zinkov, and Nada Amin. Verified Multi-Step Synthesis using Large Language Models and Monte Carlo Tree Search. *arXiv preprint arXiv:2402.08147*, 2024.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020a. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020b.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Quentin Carbonneaux, Gal Cohen, Jonas Gehring, Jacob Kahn, Jannik Kossen, Felix Kreuk, Emily McMilin, Michel Meyer, Yuxiang Wei, David Zhang, et al. Cwm: An open-weights llm for research on code generation with world models. *arXiv preprint arXiv:2510.02387*, 2025.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Shicong Cen, Jincheng Mei, Katayoon Goshvadi, Hanjun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yuejie Chi, and Bo Dai. Value-Incentivized Preference Optimization: A Unified Approach to Online and Offline RLHF. *arXiv preprint arXiv:2405.19320*, 2024.

Seth Chaiklin et al. The zone of proximal development in Vygotsky's analysis of learning and instruction. *Vygotsky's educational theory in cultural context*, 1(2):39–64, 2003.

Huayu Chen, Guande He, Hang Su, and Jun Zhu. Noise Contrastive Alignment of Language Models with Explicit Rewards. *arXiv preprint arXiv:2402.05369*, 2024a.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*, 2023.

Ziru Chen, Michael White, Raymond Mooney, Ali Payani, Yu Su, and Huan Sun. When is Tree Search Useful for LLM Planning? It Depends on the Discriminator. *arXiv preprint arXiv:2402.10890*, 2024b.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024c.

Ching-An Cheng, Andrey Kolobov, Dipendra Misra, Allen Nie, and Adith Swaminathan. LLF-Bench: Benchmark for Interactive Learning from Language Feedback. *arXiv preprint arXiv:2312.06853*, 2023.

Ching-An Cheng, Allen Nie, and Adith Swaminathan. Trace is the Next AutoDiff: Generative Optimization with Rich Feedback, Execution Traces, and LLMs. *arXiv preprint arXiv:2406.16218*, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Rohan Chitnis, Tom Silver, Joshua B Tenenbaum, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Learning neuro-symbolic relational transition models for bilevel planning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4166–4173. IEEE, 2022.

Eugene Choi, Arash Ahmadian, Matthieu Geist, Oilvier Pietquin, and Mohammad Gheshlaghi Azar. Self-Improving Robust Preference Optimization. *arXiv preprint arXiv:2406.01660*, 2024a.

Eugene Choi, Arash Ahmadian, Matthieu Geist, Olivier Pietquin, and Mohammad Gheshlaghi Azar. Self-Improving Robust Preference Optimization. *arXiv preprint arXiv:2406.01660*, 2024b.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. Inference-Aware Fine-Tuning for Best-of-N Sampling in Large Language Models. *arXiv preprint arXiv:2412.15287*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24:1–113, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Rémi Coulom. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*, pages 72–83. Springer, 2006.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Konrad Czechowski, Tomasz Odrzygozdz, Marek Zbysinski, Michal Zawalski, Krzysztof Olejnik, Yuhuai Wu, Lukasz Kucinski, and Piotr Milos. Subgoal search for complex reasoning tasks. *Advances in Neural Information Processing Systems*, 34:624–638, 2021.

Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*, 2024.

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations*, 2021.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Provably sample efficient rlhf via active preference optimization. *arXiv preprint arXiv:2402.10500*, 2024.

Peter Dayan and Geoffrey E Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Leonardo De Moura and Nikolaj Bjørner. Satisfiability Modulo Theories: Introduction and Applications. *Communications of the ACM*, 54(9):69–77, 2011.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

Muxi Diao, Rumei Li, Shiyang Liu, Guogang Liao, Jingang Wang, Xunliang Cai, and Weiran Xu. SEAS: Self-Evolving Adversarial Safety Optimization for Large Language Models. *arXiv preprint arXiv:2408.02632*, 2024.

Mucong Ding, Souradip Chakraborty, Vibhu Agrawal, Zora Che, Alec Koppel, Mengdi Wang, Amrit Bedi, and Furong Huang. Sail: Self-improving efficient online alignment of large language models. *arXiv preprint arXiv:2406.15567*, 2024a.

Zihan Ding, Amy Zhang, Yuandong Tian, and Qinqing Zheng. Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning. *arXiv preprint arXiv:2402.03570*, 2024b.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Maintaining plasticity in deep continual learning. *arXiv preprint arXiv:2306.13812*, 2023.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. RLHF Workflow: From Reward Modeling to Online RLHF, 2024a.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024b.

Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. Neural logic machines. *ICLR*, 2019.

Shi Dong and Benjamin Van Roy. An information-theoretic analysis for thompson sampling with many actions. *Advances in Neural Information Processing Systems*, 31, 2018.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

Yuqing Du, Pieter Abbeel, and Aditya Grover. It takes four to tango: Multiagent selfplay for automatic curriculum generation. *arXiv preprint arXiv:2202.10608*, 2022.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024a.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024b.

Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient Exploration for LLMs. *arXiv preprint arXiv:2402.00396*, 2024a.

Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient Exploration for LLMs. *arXiv preprint arXiv:2402.00396*, 2024b.

Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient exploration for llms. *arXiv preprint arXiv:2402.00396*, 2024c.

Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, et al. Competitive programming with large reasoning models. *arXiv preprint arXiv:2502.06807*, 2025.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Ignacio Esponda and Demian Pouzo. Equilibrium in misspecified Markov decision processes. *arXiv preprint arXiv:1502.06901*, 2015.

Ignacio Esponda and Demian Pouzo. Berk–Nash equilibrium: A framework for modeling agents with misspecified models. *Econometrica*, 84(3):1093–1130, 2016.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018a.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018b.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Benjamin Eysenbach, Vivek Myers, Sergey Levine, and Ruslan Salakhutdinov. Contrastive representations make planning easy. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023. URL `https://openreview.net/forum?id=W0bhHvQK60`.

Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39(1):42–47, 1953.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is Wrong with Perplexity for Long-context Language Modeling? *arXiv preprint arXiv:2410.23771*, 2024.

Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. *arXiv preprint arXiv:2002.07017*, 2020.

Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.

Arnaud Fickinger, Hengyuan Hu, Brandon Amos, Stuart Russell, and Noam Brown. Scalable Online Planning via Reinforcement Learning Fine-Tuning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=D0xGh031I9m`.

Adam Fisch, Jacob Eisenstein, Vicky Zayats, Alekh Agarwal, Ahmad Beirami, Chirag Nagpal, Pete Shaw, and Jonathan Berant. Robust Preference Optimization through Reward Model Distillation. *arXiv preprint arXiv:2405.19316*, 2024.

Yoav Freund and Robert E. Schapire. (Adaptive Game Playing Using Multiplicative Weights). *Games and Economic Behavior*, 29:79–103, 1999.

Shi Fu, Fengxiang He, Xinmei Tian, and Dacheng Tao. Convergence of Bayesian Bilevel Optimization. In *The Twelfth International Conference on Learning Representations*, 2023.

Yuwei Fu, Di Wu, and Benoit Boulet. A closer look at offline rl agents. *Advances in Neural Information Processing Systems*, 35:8591–8604, 2022.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023a.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023b. URL `https://zenodo.org/records/10256836`.

Yang Gao, Dana Alon, and Donald Metzler. Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*, 2024.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019a.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019b.

Dibya Ghosh, Abhishek Gupta, Justin Fu, Ashwin Reddy, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals without reinforcement learning, 2020. URL https://openreview.net/forum?id=ByxoqJrtvr.

Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the Gap between TD Learning and Supervised Learning–A Generalisation Point of View. *arXiv preprint arXiv:2401.11237*, 2024a.

Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the gap between td learning and supervised learning–a generalisation point of view. *arXiv preprint arXiv:2401.11237*, 2024b.

Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D Manning. Synthetic data generation & multi-step rl for reasoning & tool use. *arXiv preprint arXiv:2504.04736*, 2025.

Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014b.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*, 2024a.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*, 2024b.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*, 2023.

Roger Grosse. Bilevel Optimization and Generalization, 2022.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging Pre-trained Large Language Models to Construct and Utilize World Models for Model-based Task Planning. *arXiv preprint arXiv:2305.14909*, 2023.

Lin Gui, Cristina Gârbacea, and Victor Veitch. BoNBoN Alignment for Large Language Models and the Sweetness of Best-of-n Sampling. *arXiv preprint arXiv:2406.00832*, 2024.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*, 2023.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual Pre-Training of Large Language Models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

Johan E. Gustafsson. Decisions under Ignorance, 2022. Lecture notes.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Ian Hamilton, Nick Tawn, and David Firth. The many routes to the ubiquitous Bradley-Terry model. *arXiv preprint arXiv:2312.13619*, 2023.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. *arXiv preprint arXiv:2102.06203*, 2021.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL `https://arxiv.org/abs/2402.14008`.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023a.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023b.

Yuval Heller and Eyal Winter. Biased-belief equilibrium. *American Economic Journal: Microeconomics*, 12(2):1–40, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring Coding Challenge Competence With APPS. *NeurIPS*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps, 2020. URL `https://arxiv.org/abs/2011.01060`.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Joey Hong, Branislav Kveton, Sumeet Katariya, Manzil Zaheer, and Mohammad Ghavamzadeh. Deep hierarchy in bandits. In *International Conference on Machine Learning*, pages 8833–8851. PMLR, 2022a.

Joey Hong, Branislav Kveton, Manzil Zaheer, and Mohammad Ghavamzadeh. Hierarchical bayesian bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 7724–7741. PMLR, 2022b.

Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. OpenRLHF: An Easy-to-use, Scalable and High-performance RLHF Framework. *arXiv preprint arXiv:2405.11143*, 2024a.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scene as blender code. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024b.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. *arXiv preprint arXiv:2412.01951*, 2024.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022a.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022b.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. Open-Endedness is Essential for Artificial Superhuman Intelligence. *arXiv preprint arXiv:2406.04268*, 2024a.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. Open-Endedness is Essential for Artificial Superhuman Intelligence. *arXiv preprint arXiv:2406.04268*, 2024b.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, and Yaodong Yang. Aligner: Achieving Efficient Alignment through Weak-to-Strong Correction. *arXiv preprint arXiv:2402.02416*, 2024a.

Xiang Ji, Sanjeev Kulkarni, Mengdi Wang, and Tengyang Xie. Self-Play with Adversarial Critic: Provable and Scalable Offline Alignment for Language Models. *arXiv preprint arXiv:2406.04274*, 2024b.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*, 2022a.

Albert Qiaochu Jiang, Wenda Li, and Mateja Jamnik. Multilingual Mathematical Autoformalization, 2024a. URL `https://openreview.net/forum?id=QqdloE1QH2`.

Angela H Jiang, Daniel L-K Wong, Giulio Zhou, David G Andersen, Jeffrey Dean, Gregory R Ganger, Gauri Joshi, Michael Kaminksy, Michael Kozuch, Zachary C Lipton, et al. Accelerating deep learning by focusing on the biggest losers. *arXiv preprint arXiv:1910.00762*, 2019.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025a.

Dongwei Jiang, Marcio Fonseca, and Shay B Cohen. LeanReasoner: Boosting Complex Logical Reasoning with Lean. *arXiv preprint arXiv:2403.13312*, 2024b.

Minqi Jiang. Learning Curricula in Open-Ended Worlds. *arXiv preprint arXiv:2312.03126*, 2023.

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021a.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950. PMLR, 2021b.

Minqi Jiang, Andrei Lupu, and Yoram Bachrach. Bootstrapping Task Spaces for Self-Improvement. *arXiv preprint arXiv:2509.04575*, 2025b.

Nan Jiang. Notes on state abstractions, 2018.

Yiding Jiang, Evan Liu, Benjamin Eysenbach, J Zico Kolter, and Chelsea Finn. Learning options via compression. *Advances in Neural Information Processing Systems*, 35:21184–21199, 2022b.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws. *arXiv preprint arXiv:2410.11820*, 2024c.

Fangkai Jiao, Chengwei Qin, Zhengyuan Liu, Nancy F Chen, and Shafiq Joty. Learning Planning-based Reasoning by Trajectories Collection and Process Reward Synthesizing. *arXiv preprint arXiv:2402.00658*, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=VTF8yNQM66`.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International conference on machine learning*, pages 4880–4889. PMLR, 2020.

Tyler B Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems*, 31, 2018.

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pages 1470–1477, 2011. doi:10.1109/ICRA.2011.5980391.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pages 669–679. PMLR, 2020.

John Maynard Keynes. *A treatise on probability*. Courier Corporation, 1921.

Rawal Khirodkar and Kris M Kitani. Adversarial domain randomization. *arXiv preprint arXiv:1812.00491*, 2018.

Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sDPO: Don't Use Your Data All at Once. *arXiv preprint arXiv:2403.19270*, 2024.

Craig A Knoblock. Learning Abstraction Hierarchies for Problem Solving. In *AAAI*, pages 923–928, 1990.

W Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions. *arXiv preprint arXiv:2206.02231*, 2022a.

W Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions. *arXiv preprint arXiv:2206.02231*, 2022b.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free!, 2019. URL `https://openreview.net/forum?id=r1lgTGL5DE`.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Nishanth Kumar, Willie McClinton, Rohan Chitnis, Tom Silver, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Learning Efficient Abstract Planning Models that Choose What to Predict. In *Conference on Robot Learning*, pages 2070–2095. PMLR, 2023a.

Nishanth Kumar, Willie McClinton, Kathryn Le, , and Tom Silver. Bilevel Planning for Robots: An Illustrated Introduction, 2023b. https://lis.csail.mit.edu/bilevel-planning-for-robots-an-illustrated-introduction.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems*, 31, 2018.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. Hypertree proof search for neural theorem proving. *Advances in Neural Information Processing Systems*, 35:26337–26349, 2022.

Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. In *International conference on machine learning*, pages 2917–2926. PMLR, 2018.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

leanprover-community. mathlib4. `https://github.com/leanprover-community/mathlib4`, 2024. Accessed: 2024-05-15.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pages 1702–1712. PMLR, 2022.

Shane Legg, Marcus Hutter, et al. A collection of definitions of intelligence. *Frontiers in Artificial Intelligence and applications*, 157:17, 2007.

Lucas Lehnert, Sainbayar Sukhbaatar, Paul Mcvay, Michael Rabbat, and Yuandong Tian. Beyond A*: Better Planning with Transformers via Search Dynamics Bootstrapping. *arXiv preprint arXiv:2402.14083*, 2024a.

Lucas Lehnert, Sainbayar Sukhbaatar, Paul Mcvay, Michael Rabbat, and Yuandong Tian. Beyond A*: Better Planning with Transformers via Search Dynamics Bootstrapping. *arXiv preprint arXiv:2402.14083*, 2024b.

Wolfgang Lehrach, Daniel Hennes, Miguel Lazaro-Gredilla, Xinghua Lou, Carter Wendelken, Zun Li, Antoine Dedieu, Jordi Grau-Moya, Marc Lanctot, Atil Iscen, et al. Code World Models for General Game Playing. *arXiv preprint arXiv:2510.04542*, 2025.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1K: A Benchmark for Embodied AI with 1,000 Everyday Activities and Realistic Simulation. In *6th Annual Conference on Robot Learning*, 2022.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023a.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*, 2024.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. Selective reflection-tuning: Student-selected data recycling for llm instruction-tuning. *arXiv preprint arXiv:2402.10110*, 2024a.

Qiyang Li, Yuexiang Zhai, Yi Ma, and Sergey Levine. Understanding the complexity gains of single-task rl with a curriculum. In *International Conference on Machine Learning*, pages 20412–20451. PMLR, 2023b.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline. *arXiv preprint arXiv:2406.11939*, 2024b.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, et al. DeepAgent: A General Reasoning Agent with Scalable Toolsets. *arXiv preprint arXiv:2510.21618*, 2025a.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025b.

Zhaoyu Li, Jialiang Sun, Logan Murphy, Qidong Su, Zenan Li, Xian Zhang, Kaiyu Yang, and Xujie Si. A Survey on Deep Learning for Theorem Proving. *arXiv preprint arXiv:2404.09939*, 2024c.

Zhuofeng Li, Haoxiang Zhang, Seungju Han, Sheng Liu, Jianwen Xie, Yu Zhang, Yejin Choi, James Zou, and Pan Lu. In-the-Flow Agentic System Optimization for Effective Planning and Tool Use. *arXiv preprint arXiv:2510.05592*, 2025c.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023c.

Kaiqu Liang, Zixu Zhang, and Jaime Fernández Fisac. Introspective Planning: Guiding Language-Enabled Agents to Refine Their Own Uncertainty. *arXiv preprint arXiv:2402.06529*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023a.

Chris Yuhao Liu and Liang Zeng. Skywork Reward Model Series. `https://huggingface.co/Skywork`, September 2024. URL `https://huggingface.co/Skywork`.

Huihan Liu, Alice Chen, Yuke Zhu, Adith Swaminathan, Andrey Kolobov, and Ching-An Cheng. Interactive robot learning from verbal correction. *arXiv preprint arXiv:2310.17555*, 2023b.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023c.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023d.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, et al. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*, 2024.

Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. Iterative machine teaching. In *International Conference on Machine Learning*, pages 2149–2158. PMLR, 2017.

Zhihan Liu, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. Reason for future, act for now: A principled framework for autonomous llm agents with provable sample efficiency. *arXiv preprint arXiv:2309.17382*, 2023e.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*, 2024.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Matthew Macfarlane, Edan Toledo, Donal Byrne, Paul Duckworth, and Alexandre Laterre. SPO: Sequential Monte Carlo Policy Optimisation. *Advances in Neural Information Processing Systems*, 37:1019–1057, 2024.

Matthew MacKay, Paul Vicol, Jon Lorraine, David Duvenaud, and Roger Grosse. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. *arXiv preprint arXiv:1903.03088*, 2019.

Maryam Majzoubi, Chicheng Zhang, Rajan Chari, Akshay Krishnamurthy, John Langford, and Aleksandrs Slivkins. Efficient contextual bandits with continuous actions. *Advances in Neural Information Processing Systems*, 33:349–360, 2020.

Jacob Makar-Limanov, Arjun Prakash, Denizalp Goktas, Amy Greenwald, and Nora Ayanian. STA-RLHF: Stackelberg Aligned Reinforcement Learning with Human Feedback. In *Coordination and Cooperation for Multi-Agent Reinforcement Learning Methods Workshop*, 2024. URL `https://openreview.net/forum?id=Jcn8pxwRa9`.

Matthieu Martin, Panayotis Mertikopoulos, Thibaud Rahier, and Houssam Zenati. Nested bandits. In *International Conference on Machine Learning*, pages 15093–15121. PMLR, 2022.

Bogdan Mazoure, Benjamin Eysenbach, Ofir Nachum, and Jonathan Tompson. Contrastive value learning: Implicit models for simple offline rl. In *Conference on Robot Learning*, pages 1257–1267. PMLR, 2023.

David McAllester. Metaphor and mathematics. `https://machinethoughts.wordpress.com/2014/08/03/metaphor-and-mathematics/`, 2014. Accessed: 2024-05-15.

David McAllester. Why we need a new foundation of mathematics. `https://machinethoughts.wordpress.com/2016/01/01/why-we-need-a-new-foundation-of-mathematics/`, 2016. Accessed: 2024-05-15.

David McAllester. Information theoretic co-training. *arXiv preprint arXiv:1802.07572*, 2018.

David McAllester. MathZero, the classification problem, and set-theoretic type theory. *arXiv preprint arXiv:2005.05512*, 2020.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. *arXiv preprint arXiv:2405.14734*, 2024.

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, and Andy R. Terrel. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017. doi:10.7717/peerj-cs.103. URL `https://doi.org/10.7717/peerj-cs.103`.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Mitigating Reward Hacking via Information-Theoretic Reward Modeling. *arXiv preprint arXiv:2402.09345*, 2024.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022.

miniF2F. Automated Theorem Proving on miniF2F Test. `https://paperswithcode.com/sota/automated-theorem-proving-on-minif2f-test`, 2024. Accessed: May 9, 2024.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997. ISBN 978-0070428072.

Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*, pages 625–635. Springer, 2021.

Gabriel Mukobi, Peter Chatain, Su Fong, Robert Windesheim, Gitta Kutyniok, Kush Bhatia, and Silas Alberti. Superhf: Supervised iterative learning from human feedback. *arXiv preprint arXiv:2310.16763*, 2023.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active Preference Learning for Large Language Models. *arXiv preprint arXiv:2402.08114*, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023a.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 2023b.

Rémi Munos et al. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

Kevin Murphy. Reinforcement learning: an overview. *arXiv preprint arXiv:2412.05265*, 2024.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

John F Nash et al. Non-cooperative games. *Princeton University*, 1950.

Peter Neal. The generalised coupon collector problem. *Journal of Applied Probability*, 45(3): 621–629, 2008.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

Allen Nie, Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Importance of Directional Feedback for LLM-based Optimizers. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Allen Nie, Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. The Importance of Directional Feedback for LLM-based Optimizers. *arXiv preprint arXiv:2405.16434*, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pages 16828–16847. PMLR, 2022.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Large language models can do parallel decoding. *arXiv preprint arXiv:2307.15337*, 2023.

Robert Nishihara. Take on DeepSeek-R1, 2025. LinkedIn post.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

OpenAI. OpenAI o3 and o4-mini system card. *OpenAI Technical Report*, 2025.

Laurent Orseau and Levi HS Lelis. Policy-guided heuristic search with guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12382–12390, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Aldo Pacchiano, Aadirupa Saha, and Jonathan Lee. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Michael Painter, Mohamed Baioumy, Nick Hawes, and Bruno Lacerda. Monte Carlo Tree Search with Boltzmann Exploration. *Advances in Neural Information Processing Systems*, 36, 2024.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing Failure Modes of Preference Optimisation with DPO-Positive. *arXiv preprint arXiv:2402.13228*, 2024.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024a.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024b.

Giambattista Parascandolo, Lars Buesing, Josh Merel, Leonard Hasenclever, John Aslanides, Jessica B Hamrick, Nicolas Heess, Alexander Neitz, and Theophane Weber. Divide-and-conquer monte carlo tree search for goal-directed planning. *arXiv preprint arXiv:2004.11410*, 2020.

Aaron Parisi, Yao Zhao, and Noah Fiedel. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024a.

Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. Is Value Learning Really the Main Bottleneck in Offline RL? *arXiv preprint arXiv:2406.09329*, 2024b.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR, 2022.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *CoRR*, abs/1508.00305, 2015. URL `http://arxiv.org/abs/1508.00305`.

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. Curry-dpo: Enhancing alignment using curriculum learning & ranked preferences. *arXiv preprint arXiv:2403.07230*, 2024.

Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. *Advances in Neural Information Processing Systems*, 32, 2019.

Arthur C Pigou. Some aspects of welfare economics. *The American Economic Review*, 41 (3):287–302, 1920.

Gabriel Poesia, Kanishk Gandhi, Eric Zelikman, and Noah Goodman. Certified deductive reasoning with language models. *arXiv preprint arXiv:2306.04031*, 2023.

Gabriel Poesia, David Broman, Nick Haber, and Noah D Goodman. Learning Formal Mathematics From Intrinsic Motivation. *arXiv preprint arXiv:2407.00695*, 2024.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. *arXiv preprint arXiv:2202.01344*, 2022.

Elena Popovici, Anthony Bucci, R Paul Wiegand, and Edwin D De Jong. Coevolutionary principles., 2012.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, et al. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*, 2025.

Ori Press, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. The Entropy Enigma: Success and Failure of Entropy Minimization. *arXiv preprint arXiv:2405.05012*, 2024.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling Laws for Reward Model Overoptimization in Direct Alignment Algorithms. *arXiv preprint arXiv:2406.02900*, 2024a.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q*: Your Language Model is Secretly a Q-Function. *arXiv preprint arXiv:2404.12358*, 2024b.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q*: Your Language Model is Secretly a Q-Function. *arXiv preprint arXiv:2404.12358*, 2024c.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pages 7953–7963. PMLR, 2020a.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pages 7953–7963. PMLR, 2020b.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pages 7953–7963. PMLR, 2020c.

Thomas S Ray. An approach to the synthesis of life. *The philosophy of artificial life*, pages 111–145, 1991.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline Regularised Reinforcement Learning for Large Language Models Alignment. *arXiv preprint arXiv:2405.19107*, 2024.

Nir Rosenfeld and Haifeng Xu. Machine Learning Should Maximize Welfare, Not (Only) Accuracy. *arXiv preprint arXiv:2502.11981*, 2025.

Stephane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct Nash Optimization: Teaching Language Models to Self-Improve with General Preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Advances in neural information processing systems*, 27, 2014.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1): 1–96, 2018.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

Mikayel Samvelyan, Akbir Khan, Michael Dennis, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Roberta Raileanu, and Tim Rocktäschel. MAESTRO: Open-ended environment design for multi-agent reinforcement learning. *arXiv preprint arXiv:2303.03376*, 2023.

Leonard J Savage. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silve. Prioritized Experience Replay. *arXiv preprint arXiv:1511.05952*, 2015.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. *In Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, 1991.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hans-Paul Schwefel. *Evolutionsstrategien für die numerische Optimierung*. Springer, 1977.

Rajat Sen, Alexander Rakhlin, Lexing Ying, Rahul Kidambi, Dean Foster, Daniel N Hill, and Inderjit S Dhillon. Top-k extreme contextual bandits with arm hierarchy. In *International Conference on Machine Learning*, pages 9422–9433. PMLR, 2021.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on Incorrect Synthetic Data Scales the Efficiency of LLM Math Reasoning by Eight-Fold. *arXiv preprint arXiv:2406.14532*, 2024.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *arXiv preprint arXiv:2402.06886*, 2024.

Lucy Xiaoyang Shi, Yunfan Jiang, Jake Grigsby, Linxi Fan, Yuke Zhu, et al. Cross-Episodic Curriculum for Transformer Agents. *arXiv preprint arXiv:2310.08549*, 2023.

Ruizhe Shi, Runlong Zhou, and Simon S Du. The Crucial Role of Samplers in Online Direct Preference Optimization. *arXiv preprint arXiv:2409.19605*, 2024a.

Ruizhe Shi, Runlong Zhou, and Simon S Du. The Crucial Role of Samplers in Online Direct Preference Optimization. *arXiv preprint arXiv:2409.19605*, 2024b.

Shuqing Shi, Xiaobin Wang, Zhiyou Yang, Fan Zhang, and Hong Qu. Double Thompson Sampling in Finite stochastic Games. *arXiv preprint arXiv:2202.10008*, 2022.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Olivier Sigaud, Gianluca Baldassarre, Cedric Colas, Stéphane Doncieux, Richard Duro, Nicolas Perrin-Gilbert, and Vieri-Giuliano Santucci. A definition of open-ended learning problems for goal-conditioned agents. *arXiv preprint arXiv:2311.00344*, 2023.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016a.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David P. Reichert, Neil C. Rabinowitz, André Barreto, and Thomas Degris. The predictron: End-to-end learning and planning. *CoRR*, abs/1612.08810, 2016b. URL `http://arxiv.org/abs/1612.08810`.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017b.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021a.

Tom Silver, Rohan Chitnis, Joshua Tenenbaum, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning symbolic operators for task and motion planning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3182–3189. IEEE, 2021b.

Tom Silver, Ashay Athalye, Joshua B Tenenbaum, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Learning neuro-symbolic skills for bilevel planning. *arXiv preprint arXiv:2206.10680*, 2022.

Tom Silver, Rohan Chitnis, Nishanth Kumar, Willie McClinton, Tomás Lozano-Pérez, Leslie Kaelbling, and Joshua B Tenenbaum. Predicate invention for bilevel planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023a.

Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Pack Kaelbling, and Michael Katz. Generalized Planning in PDDL Domains with Pretrained Large Language Models. *arXiv preprint arXiv:2305.11014*, 2023b.

Herbert A Simon. *The Sciences of the Artificial*. MIT press, 1969.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023a.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023b.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023c.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*, 2025.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proceedings of the 24th annual Conference On Learning Theory*, pages 679–702. JMLR Workshop and Conference Proceedings, 2011.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Yuda Song, Julia Kempe, and Remi Munos. Outcome-based Exploration for LLM Reasoning. *arXiv preprint arXiv:2509.6941*, 2025.

Russell K Standish. Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(02):167–175, 2003.

Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself*, 2017a.

Kenneth O Stanley, Joel Lehman, and Lisa Soros. Open-endedness: The last grand challenge you've never heard of: While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself. *O'Reilly Online*, 2017b.

Jacob Steinhardt. Beyond Bayesians and Frequentists, 2012.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*, 2017.

Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. BBox-Adapter: Lightweight Adapting for Black-Box Large Language Models. *arXiv preprint arXiv:2402.08219*, 2024a.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024b.

Ilya Sutskever. Scaling the right thing matters more now than ever, 2024. LessWrong post.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998a.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998b.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.

Richard S Sutton, Marlos C Machado, G Zacharias Holland, David Szepesvari, Finbarr Timbers, Brian Tanner, and Adam White. Reward-respecting subtasks for model-based reinforcement learning. *Artificial Intelligence*, 324:104001, 2023.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Kunio Takezawa. *Introduction to nonparametric regression*. John Wiley & Sons, 2005.

Hao Tang, Darren Key, and Kevin Ellis. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. *Advances in Neural Information Processing Systems*, 37:70148–70212, 2024a.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024b.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024c.

Deepseek Team, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team, M Reid, N Savinov, D Teplyashin, Lepikhin Dmitry, T Lillicrap, JB Alayrac, R Soricut, A Lazaridou, O Firat, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024a.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024b.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, et al. Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*, 2021.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

tokenbender. avatarl: training language models from scratch with pure reinforcement learning. *github repository*, 2025. URL `https://github.com/tokenbender/avatarl`.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv: Arxiv-2402.10176*, 2024.

Hoang Tran, Chris Glaze, and Braden Hancock. Iterative DPO Alignment. Technical report, Snorkel AI, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition, 2022. URL `https://arxiv.org/abs/2108.00573`.

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. Proximal curriculum for reinforcement learning agents. *arXiv preprint arXiv:2304.12877*, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-based and outcome-based feedback. *NeurIPS MATH-AI Workshop*, 2022.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). *arXiv preprint arXiv:2206.10498*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL `https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024`.

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? Limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*, 2024.

Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Die Handelsblatt-Bibliothek "Klassiker der Nationalökonomie". J. Springer, 1934. URL `https://books.google.com/books?id=wihBAAAAIAAJ`.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024a.

Hao Wang. Proving theorems by pattern recognition—ii. *Bell system technical journal*, 40 (1):1–41, 1961.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. *arXiv preprint arXiv:2406.12845*, 2024b.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. Making large language models better reasoners with alignment. *arXiv preprint arXiv:2309.02144*, 2023a.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. Can Language Models Serve as Text-Based World Simulators? *arXiv preprint arXiv:2406.06485*, 2024c.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. Efficientzero v2: Mastering discrete and continuous control with limited data. *arXiv preprint arXiv:2403.00564*, 2024d.

Tianyu Wang, Weicheng Ye, Dawei Geng, and Cynthia Rudin. Towards practical lipschitz bandits. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, pages 129–138, 2020.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pages 36411–36430. PMLR, 2023b.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022b.

Yuanhao Wang, Qinghua Liu, and Chi Jin. Is RLHF More Difficult than Standard RL? *arXiv preprint arXiv:2306.14111*, 2023d.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024e.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Sean Welleck and Rahul Saha. LLMSTEP: LLM proofstep suggestions in Lean. *arXiv preprint arXiv:2310.18457*, 2023.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. *Advances in Neural Information Processing Systems*, 35:4913–4927, 2022a.

Sean Welleck, Peter West, Jize Cao, and Yejin Choi. Symbolic brittleness in sequence models: on systematic generalization in symbolic mathematics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8629–8637, 2022b.

Lilian Weng. Curriculum for reinforcement learning. *lilianweng.github.io*, Jan 2020. URL `https://lilianweng.github.io/posts/2020-01-29-curriculum-rl/`.

David Wilkins. Using patterns and plans in chess. *Artificial intelligence*, 14(2):165–203, 1980.

Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

Huasen Wu and Xin Liu. Double thompson sampling for dueling bandits. *Advances in neural information processing systems*, 29, 2016.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024a.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024b.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Chenjun Xiao, Ruitong Huang, Jincheng Mei, Dale Schuurmans, and Martin Müller. Maximum entropy monte-carlo planning. *Advances in Neural Information Processing Systems*, 32, 2019.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

Tengyang Xie and Nan Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559. PMLR, 2020.

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory Preference Optimization: Harnessing Implicit Q*-Approximation for Sample-Efficient RLHF. *arXiv preprint arXiv:2405.21046*, 2024.

Huajian Xin, Haiming Wang, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, et al. Lego-prover: Neural theorem proving with growing libraries. *arXiv preprint arXiv:2310.00656*, 2023a.

Huajian Xin, Haiming Wang, Chuanyang Zheng, Lin Li, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, et al. LEGO-Prover: Neural Theorem Proving with Growing Libraries. *arXiv preprint arXiv:2310.00656*, 2023b.

Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*, 2023.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Forty-first International Conference on Machine Learning*, 2024.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023a.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023b.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024a.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing. *arXiv preprint arXiv:2406.08464*, 2024b.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing Systems*, 36, 2024.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models 2021. *arXiv preprint arXiv:2105.13626*, 2021.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent computer interfaces enable software engineering language models, 2024a.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. *arXiv preprint arXiv:2404.01730*, 2024b.

Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *International Conference on Machine Learning*, pages 6984–6994. PMLR, 2019.

Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. Leandojo: Theorem proving with retrieval-augmented language models. *arXiv preprint arXiv:2306.15626*, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. Online Iterative Reinforcement Learning from Human Feedback with General Preference Model. *arXiv preprint arXiv:2402.07314*, 2024a.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34: 25476–25488, 2021a.

Ziyu Ye, Yuxin Chen, and Haitao Zheng. Understanding the effect of bias in deep anomaly detection. *arXiv preprint arXiv:2105.07346*, 2021b.

Ziyu Ye, Rishabh Agarwal, Tianqi Liu, Rishabh Joshi, Sarmishta Velury, Quoc V Le, Qijun Tan, and Yuan Liu. Scalable reinforcement post-training beyond static human prompts: Evolving alignment via asymmetric self-play. *arXiv preprint arXiv:2411.00062*, 2024b.

Ziyu Ye, Jiacheng Chen, Jonathan Light, Yifei Wang, Jiankai Sun, Mac Schwager, Philip Torr, Guohao Li, Yuxin Chen, Kaiyu Yang, et al. Reasoning in reasoning: A hierarchical framework for better and faster neural theorem proving. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024c.

Yuanqing Yu, Zhefan Wang, Weizhi Ma, Shuai Wang, Chuhan Wu, Zhiqiang Guo, and Min Zhang. StepTool: Enhancing Multi-Step Tool Usage in LLMs through Step-Grained Reinforcement Learning. *arXiv preprint arXiv:2410.07745*, 2024a.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36, 2024b.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024a.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024b.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. TextGrad: Automatic" Differentiation" via Text. *arXiv preprint arXiv:2406.07496*, 2024.

Michał Zawalski, Michał Tyrolski, Konrad Czechowski, Tomasz Odrzygozdz, Damian Stachura, Piotr Pikekos, Yuhuai Wu, Lukasz Kucinski, and Piotr Milos. Fast and precise: Adjusting planning horizon with adaptive subgoal search. *arXiv preprint arXiv:2206.00702*, 2022.

Eric Zelikman, Qian Huang, Gabriel Poesia, Noah D Goodman, and Nick Haber. Parsel: A unified natural language framework for algorithmic reasoning. *arXiv preprint arXiv:2212.10561*, 2022a.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022b.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level Direct Preference Optimization. *arXiv preprint arXiv:2404.11999*, 2024.

Guodong Zhang. *Deep Learning Dynamics: From Minimization to Games*. PhD thesis, University of Toronto (Canada), 2023.

Guodong Zhang, Yuanhao Wang, Laurent Lessard, and Roger B Grosse. Near-optimal local convergence of alternating gradient descent-ascent for minimax optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 7659–7679. PMLR, 2022.

Shenao Zhang, Donghan Yu, Hiteshi Sharma, Ziyi Yang, Shuohang Wang, Hany Hassan, and Zhaoran Wang. Self-Exploring Language Models: Active Preference Elicitation for Online Alignment. *arXiv preprint arXiv:2405.19332*, 2024.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo. *arXiv preprint arXiv:2404.17546*, 2024.

Xueliang Zhao, Wenda Li, and Lingpeng Kong. Decomposing the Enigma: Subgoal-based Demonstration Learning for Formal Theorem Proving. *arXiv preprint arXiv:2305.16366*, 2023a.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023b.

Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for offline goal reaching. *arXiv preprint arXiv:2306.03346*, 2023a.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. *arXiv preprint arXiv:2310.20141*, 2023b.

Chuanyang Zheng, Haiming Wang, Enze Xie, Zhengying Liu, Jiankai Sun, Huajian Xin, Jianhao Shen, Zhenguo Li, and Yu Li. Lyra: Orchestrating dual correction in automated theorem proving. *arXiv preprint arXiv:2309.15806*, 2023c.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. Take a step back: evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*, 2023d.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623, 2023e.

Rui Zheng, Hongyi Guo, Zhihan Liu, Xiaoying Zhang, Yuanshun Yao, Xiaojun Xu, Zhaoran Wang, Zhiheng Xi, Tao Gui, Qi Zhang, et al. Toward Optimal LLM Alignments Using Two-Player Games. *arXiv preprint arXiv:2406.10977*, 2024.

Tan Zhi-Xuan, Micah Carroll, Matija Franklin, and Hal Ashton. Beyond Preferences in AI Alignment. *Philosophical Studies*, pages 1–51, 2024.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. Self-discover: Large language models self-compose reasoning structures. *arXiv preprint arXiv:2402.03620*, 2024.

Banghua Zhu, Michael I Jordan, and Jiantao Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.