Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models

Zixiang Chen [* 1] Yihe Deng [* 1] Huizhuo Yuan [* 1] Kaixuan Ji [1] Quanquan Gu [1]

Abstract

Harnessing the power of human-annotated data through Supervised Fine-Tuning (SFT) is pivotal for advancing Large Language Models (LLMs). In this paper, we delve into the prospect of growing a strong LLM out of a weak one without the need for acquiring additional human-annotated data. We propose a new fine-tuning method called Self-Play fIne-tuNing (SPIN), which starts from a supervised fine-tuned model. At the heart of SPIN lies a self-play mechanism, where the LLM refines its capability by playing against instances of itself. More specifically, the LLM generates its own training data from its previous iterations, refining its policy by discerning these self-generated responses from those obtained from human-annotated data. Our method progressively elevates the LLM from a nascent model to a formidable one, unlocking the full potential of human-annotated demonstration data for SFT. Theoretically, we prove that the global optimum to the training objective function of our method is achieved only when the LLM policy aligns with the target data distribution. Empirically, we evaluate our method on several benchmark datasets including the HuggingFace Open LLM Leaderboard, MT-Bench, and datasets from Big-Bench. Our results show that SPIN can significantly improve the LLM's performance across a variety of benchmarks and even outperform models trained through direct preference optimization (DPO) supplemented with extra GPT-4 preference data. This sheds light on the promise of self-play, enabling the achievement of human-level performance in LLMs without the need for expert opponents. Codes are available at https://github.com/uclaml/SPIN.

[*]Equal contribution [1]Department of Computer Science, University of California, Los Angeles, CA 90095, USA. Correspondence to: Quanquan Gu <qgu@cs.ucla.edu>.

1 Introduction

Large Language Models (LLMs) have began a groundbreaking era in artificial general intelligence (AGI), demonstrating extraordinary capabilities across a wide range of domains that require intricate reasoning and specialized knowledge. These models excel in areas such as mathematical reasoning/problem solving (Cobbe et al., 2021; Wei et al., 2022; Lewkowycz et al., 2022), code generation/programming (Chen et al., 2021; Austin et al., 2021; Li et al., 2022), text generation (Bubeck et al., 2023; Anil et al., 2023; Touvron et al., 2023), summarization and creative writing, among others. A significant advancement in LLMs is the post-pre-training alignment with the more desirable behaviors (Mishra et al., 2021; Victor et al., 2022; Chung et al., 2022; Thoppilan et al., 2022), a process often reliant on the costly human-annotated data. Typical alignment methods include Supervised Fine-Tuning (SFT) (Ouyang et al., 2022; Tunstall et al., 2023a) based on human demonstrations, and Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Bai et al., 2022a) based on human preferences.

All the aforementioned alignment methods require a substantial volume of human annotated data. Therefore, there is increasing interest in developing fine-tuning methods that can effectively utilize human data, thereby streamlining the alignment process. This motivates us to study fine-tuning LLMs without the need for additional human-annotated data beyond the fine-tuning dataset. Our study is also related to the broader goal of converting weak models to strong models without the requirement for extra training data, which is of central interest in machine learning that can be traced back to the boosting algorithms (Kearns & Valiant, 1994; Schapire, 1990; Freund, 1995; Freund & Schapire, 1997). The self-training algorithm (Vapnik, 1999; Grandvalet & Bengio, 2004; Lee, 2013) has also been proved to be able to convert weak learners to strong learners in mixture models without the need for additional labeled data (Frei et al., 2022; Kou et al., 2022). However, the pursuit of autonomously enhancing a weak LLM without external guidance is both intriguing and understudied. This raises the following question:

Can we empower a weak LLM to improve itself without acquiring additional human annotated data?

In this paper, we answer this question affirmatively. Inspired by the success of self-play mechanisms (Samuel, 2000) in games, exemplified by AlphaGo Zero (Silver et al., 2017b), AlphaZero (Silver et al., 2017a), with historical roots traced back to TD-Gammon (Tesauro et al., 1995), we propose to convert a weak LLM to a strong one through the lens of self-play, where the model is enhanced by playing against itself without requiring any direct supervision. In particular, we propose a novel fine-tuning method called Self-Play fIne-tuNing (SPIN), which begins from a supervised fine-tuned model. SPIN allows the LLM to engage in self-play, eliminating the need for an expert annotator such as a human or more advanced LLMs like GPT-4. In detail, with the LLM from previous iteration t denoted by $p_{\boldsymbol{\theta}_t}$, we employ it to generate responses $\mathbf{y}'$ to the prompts $\mathbf{x}$ in the human-annotated SFT dataset. The subsequent objective is to find a new LLM $p_{\boldsymbol{\theta}_{t+1}}$, capable of distinguishing the responses $\mathbf{y}'$ generated by $p_{\boldsymbol{\theta}_t}$ from the responses $\mathbf{y}$ generated by humans. This process can be seen as a two-player game: the main player, or the new LLM $p_{\boldsymbol{\theta}_{t+1}}$, seeks to discern between the responses of the opponent player $p_{\boldsymbol{\theta}_t}$ and human-generated responses, while the opponent, or the old LLM $p_{\boldsymbol{\theta}_t}$, generates responses as similar as possible to those in the human-annotated SFT dataset. The new LLM $p_{\boldsymbol{\theta}_{t+1}}$ is obtained by fine-tuning the old one $p_{\boldsymbol{\theta}_t}$ to prefer responses from p_{data} over $p_{\boldsymbol{\theta}_t}$, resulting in a distribution $p_{\boldsymbol{\theta}_{t+1}}$ that is more aligned with p_{data}. In the next iteration, the newly obtained LLM $p_{\boldsymbol{\theta}_{t+1}}$ becomes the opponent for response generation, with the self-play process aiming for the LLM to eventually converge to $p_{\boldsymbol{\theta}^*} = p_{\mathrm{data}}$, so that the strongest possible LLM can no longer differentiate the responses generated by its previous version and those generated by the human.

Interestingly, our method exhibits similarity with the recently introduced direct preference optimization (DPO) method (Rafailov et al., 2023), with the notable distinction being the self-play nature of our method. Consequently, our approach stands out by eliminating the need for extra human preference data, a requirement present in the DPO method. Additionally, the self-play mechanism in our method resembles the idea of generative adversarial networks (GAN) (Goodfellow et al., 2014; Arjovsky et al., 2017), albeit that both the discriminator (main player) and the generator (the opponent) in our method are instances of the same LLM from different iterations. Theoretically, we prove that our method converges when the distribution of the LLM is identical to the target data distribution, i.e., $p_{\boldsymbol{\theta}_t} = p_{\mathrm{data}}$. Our experimental results on `zephyr-7b-sft-full` (Tunstall et al., 2023a), a fine-tuned LLM based on Mistral-7B (Jiang et al., 2023), show that while continued training using SFT on its own SFT

dataset Ultrachat200k (Ding et al., 2023) reaches a performance plateau or even diminished evaluation scores, our method consistently improves `zephyr-7b-sft-full` across successive iterations while leveraging only a **50**k subset of Ultrachat200k dataset. Ultimately, SPIN effectively improves the base model's average score from 58.14 to $\mathbf{63.16}$ on the HuggingFace Open LLM Leaderboard (Beeching et al., 2023) with remarkable $10\%+$ improvement in scores on GSM8k and TruthfulQA, and from 5.94 to $\mathbf{6.78}$ on MT-Bench (Zheng et al., 2023). Notably, SPIN achieves results that are even comparable to models trained on additional 62k preference dataset (Tunstall et al., 2023a) on Open LLM leaderboard and MT-Bench.

Concurrent to our work, Singh et al. (2023) proposed the use of synthetic data with binary feedback in self-training, reducing the reliance on human data. In contrast, our approach eliminates the need for additional binary feedback from humans or an extra reward model thanks to the self-play mechanism. Additionally, Burns et al. (2023) employed a weak LLM model as the guidance to train stronger LLMs in a fashion of weak-to-strong generation. Unlike Burns et al. (2023), which necessitates both a weak supervisor and a strong model, our SPIN operates effectively with a single LLM.

Notation. We use lowercase letters and lowercase bold-face letters to denote scalars and vectors, respectively. We use $[N]$ to denote the index set $\{1, \ldots, N\}$. In the function space, let $\mathcal{F}$ be the function class. The symbol q_{data} designates the target data distribution, while p represents the conditional probability of LLM's response (i.e., LLM policy).

2 Related Work

Self-Play. Self-play (Samuel, 1959; Tesauro et al., 1995), where the algorithm learns by playing against itself, has gained notable attention due to its effectiveness in multi-agent reinforcement learning (MARL). This method involves agents engaging in interactions with copies of themselves, enabling an increasing level of challenge and complexity within the learning environment. A fundamental work in the field of self-play is AlphaGo Zero (Silver et al., 2017b), which demonstrated exceptional performance against human players using a self-play learning scheme. Subsequent research has expanded upon the concept of self-play, exploring various adaptations and implementations (Anthony et al., 2017; Lanctot et al., 2017; Bansal et al., 2018; Hernandez-Leal et al., 2018; Muller et al., 2019; Vinyals et al., 2019). Our method takes the self-play approach akin to AlphaGo Zero, which can convert a weak model to a strong one without additional human-annotated data. While the effectiveness of self-play in MARL is well-established, to our knowledge, our work is the first to apply

this approach to the enhancement of LLMs.

Synthetic Data for LLMs. In the context of supervised fine-tuning (SFT) of LLMs, human-crafted data has proven to be a remarkably effective source that enhances the performance of LLMs on tasks such as code generation (Roziere et al., 2023; Yang et al., 2023) and mathematical reasoning (Yuan et al., 2023; Luo et al., 2023). While human data typically exhibits high quality, acquiring sufficient amount of such data poses a challenge in cost. In light of this consideration, the use of synthetic data has become increasingly popular and considered as a proxy for human data. This approach primarily leverages advanced LLMs such as the GPT series (Radford et al., 2019; Brown et al., 2020; OpenAI, 2023) as the guidance to generate high-quality data (Josifoski et al., 2023; Taori et al., 2023; Chiang et al., 2023; Li et al., 2023). Recent research has also highlighted the rephrasing capability of LLMs in prompting for better LLM response (Deng et al., 2023; Prasad et al., 2023) as well as augmenting synthetic data for more effective SFT (Yu et al., 2023; Liu et al., 2023). In contrast to prior studies that utilized more advanced models for synthetic data generation when pre-training or fine-tuning a target model, our approach directly generates synthetic data from the target model itself.

3 Problem Setting and Preliminaries

We consider a Large Language Model (LLM) parameterized by $\boldsymbol{\theta}$ and denoted by $p_{\boldsymbol{\theta}}$. The model takes as input a sequence $\mathbf{x} = [x_1, \ldots, x_n]$, commonly referred to as the prompt, to generate the corresponding response $\mathbf{y} = [y_1, \ldots, y_m]$. The response $\mathbf{y}$ is therefore considered as a sample from the conditional probability distribution $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$. In LLMs, x_i and y_j represent individual tokens from a predetermined vocabulary within the sequences $\mathbf{x}$ and $\mathbf{y}$, respectively. The auto-regressive model $p_{\boldsymbol{\theta}}$ generates tokens sequentially for a given position, leveraging only the sequence of previously generated tokens. This model therefore constitutes a Markov process, where the conditional probability distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ can be expressed through a decomposition as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} p_{\boldsymbol{\theta}}(y_j|\mathbf{x}, \mathbf{y}_{<j}),$$

where $\mathbf{y}_{<1}$ is null and $\mathbf{y}_{<j} = [y_1, \ldots, y_{j-1}]$ for $j = 2, \ldots, m$. In the following, we review two major fine-tuning methods for LLMs: supervised fine-tuning and reinforcement learning (RL) fine-tuning.

3.1 Supervised Fine-Tuning

Supervised fine-tuning (SFT) is employed to tailor a pre-trained LLM to specific downstream tasks, leveraging relatively smaller dataset of labeled examples in comparison

to the large-scale pre-training data (Ouyang et al., 2022; Yu et al., 2023). In this context, we consider a specific task where the prompts, denoted by $\mathbf{x}$, are derived from a specified distribution $q(\cdot)$. The notation $p_{\mathrm{data}}(\cdot|\mathbf{x})$ then represents the probability distribution of the associated high-quality responses $\mathbf{y}$ from the training data. Consequently, SFT involves training the LLM to minimize the following negative log-likelihood loss associated with these distributions,

$$L_{\mathrm{SFT}}(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x}\sim q(\cdot), \mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x})}\Big[\log p_{\boldsymbol{\theta}}\big(\mathbf{y}|\mathbf{x}\big)\Big]. \quad (3.1)$$

It should be noted that excluding $\mathbf{x} \sim q(\cdot)$ from the expectation term yields the typical cross-entropy loss, expressed as $-\mathbb{E}_{\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})]$. $L_{\mathrm{SFT}}(\boldsymbol{\theta})$ attains its minimum when the model's predictive distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ aligns perfectly with the distribution of the labeled high-quality responses $p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$.

Consequently, the LLM after SFT is anticipated to generate responses that closely resemble those from $p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$. This procedure is therefore expected to significantly enhance the model's performance in generating appropriate responses for a specific task.

3.2 RL Fine-Tuning

RL fine-tuning (Christiano et al., 2017; Bai et al., 2022a; Gao et al., 2023a) offers another method for enhancing the specific capabilities of general-purpose pre-trained models. Typically, RL fine-tuning is employed subsequent to SFT to achieve improved alignment for LLMs (Tunstall et al., 2023a).

For a given sequence pair $(\mathbf{x}, \mathbf{y})$, RL fine-tuning necessitates a deterministic reward function $r(\mathbf{x}, \mathbf{y})$. The higher the reward $r(\mathbf{x}, \mathbf{y})$, the better the response $\mathbf{y}$ is to the given prompt $\mathbf{x}$. The objective of the RL fine-tuning process is then to maximize the following objective function:

$$\begin{aligned} L_{\mathrm{RL}}(\boldsymbol{\theta}) = {} & \mathbb{E}_{\mathbf{x}\sim q(\cdot), \mathbf{y}\sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}[r(\mathbf{x}, \mathbf{y})] \\ & - \lambda \mathbb{E}_{\mathbf{x}\sim q(\cdot)}\mathrm{KL}\big(p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})||p_{\mathrm{ref}}(\cdot|\mathbf{x})\big), \end{aligned}$$

where the Kullback-Leibler (KL) regularization enforces the new model $p_{\boldsymbol{\theta}}$ to be close to the reference model p_{ref}, and $\lambda > 0$ is the regularization parameter to control the deviation of the new model $p_{\boldsymbol{\theta}}$ from the reference model p_{ref}. In practice, the reference model p_{ref} is often initialized as the supervised fine-tuned model. The inclusion of KL regularization is vital for preventing excessive deviation from the reference model, which in turn reduces the risk of mode collapse.

Meanwhile, the primary challenge in RL fine-tuning lies in finding a good reward function. Typically, this function requires training on a preference dataset. The compilation of such a dataset demands significant resources, often

involving comprehensive evaluations either by human annotators, i.e., reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022a) or strong AI agents, i.e., reinforcement learning from AI feedback (RLAIF) (Bai et al., 2022b).

4 Method

In this section, we introduce a new fine-tuning method for enhancing the performance of LLMs without relying on additional human or AI feedback. Consider a high-quality supervised fine-tuning (SFT) dataset $S_{\text{SFT}} = \{(\mathbf{x}, \mathbf{y})\}_{i=1}^{n}$, which are sampled from the marginal distribution $q(\mathbf{x})$ and $p_{\text{data}}(\mathbf{y}|\mathbf{x})$. Given a supervised fine-tuned LLM $p_{\boldsymbol{\theta}_0}$, further application of the SFT approach in (3.1) with S_{SFT} will be ineffective and potentially lead to worse performance. In addition, without human and/or AI feedback, it becomes infeasible to acquire a preference dataset for RL fine-tuning (e.g., RLHF and RLAIF). This hinders the application of RL fine-tuning techniques.

We evaluate $p_{\boldsymbol{\theta}_0}$ against S_{SFT}, where $p_{\boldsymbol{\theta}_0}$ is the LLM achieved by SFT using (3.1). We notice a persistent quality gap between the groundtruth response $\mathbf{y}$ from S_{SFT} and the LLM-generated response $\mathbf{y}' \sim p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ (refer to Figure 1). This disparity indicates that there is still room for improvement over $p_{\boldsymbol{\theta}_0}$. Therefore, we propose to use the synthetic data generated by the LLM to enhance LLM's performance starting from $p_{\boldsymbol{\theta}_0}$ iteratively.

4.1 Self-Play Fine-Tuning (SPIN)

Let us consider a two-player game, where the main player's objective is to distinguish the responses generated by the LLM and those generated by the human. Meanwhile, the opponent's role is to generate responses that are indistinguishable from the human's responses. The core of our method is the self-play mechanism, where both the main player and the opponent are the same LLM, but from different iterations. More specifically, the opponent is the old LLM from the previous iteration, and the main player is the new LLM to be learned in the current iteration.

In iteration $t+1$, the opponent is the LLM from the previous iteration, denoted by $p_{\boldsymbol{\theta}_t}$, which generates responses $\mathbf{y}'$ for those prompts $\mathbf{x}$ in the SFT dataset according to $p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$. Our method, therefore, consists of the following two steps at iteration $t+1$: (1) training the main player, and (2) updating the opponent player.

Training the Main Player. We begin with illustrating how we expect a main player is trained to distinguish LLM responses from human responses. Motivated by integral probability metric (IPM) (Müller, 1997), we formulate our objective function such that the main player f_{t+1} maximizes the expected value gap between the target data distribution

p_{data} and the opponent player's distribution $p_{\boldsymbol{\theta}_t}$:

$$f_{t+1} = \underset{f \in \mathcal{F}_t}{\arg\max} \, \mathbb{E}\big[f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x}, \mathbf{y}')\big], \quad (4.1)$$

where the expectation is computed over the distributions $\mathbf{x} \sim q(\cdot), \mathbf{y} \sim p_{\text{data}}(\cdot|\mathbf{x}), \mathbf{y}' \sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$, and $\mathcal{F}_t$ is a sequence of highly expressive function classes that we will determine in later deduction. The subscript t in $\mathcal{F}_t$ is due to that the function class is dependent on $p_{\boldsymbol{\theta}_t}$. Given such a f_{t+1} and a response sequence $\mathbf{y}$ to the prompt $\mathbf{x}$, the value of $f_{t+1}(\mathbf{x}, \mathbf{y})$ reflects the main player's degree of belief that $\mathbf{y}$ originates from p_{data} rather than $p_{\boldsymbol{\theta}_t}$. Ideally, the main player f_{t+1} should yield a high value when $\mathbf{y} \sim p_{\text{data}}(\cdot|\mathbf{x})$ and a low value when $\mathbf{y}' \sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$, where $p_{\boldsymbol{\theta}_t}$ is the opponent's distribution. Instead of solving (4.1), we can also solve the following more general optimization problem,

$$f_{t+1} = \underset{f \in \mathcal{F}_t}{\arg\min} \, \mathbb{E}\big[\ell\big(f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x}, \mathbf{y}')\big)\big], \quad (4.2)$$

where the expectation is computed over the distribution $\mathbf{x} \sim q(\cdot), \mathbf{y} \sim p_{\text{data}}(\cdot|\mathbf{x}), y' \sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$, and $\ell(\cdot)$ is a loss function that is both monotonically decreasing and convex. For example, a linear loss function $\ell(t) = -t$ reduces (4.2) to the minimization version of (4.1). However, the use of a linear loss function results in an unbounded objective value, which, during continuous training, leads to a negative infinite value of $f(\mathbf{x}, \mathbf{y}')$ on the opponent player's responses. Therefore, in our work, we choose the logistic loss function $\ell(t) := \log(1+\exp(-t))$ for its non-negativity, smoothness, and exponentially decaying tail as $t \to \infty$. Such a choice of loss function aids in preventing the excessive growth in the absolute value of f.

Updating the Opponent Player. Previously we have discussed the training of f_{t+1} given the opponent player's distribution $p_{\boldsymbol{\theta}_t}$. Now suppose we have optimized our main player f_{t+1} that can distinguish p_{data} from $p_{\boldsymbol{\theta}_t}$, within a certain function class $\mathcal{F}_t$, we elaborate how we get parameter $\boldsymbol{\theta}_{t+1}$ of the opponent player. Specifically, when presented with two responses $\mathbf{y}$ and $\mathbf{y}'$ to the same prompt $\mathbf{x}$, f_{t+1} assesses the values $f_{t+1}(\mathbf{x}, \mathbf{y})$ and $f_{t+1}(\mathbf{x}, \mathbf{y}')$. It then infers that the response with the higher value is from the real data distribution p_{data} and the response with lower value is attributed to the LLM $p_{\boldsymbol{\theta}_t}$. Subsequently, the objective of the opponent player is to find a better LLM that generates responses indistinguishable from p_{data} for the main player. This is achieved by maximizing the expected value $\mathbb{E}_{\mathbf{x} \sim q(\cdot), \mathbf{y} \sim p(\cdot|\mathbf{x})}[f_{t+1}(\mathbf{x}, \mathbf{y})]$. In addition, to prevent excessive deviation of $p_{\boldsymbol{\theta}_{t+1}}$ from $p_{\boldsymbol{\theta}_t}$ and stabilize the self-play, we incorporate a Kullback-Leibler (KL) regularization term. Putting these together gives rise to the following optimization problem:

$$\underset{p}{\arg\max} \, \mathbb{E}_{\mathbf{x} \sim q(\cdot), \mathbf{y} \sim p(\cdot|\mathbf{x})}[f_{t+1}(\mathbf{x}, \mathbf{y})]$$
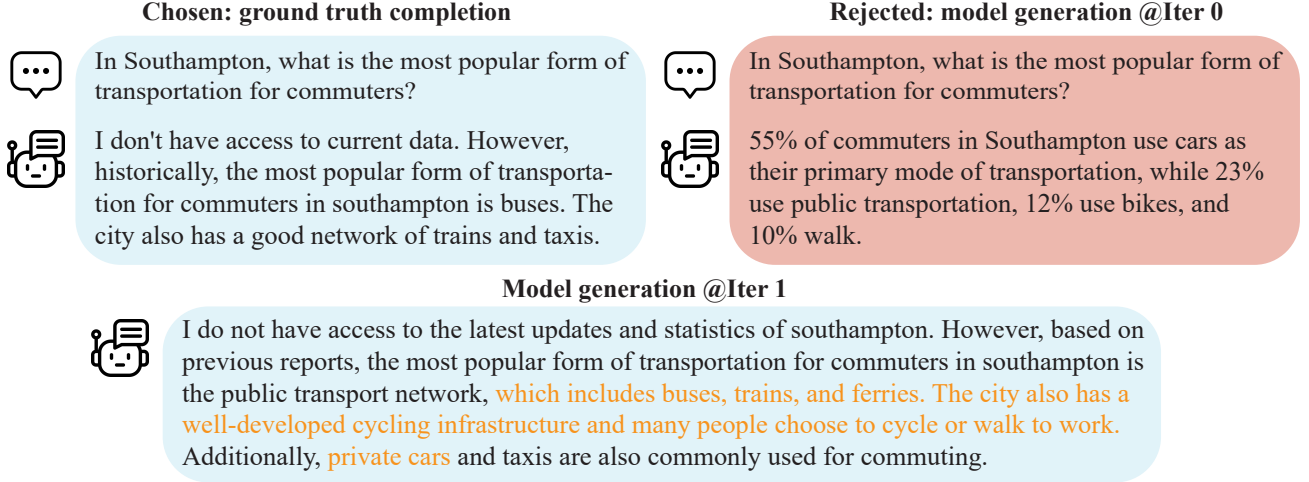
Chosen: ground truth completion

[...] In Southampton, what is the most popular form of transportation for commuters?

[🤖] I don't have access to current data. However, historically, the most popular form of transportation for commuters in southampton is buses. The city also has a good network of trains and taxis.

Rejected: model generation @Iter 0

[...] In Southampton, what is the most popular form of transportation for commuters?

[🤖] 55% of commuters in Southampton use cars as their primary mode of transportation, while 23% use public transportation, 12% use bikes, and 10% walk.

Model generation @Iter 1

[🤖] I do not have access to the latest updates and statistics of southampton. However, based on previous reports, the most popular form of transportation for commuters in southampton is the public transport network, which includes buses, trains, and ferries. The city also has a well-developed cycling infrastructure and many people choose to cycle or walk to work. Additionally, private cars and taxis are also commonly used for commuting.

Figure 1. Example of ground truth completion compared to the fine-tuned model generation at iteration 0 and 1. We can observe that the model generation at iteration 0, although fluent, incorrectly quantifies transportation preferences with specific percentages that are potentially hallucinations. The model generation at iteration 1 provides a qualitative summary of the transportation forms at Southampton without specific percentage, aligning more closely with the ground truth while adding more details.

$$- \lambda \mathbb{E}_{\mathbf{x} \sim q(\cdot)} \text{KL}\big(p(\cdot|\mathbf{x})||p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})\big), \qquad (4.3)$$

where $\lambda > 0$ is the regularization parameter. Notably, (4.3) has a closed-form solution $\widehat{p}(\cdot|\mathbf{x})$:

$$\widehat{p}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \exp\big(\lambda^{-1} f_{t+1}(\mathbf{x}, \mathbf{y})\big). \qquad (4.4)$$

It is worth noting that $\widehat{p}(\cdot|\mathbf{x})$ is not guaranteed to be belong to the LLM space $\{p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})|\boldsymbol{\theta} \in \boldsymbol{\Theta}\}$. Since we hope that the closed-form solution $\widehat{p}$ in the probability space can be realized by an LLM with parameter $\boldsymbol{\theta}$, i.e., $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \widehat{p}(\mathbf{y}|\mathbf{x})$, solving for $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \exp\big(\lambda^{-1} f_{t+1}(\mathbf{x}, \mathbf{y})\big)$ gives $f_{t+1}(\mathbf{x}, \mathbf{y}) = \lambda \cdot \log \frac{p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}$. This suggests the following function class $\mathcal{F}_t$ for f_{t+1}:

$$\mathcal{F}_t = \left\{ \lambda \cdot \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} \bigg| \boldsymbol{\theta} \in \boldsymbol{\Theta} \right\}, \qquad (4.5)$$

where $\boldsymbol{\Theta}$ is the parameter space of LLMs being considered. Given the choice of $\mathcal{F}_t$ in (4.5), optimizing (4.2) gives f_{t+1} parameterized by $\boldsymbol{\theta}_{t+1}$ in the following form:

$$f_{t+1}(\mathbf{x}, \mathbf{y}) = \lambda \cdot \log \frac{p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}. \qquad (4.6)$$

Substituting (4.6) into (4.4) yields $\widehat{p}(\mathbf{y}|\mathbf{x}) = p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})$. In other words, $\boldsymbol{\theta}_{t+1}$ learned from (4.2) is exactly the LLM parameter for our ideal opponent selection.

End-to-end Training Objective. We integrate the previously discussed two steps into a single end-to-end training objective with an update rule of $\boldsymbol{\theta}_{t+1}$. Specifically, plugging (4.5) into (4.2) arrives at the update rule $\boldsymbol{\theta}_{t+1} =$

$\text{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L_{\text{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$, where L_{SPIN} is the training objective defined as follows

$$L_{\text{SPIN}} = \mathbb{E}\left[\ell\left(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right], \tag{4.7}$$

where the expectation is computed over the distribution $\mathbf{x} \sim q(\cdot), \mathbf{y} \sim p_{\text{data}}(\cdot|\mathbf{x}), \mathbf{y}' \sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$. We summarize the iterative self-play process of our method SPIN as follows,

$$\cdots \rightarrow \underbrace{p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}_{\text{Opponent Player at } t} \rightarrow \underbrace{\lambda \cdot \log \frac{p_{\boldsymbol{\theta}_{t+1}}(\cdot|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}}_{\text{Main Player at } t+1}$$

$$\rightarrow \underbrace{p_{\boldsymbol{\theta}_{t+1}}(\cdot|\mathbf{x})}_{\text{Opponent Player at } t+1} \rightarrow \cdots$$

Namely, the opponent player chosen from the previous iteration t is employed to train the main player at iteration $t+1$, resulting in the LLM parameterized by $\boldsymbol{\theta}_{t+1}$. Then we determine the next opponent player at iteration $t+1$ by directly copying the LLM parameter $\boldsymbol{\theta}_{t+1}$, which is then used in training the main player at iteration $t+2$. The detailed algorithm is presented in Algorithm 1.

4.2 Comparison between SPIN and DPO

In Section 4.1, we propose Self-Play Fine-Tuning (SPIN) with an end-to-end training objective (4.7) for each iteration. (4.7) bears resemblance to direct preference optimization (DPO) (Rafailov et al., 2023) for RL fine-tuning. However, SPIN and DPO are fundamentally different. DPO is based on the Bradley-Terry (BT) model: $p(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x}) = \frac{\exp(r^*(\mathbf{x}, \mathbf{y}_1))}{\exp(r^*(\mathbf{x}, \mathbf{y}_1)) + \exp(r^*(\mathbf{x}, \mathbf{y}_2))}$, and maximizes the

---

Algorithm 1 Self-Play Fine-Tuning (SPIN)

Input: $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in [N]}$: SFT Dataset, $p_{\boldsymbol{\theta}_0}$: LLM with parameter $\boldsymbol{\theta}_0$, T: Number of iterations.
for $t = 0, \ldots, T - 1$ **do**
   for $i = 1, \ldots N$ **do**
      Generate synthetic data $\mathbf{y}'_i \sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}_i)$.
   end for
   Update $\boldsymbol{\theta}_{t+1} = \mathrm{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sum_{i \in [N]} \ell \Big(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i)}{p_{\boldsymbol{\theta}_t}(\mathbf{y}_i|\mathbf{x}_i)} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}'_i|\mathbf{x}_i)}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'_i|\mathbf{x}_i)} \Big)$.
end for
Output: $\boldsymbol{\theta}_T$.

---

log-likelihood of $p(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x})$ by direct policy optimization without explicit reward estimation. In contrast, SPIN relies on maximizing the IPM to compete with an increasingly stronger version of itself. More detailed comparisons are highlighted as follows:

1. DPO does not inherently lead to iterative training. More specifically, DPO aims to match the preference probability $p(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x})$ induced from its reward model with the data distribution $p_{\mathrm{data}}(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x})$ in a single iteration. On the contrary, SPIN's self-play mechanism naturally leads to an iterative training procedure. SPIN iteratively refines its generation distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ to match the target distribution $p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$ across iterations.
2. SPIN only requires the SFT dataset, represented by pairs $(\mathbf{x}, \mathbf{y})$. In contrast, DPO necessitates a preference dataset, represented by $(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l)$, where $\mathbf{y}_w$ and $\mathbf{y}_l$ denote the winner (chosen) and loser (rejected) responses, respectively. Moreover, SPIN can be applied between SFT and RL fine-tuning.
3. In SPIN, we can choose different loss functions ℓ which only need to be convex and decreasing (detailed later in Theorem 5.2), which includes correlation loss, hinge loss, and logistic loss. Only when ℓ is chosen as the logistic loss would the training objective of SPIN become similar to that of DPO.

Recently, Xu et al. (2023) proposed to use iterative preference optimization with the Pairwise Cringe Loss (PCO), and generalized DPO to iterative DPO. Concurrent to our work, Yuan et al. (2024) further proposed a framework named "self-rewarding language models", which leverages the LLM itself as the reward model to provide the preference feedback, and employs iterative DPO to train the LLM. Compared with Xu et al. (2023); Yuan et al. (2024), SPIN's self-assessment is implicit, as no intermediate reward or preference feedback is required.

5 Theoretical Analysis

In this section, we provide a theoretical analysis for Algorithm 1 in Section 4. Under monotonicity and convexity

assumption of the objective function ℓ, we show that the global optimum is obtained if and only if parameter $\boldsymbol{\theta}_t$ generates data distribution. We summarize our assumptions as follows:

Assumption 5.1. The loss function $\ell(t) : \mathbb{R} \to \mathbb{R}$ is monotonically decreasing, i.e., $\forall t, \ell'(t) \leq 0$ and satisfies $\ell'(0) < 0$. In addition, $\ell(t)$ is a convex function.

Assumption 5.1 holds for a wide range of loss functions commonly used in machine learning, including correlation loss $\ell(t) = 1 - t$, hinge loss $\ell(t) = \max(0, 1 - t)$, exponential loss $\ell(t) = \exp(-t)$ and logistic loss $\ell(t) = \log(1 + \exp(-t))$. Under Assumptions 5.1, we present the following theorem, which is pivotal in understanding the optimization dynamics of our method.

Theorem 5.2. Under Assumption 5.1, suppose there exists $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = p_{\mathrm{data}}(\cdot|\mathbf{x})$, then we have that

- (Sufficiency) If $p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}) = p_{\mathrm{data}}(\cdot|\mathbf{x})$, then $\boldsymbol{\theta}_t$ is the global minimum of (4.7) for any $\lambda \geq 0$.
- (Necessity) If $p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}) \neq p_{\mathrm{data}}(\cdot|\mathbf{x})$, there exists an appropriately chosen λ, such that $\boldsymbol{\theta}_t$ is not the global minimum of (4.7).

Remark 5.3. Theorem 5.2 suggests that under certain conditions, the optimization process of our method naturally stops at the point $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = p_{\mathrm{data}}(\cdot|\mathbf{x})$, implying the effectiveness of our approach in aligning the LLM's distribution with the target data distribution. Moreover, Theorem 5.2 also indicates that the optimization process only stops when the global optimality is achieved, i.e., the LLM's distribution aligns with the target data distribution.

For the logistic loss function $\ell(t) = \log(1 + \exp(-t))$, the following theorem gives a more precise characterization of the opponent player, enabling a better understanding of SPIN.

Theorem 5.4. Consider the choice of logistic loss $\ell(t) = \log(1 + \exp(-t))$ in SPIN. Suppose that $p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \big)^{1/\lambda}$ lies in the LLM space $\{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})|\boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ and $\boldsymbol{\theta}_{t+1}$ is global minimum of $L_{\mathrm{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$, then the opponent player at iteration $t + 1$ satisfies

$$p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \big)^{1/\lambda}.$$

Remark 5.5. According to Theorem 5.4, the model update from $p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})$ to $p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})$ tends to increase the probability $p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})$ when $p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})$ is less than $p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$, and decrease it when $p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})$ is greater than $p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$. Thus, Theorem 5.4 further confirms that our method's optimization process naturally converges to the point where $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ equals $p_{\mathrm{data}}(\cdot|\mathbf{x})$. The update of the opponent player is controlled by $\big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}) \big)^{1/\lambda}$, which is regulated by

the factor $1/\lambda$. A smaller λ results in a larger change of the opponent player, while a larger λ leads to a smaller change. Therefore, as $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ approaches $p_{\text{data}}(\cdot|\mathbf{x})$, increasing λ enhances the stability of LLM training. This observation aligns with (4.3), where λ is the regularization parameter of the KL regularization that is employed to control the deviation of the opponent player.

6 Experiments

This section provides a detailed empirical analysis of SPIN. Our findings highlight several key points: (1) SPIN markedly enhances model performance across a wide range of evaluation benchmarks by breaking the limit of SFT; (2) even without introducing new human annotated data, SPIN at iteration 0 achieves performance on par to DPO training that utilizes even more data; (3) iterative training is a necessary component in SPIN as it breaks the limit of multi-epoch training.

6.1 Experiment Setup

Model and Datasets. In this study, we adopt zephyr-7b-sft-full as our base model. This model derives from the pre-trained Mistral-7B (Jiang et al., 2023) and has been further fine-tuned on the SFT dataset Ultrachat200k[1] by HuggingFace. Ultrachat200k represents a high-quality 200k subset of the larger UltraChat (Ding et al., 2023) corpus, which comprises approximately 1.4M dialogues produced using OpenAI's Turbo APIs. From Ultra-Chat200k, We randomly sample 50k prompts and use the base model to generate the synthetic responses. We subsequently follow the optimization method described in Section 4.1 for further training. In multiple iterations, we leverage the synthetic data from the most recent iteration and add to the newly generated synthetic data, therefore resulting in a synthetic dataset size of 50k at iteration 0 and 100k at iteration 1, 2 and 3. At each iteration, we train our model for 2 epochs.

Evaluation. We employed the widely used Huggingface Open LLM Leaderboard (Beeching et al., 2023) as our evaluation benchmark, using the same Language Model Evaluation Harness library (Gao et al., 2023b). This leaderboard encompasses 6 different datasets, each focusing on a a specific capability of LLMs. Collectively, these datasets provide a thorough assessment framework, evaluating LLMs on commonsense reasoning (Arc (Clark et al., 2018), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021)), multi-task language understanding (MMLU(Hendrycks et al., 2020)), human falsehood mimic (TruthfulQA (Lin et al., 2021)) and math problem solving (GSM8k (Cobbe et al., 2021)). We leave further implemen-

tation details to Appendix B with detailed evaluation setting adopted by both the leaderboard and our experiments.

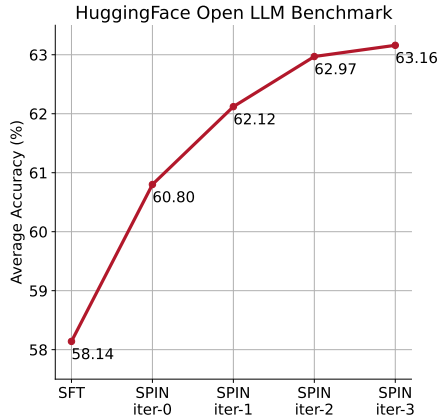## 6.2 SPIN Effectively Improves Benchmark Performance

Figure 2. The average score of SPIN at different iterations on the HuggingFace Open LLM leaderboard datasets. For "SFT", we report the performance of our base model zephyr-7b-sft-full, which has been fine-tuned on the same dataset we use to generate synthetic data.

In Figure 2, we demonstrate the effectiveness of SPIN using HuggingFace Open LLM Leaderboard as the evaluation. In Figure 3, we compare the performance of our fine-tuned model by SPIN after iterations 0 to 3 with the base model zephyr-7b-sft-full on each task included in the leaderboard. Detailed performances are presented in Table 4 in Appendix B. We can observe that SPIN exhibits remarkable effectiveness in improving the model's performance by further leveraging the SFT dataset, on which the base model has already been fully fine-tuned. At iteration 0, where model responses are generated from zephyr-7b-sft-full, we observe an overall improvement of 2.66% on the average score. The improvement is particularly significant on the TruthfulQA and GSM8k benchmarks, with improvement exceeding 5% and 10% respectively. At iteration 1, we employ the LLM model from iteration 0 to generate new responses for SPIN, adhering to the procedure outlined in Algorithm 1. This iteration yields further enhancements of 1.32% on average, and especially significant on the Arc Challenge and TruthfulQA benchmarks. Subsequent iterations continue this trend of incremental improvement across various tasks. Meanwhile, the improvement at iteration $t + 1$ is naturally smaller than that at iteration t. As the iterative training progresses, the degree of improvement gradually approaches zero, suggesting that the model has reached a limiting point in the last iteration.

Comparison with DPO. zephyr-7b-beta is a model derived from zephyr-7b-sft-full, trained with DPO

---

[1] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

on approximately 62k preference data. This data, the Ultra-Feedback Binarized dataset (Cui et al., 2023)[2], comprises both chosen and rejected completions evaluated by GPT-4. We note that, DPO requires either human input or advanced language model feedback to determine the preference, making data generation a rather expensive procedure. In contrast, our SPIN only requires the initial model itself. Moreover, unlike DPO which requires new data source, our method exclusively leverages the existing SFT dataset. In Figure 3, we show the performance comparison of SPIN at iterations 0 and 1 (employing 50k SFT data) with DPO training, from the same SFT checkpoint. We can observe that, while DPO leverages more data from new sources, SPIN based on the existing SFT data can already achieve comparable average performance to DPO training at iteration 0. From iteration 1, SPIN even surpasses the performance of DPO on the leaderboard benchmark.

6.3 Ablation Studies

In this subsection, we examine the effect of synthetic dataset size and training epochs within an iteration. Our analysis demonstrates the effectiveness of the synthetic data used by SPIN compared to the SFT data, as well as the necessity of iterative training in SPIN. In Appendix B, we present assessment of SPIN on additional benchmark tasks.

Training Size. We investigate the effect of varying training data size on the performance of SPIN. In Figure 5, we demonstrate the effect of training size for SPIN during iteration 0 and additionally compare with SFT with the full original dataset. Specifically, for the SFT baseline, we fully fine-tune Mistral-7B on Ultrachat200k for three epochs and report first epoch performance as the starting point (with x-axis 0) in the figure for SFT. For SPIN, we report the zephyr-7b-sft-full checkpoint as the starting point, which has also been fine-tuned on Ultrachat200k for one epoch. We select the training size of SPIN at iteration 0 to be 14k, 26k, and 50k and generate the data accordingly, ensuring that the larger dataset encompasses the smaller dataset. The performance of SPIN was then evaluated after 1 epoch of self-play fine-tuning for each training size. We can observe that, while SPIN results in notable improvement with increasing training sizes, SFT on further epochs 2 and 3 fails to yield more than 1% improvement. Additional results are deferred to Appendix B.

Iterative Training v.s. Training for More Epochs. We further study the training within iteration 0 and compare with the performance achieved in iteration 1, particularly contrasting the test performance obtained from extended training duration with that from next iteration. Figure 4 depicts the performance trajectory of the model trained using

SPIN over multiple epochs at iteration 0. It is evident that the most substantial improvement occurs during the first two epochs, followed by only modest gains in subsequent epochs. Notably, SPIN exhibits robustness and stability; extending the training duration does not diminish performance but rather maintains a rather consistent level. Nevertheless, the observation suggests an inherent limitation to the performance achievable within a single iteration, thereby underscoring the necessity for iterative training. As shown by the test performance achieved at iteration 1 in the figures, extending the training in iteration 0 fails to reach the performance comparable to iteration 1.

7 Conclusion and Discussion

This paper introduces a novel fine-tuning method SPIN, to convert a weak LLM to a strong LLM by unleashing the full power of human-annotated data. Central to this method is a self-play mechanism, wherein a main player (the LLM) is fine-tuned to differentiate the responses of opponent player (the LLM from previous iteration) from the target data distribution, and the LLM is iteratively aligned with the target data distribution. Therefore, SPIN facilitates the LLM's iterative self-evaluation and enhancement through self-play. In comparison to supervised fine-tuning and RL fine-tuning methods, SPIN enables the LLM to self-improve without additional human data or feedback from stronger LLMs. Empirical results demonstrate that SPIN significantly enhances LLM performance across diverse benchmarks, even outperforming models trained with additional human data or AI feedback.

Limitation and Future Work. Our theoretical results demonstrate that the optimization process of SPIN converges if and only if the LLM's distribution aligns with p_{data}. Therefore, our study focuses on a fixed target data distribution generated by humans, which inherently imposes a ceiling on the performance of fine-tuned LLM. Exploring the dynamically changing target data distribution is an important direction to overcome this limitation and elevate the LLM's performance beyond this ceiling or even to a super-human level. Moreover, considering the resource demands of synthetic data generation, another promising avenue for further exploration is to reduce the volume of required synthetic data.

Acknowledgement

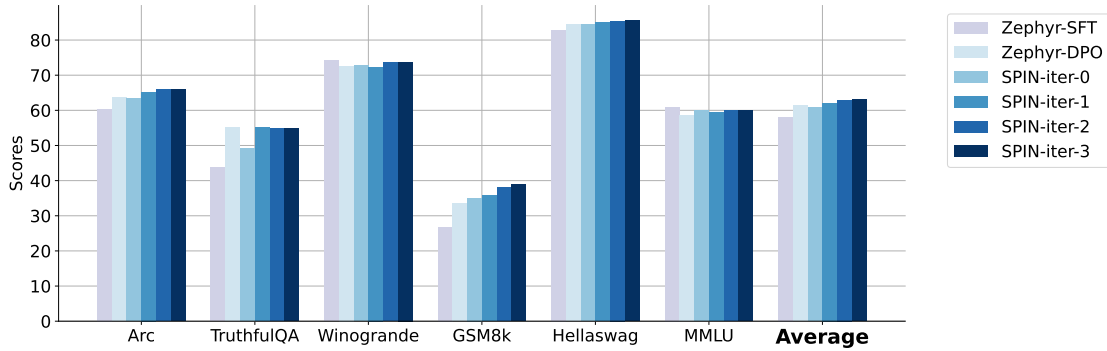[2]https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

Figure 3. Performance comparison with DPO training across the six benchmark datasets. Self-play at iteration 0 achieves comparable performance to DPO training with 62k new data. At iteration 1, self-play has already surpassed DPO training on the majority of datasets.
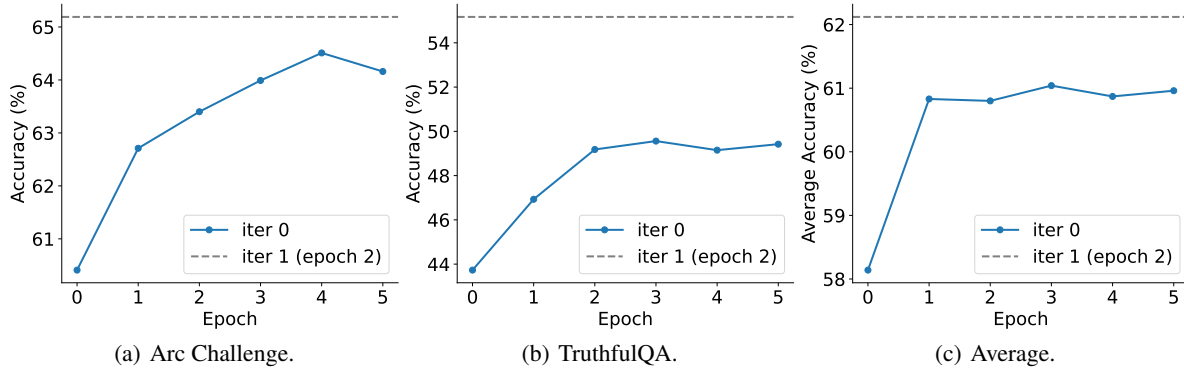

| (a) Arc Challenge. | (b) TruthfulQA. | (c) Average. |

Figure 4. The `SPIN` training dynamics of `zephyr-7b-sft-full` on the 50k synthetic data with regard to the number of training epochs during iteration 0. We can observe that iterative training is pivotal as training for more epochs during iteration 0 reaches a limit and cannot surpass iteration 1.
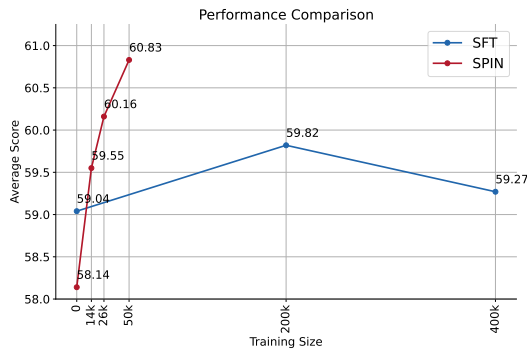
Figure 5. The scaling effect of training size of `SPIN` compared to SFT on the average score of Open LLM Leaderboard. For `SPIN`, we consider training data of sizes 14k, 26k and 50k where the larger dataset contains the smaller dataset. The starting point for `SPIN` (with x-axis 0) is the `zephyr-7b-sft-full` checkpoint, which has been fine-tuned on Ultrachat200k for 1 epoch. We report the model performance trained for 1 epoch with `SPIN` on the varying sizes of dataset. We additionally compare with SFT, where we fine-tune Mistral-7B on Ultrachat200k for 3 consecutive epochs and report the model performance at the first epoch as the starting point (with x-axis 0).

Impact Statement

This paper presents work whose goal is to advance the field of Large Language Models. We believe that our work contribute meaningfully to the field, specifically on leveraging synthetic data to enhance LLM without the requirement for human preference annotations. The synthetic data generated by `SPIN` may be used to further augment the training of various language models. Moreover, `SPIN` demonstrated a substantial improvement in LLMs' capabilities, opening new avenues for their application in various downstream tasks. This advancement underscores the transformative potential of LLM fine-tuning in both technological and societal contexts.

References

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Anthony, T., Tian, Z., and Barber, D. Thinking fast and slow

with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Bansal, T., Pachocki, J., Sidor, S., Sutskever, I., and Mordatch, I. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*, 2018.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open llm leaderboard, 2023.

bench authors, B. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., Chen, Y., Ecoffet, A., Joglekar, M., Leike, J., et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. 2023.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Cheng, P., Yang, Y., Li, J., Dai, Y., and Du, N. Adversarial preference optimization, 2023.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Cirik, V., Hovy, E., and Morency, L.-P. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*, 2016.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Deng, Y., Zhang, W., Chen, Z., and Gu, Q. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*, 2023.

Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Frei, S., Zou, D., Chen, Z., and Gu, Q. Self-training converts weak learners to strong learners in mixture models. In *International Conference on Artificial Intelligence and Statistics*, pp. 8003–8021. PMLR, 2022.

Freund, Y. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023a.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023b.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. Accelerate: Training and inference at scale made simple, efficient and adaptable., 2022.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. Is multiagent deep reinforcement learning the answer or the question? a brief survey. *learning*, 21:22, 2018.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

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

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jolicoeur-Martineau, A. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

Josifoski, M., Sakota, M., Peyrard, M., and West, R. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *arXiv preprint arXiv:2303.04132*, 2023.

Kearns, M. and Valiant, L. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

Kou, Y., Chen, Z., Cao, Y., and Gu, Q. How does semi-supervised learning with pseudo-labelers work? a case study. In *The Eleventh International Conference on Learning Representations*, 2022.

Kumar, M., Packer, B., and Koller, D. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Challenges in Representation Learning Workshop*, 2013.

Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., and Rastogi, A. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Lee, Y. J. and Grauman, K. Learning the easy things first: Self-paced visual category discovery. In *CVPR 2011*, pp. 1721–1728. IEEE, 2011.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Li, Y., Bubeck, S., Eldan, R., Giorno, A. D., Gunasekar, S., and Lee, Y. T. Textbooks are all you need ii: phi-1.5 technical report, 2023.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Liu, B., Bubeck, S., Eldan, R., Kulkarni, J., Li, Y., Nguyen, A., Ward, R., and Zhang, Y. Tinygsm: achieving> 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*, 2023.

Liu, C., He, S., Liu, K., Zhao, J., et al. Curriculum learning for natural answer generation. In *IJCAI*, pp. 4223–4229, 2018.

Liu, F., Ge, S., and Wu, X. Competence-based multimodal curriculum learning for medical report generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3001–3012, 2021.

Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J., Tao, C., Geng, X., Lin, Q., Chen, S., and Zhang, D. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Mishra, S., Khashabi, D., Baral, C., and Hajishirzi, H. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*, 2021.

Mroueh, Y. and Sercu, T. Fisher gan. *Advances in neural information processing systems*, 30, 2017.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29 (2):429–443, 1997.

Muller, P., Omidshafiei, S., Rowland, M., Tuyls, K., Perolat, J., Liu, S., Hennes, D., Marris, L., Lanctot, M., Hughes, E., et al. A generalized training approach for multiagent learning. *arXiv preprint arXiv:1909.12823*, 2019.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Prasad, A., Stengel-Eskin, E., and Bansal, M. Rephrase, augment, reason: Visual grounding of questions for vision-language models. *arXiv preprint arXiv:2310.05861*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

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

Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 44(1.2):206–226, 2000.

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., and Leike, J. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Schapire, R. E. The strength of weak learnability. *Machine learning*, 5:197–227, 1990.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017a.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017b.

Singh, A., Co-Reyes, J. D., Agarwal, R., Anand, A., Patil, P., Liu, P. J., Harrison, J., Lee, J., Xu, K., Parisi, A., et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. Baby steps: How "less is more" in unsupervised dependency parsing. In *NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.

Tesauro, G. et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023a.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rush, A. M., and Wolf, T. The alignment handbook, 2023b.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 1999.

Victor, S., Albert, W., Colin, R., Stephen, B., Lintang, S., Zaid, A., Antoine, C., Arnaud, S., Arun, R., Manan, D., et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D., and Silver, D. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II, 2019.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wu, J., Liang, Y., Akbari, H., Wang, Z., Yu, C., et al. Scaling multimodal pre-training via cross-modality gradient harmonization. *Advances in Neural Information Processing Systems*, 35:36161–36173, 2022.

Xu, J., Lee, A., Sukhbaatar, S., and Weston, J. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

Yang, Y., Singh, A. K., Elhoushi, M., Mahmoud, A., Tirumala, K., Gloeckle, F., Rozière, B., Wu, C.-J., Morcos, A. S., and Ardalani, N. Decoding data quality via synthetic corruptions: Embedding-guided pruning of code data. *arXiv preprint arXiv:2312.02418*, 2023.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Yuan, Z., Yuan, H., Li, C., Dong, G., Tan, C., and Zhou, C. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhang, D., Meng, D., Li, C., Jiang, L., Zhao, Q., and Han, J. A self-paced multiple-instance learning framework for co-saliency detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 594–602, 2015.

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K., and Carpuat, M. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*, 2018.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Further Related Work

Curriculum Learning. In deep learning, it has been observed that training models using data samples arranged in a strategically meaningful order can lead to improved performance compared to training on randomly shuffled data. This approach is commonly known as curriculum learning (Bengio et al., 2009; Soviany et al., 2022). Initial studies in curriculum learning introduced efficient algorithms that adhere to an 'easy-to-hard' progression (Spitkovsky et al., 2009; Kumar et al., 2010; Lee & Grauman, 2011; Zhang et al., 2015). In the field of Natural Language Processing (NLP), criteria such as sentence length and term frequency are commonly utilized (Cirik et al., 2016; Zhang et al., 2018; Liu et al., 2018). More recent developments include the application of curriculum learning algorithms in multi-modal learning (Liu et al., 2021; Wu et al., 2022). Our work shares a similar idea to curriculum learning, wherein the training data evolves iteratively—beginning with responses that are easy to distinguish from human-annotated data and gradually progressing to more challenging instances.

Generative Adversarial Networks. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) represent a distinct class of generative models, characterized by their unique adversarial process. To enhance training stability and data quality, Mao et al. (2017) introduced the Least Squares GAN, employing a least squares loss function for the discriminator. A significant advancement in GANs involves the use of Integral Probability Metrics (IPM) (Müller, 1997), particularly highlighted in the development of Wasserstein GAN by Arjovsky et al. (2017). This model employs IPM in its loss design, enhancing training stability. Since then, IPMs have become crucial in GAN design (Mroueh & Sercu, 2017; Gulrajani et al., 2017), particularly in constraining the discriminator to a specific function class, thereby preventing it from overpowering the generator. Furthermore, Jolicoeur-Martineau (2018) generalized IPM-based GANs by introducing relativistic discriminator and proposed Relativistic GAN. It is worth noting that the objective function defined in our (4.2) is similar to Relativistic GAN (Jolicoeur-Martineau, 2018) and reduces to an IPM framework in Wasserstein GAN (Arjovsky et al., 2017) with a linear loss. However, our approach differs in both the choice of the function class and the training procedure. Inspired by GAN, Cheng et al. (2023) proposed an adversarial learning framework named Adversarial Preference Optimization (APO) that trains the LLM and a reward model in an adversarial game. Similarly related to the spirit of our method, Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016) was proposed to train separate discriminator and policy networks in each iteration. In contrast to the above methods, SPIN relies on self-play where both the main player and the opponent player are the same LLM from two consecutive iterations.

Alignment with AI Feedback. The objective of alignment is to fine-tune LLMs to align with human intentions. In addition to using human demonstrations, AI feedback is emerging as a crucial component in the alignment process. Constitutional AI (Bai et al., 2022b) leveraged AI feedback to align language models through a combination of both supervised learning and reinforcement learning (RL) phases. In the RL phase, (Bai et al., 2022b) applied Reinforcement Learning from AI Feedback (RLAIF), training a reward model using AI-generated preferences, followed by RL using the reward. Lee et al. (2023) demonstrated that AI feedback can achieve comparable or superior performance to human feedback in RL fine-tuning. They also demonstrated that RLAIF can improve upon an SFT policy even when the LLM labeler has the same size as the policy. Saunders et al. (2022) studied the scaling properties of self-critique and introduced a framework for comparing the critique ability to generation and discrimination ability. Self-critique models employ the LLM itself to generate natural language critiques through behavioral cloning, assisting human evaluators.

B Experiments

B.1 Hyperparameters and Implementation Details

Table 1. Detailed information of HuggingFace Open LLM Leaderboard. For each evaluation dataset, we present the number of few-shot examples and metric adopted for evaluation.

Datasets	Arc	TruthfulQA	Winogrande	GSM8k	HellaSwag	MMLU
# few-shot	25	0	5	5	10	5
Metric	acc_norm	mc2	acc	acc	acc_norm	acc

We use the Alignment Handbook library (Tunstall et al., 2023b) as the codebase for our self-play fine-tuning method SPIN, which includes DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) and FlashAttention-2 (Dao, 2023) to reduce training cost. We train our models with RMSProp (Hinton et al., 2012) optimizer with no weight decay for all iterations as commonly used in fine-tuning LLMs for alignment, with a global batch size of 64, 10% warmup steps and bfloat16 precision. We set the peak learning rate to be 5e-7 for iterations 0 and 1, and decay this peak learning rate to 1e-7 for iteration 2 and 3 as we are

approaching the end of self-play fine-tuning. Lastly, we choose $\beta = 0.1$ and max sequence length to be 2048 tokens as in Tunstall et al. (2023b). We note that at the last iteration (iter-3) where the model is close to convergence, we increase the value of β to 5.0. We use the Accelerate library (Gugger et al., 2022) to generate our synthetic data using distributed inference with multiple GPUs with a global batch size of 64. We consider the prompting template "### Instruction: {prompt}\n\n### Response: " as commonly used in Taori et al. (2023). For Ultrachat200k containing multi-round conversations, we only sample the first round as our prompt and ground truth completion pairs.

B.2 Training Overhead

The cost overhead introduced by SPIN is mostly the generation of synthetic data from the LLM that we train. The cost of the fine-tuning process remains computationally equal to that of SFT and DPO. We report both the generation and training time for SPIN in Table 2 . Results were obtained using a machine with 8xA100 (80G) GPUs. For per 64 examples, the generation time and training time are 6.69s and 10s respectively.

Table 2. Generation and Training Times for Different Iterations

Iteration	Iter 0		Iter 1		Iter 2		Iter 3	
Process	Generation	Training	Generation	Training	Generation	Training	Generation	Training
Time	1.45h	4.32h	1.45h	8.64h	1.45h	8.64h	1.45h	8.64h

It is evident that the generation time is dominated by the training time at each iteration. The estimated time in Table 2 is based on the fact that we generate 50k examples per iteration. Please note that the doubled training time from iter 1 to iter 3 is attributed to the utilization of a double-sized dataset (the combination of 50k synthetic data from the previous iteration and 50k synthetic data in the current iteration), as discussed in our Section 6.1.

B.3 Additional Experiment Result for SPIN+DPO

`SPIN` requires only the SFT data to improve over the traditional SFT stage and can sit between SFT and RL finetuning. Suppose additional preference data is provided, we can use the additional data to further improve the performance of the model after SPIN using RL fine-tuning.

Starting at `SPIN` iteration 3, we further train the model with DPO for two epochs on the 62k preference data from the UltraFeedback Binarized dataset (Cui et al., 2023), which consists of both chosen and rejected responses evaluated by GPT-4. Detailed performances are presented in Table 3.

Table 3. Performance of `SPIN` + DPO based on `zephyr-7b-sft-full` across HuggingFace Open LLM Leaderboard datasets, compared with all baselines. We also denote the average improvement over last iteration in the Average column.

Model	Arc	TruthfulQA	Winogrande	GSM8k	HellaSwag	MMLU	Average
`zephyr-7b-dpo-full`	63.65	55.19	72.61	33.43	84.44	58.52	61.31
`zephyr-7b-sft-full`	60.41	43.73	74.19	26.76	82.85	60.92	58.14
`SPIN` iteration 0	63.40	49.18	72.69	35.10	84.38	60.03	$60.80_{(+2.66)}$
`SPIN` iteration 1	65.19	55.17	72.30	35.78	84.96	59.34	$62.12_{(+1.32)}$
`SPIN` iteration 2	65.96	54.91	73.56	38.06	85.41	59.93	$62.97_{(+0.85)}$
`SPIN` iteration 3	65.87	54.90	73.72	38.97	85.54	59.99	$63.16_{(+0.19)}$
`SPIN` iteration 3 + DPO	66.47	60.07	78.06	37.98	86.17	59.68	$64.05_{(+0.89)}$

We can observe that the checkpoint trained by `SPIN` can be further improved using DPO, yielding an enhancement of 0.89% on average. Notably, the improvement is particularly significant on the TruthfulQA benchmark with around 5%.

B.4 Further Experiment Results

In Table 4, we show the detailed performance of `SPIN` at different iterations on each of the task in Open LLM Leaderboard. In Table 5, we also show the performance of SFT from `zephyr-7b-sft-full` on Ultrachat200k for one epoch. While self-play fine-tuning with synthetic data from `zephyr-7b-sft-full` effectively improves its performance, simply fine-tuning it again on the SFT data leads to degraded performance, as similarly observed in Figure 5.

Further Investigation on More Tasks. Here, we further investigate the performance of `SPIN` on a broader variety of

Table 4. Test performance of SPIN based on `zephyr-7b-sft-full` across HuggingFace Open LLM Leaderboard datasets. We also denote the average improvement over last iteration in the Average column.

Model	Arc	TruthfulQA	Winogrande	GSM8k	HellaSwag	MMLU	Average
`zephyr-7b-sft-full`	60.41	43.73	74.19	26.76	82.85	60.92	58.14
SPIN iteration 0	63.40	49.18	72.69	35.10	84.38	60.03	$60.80_{(+2.66)}$
SPIN iteration 1	65.19	55.17	72.30	35.78	84.96	59.34	$62.12_{(+1.32)}$
SPIN iteration 2	65.96	54.91	73.56	38.06	85.41	59.93	$62.97_{(+0.85)}$
SPIN iteration 3	65.87	54.90	73.72	38.97	85.54	59.99	$63.16_{(+0.19)}$

Table 5. Test performance of `zephyr-7b-sft-full` fine-tuned on Ultrachat200k for 1 more epoch across HuggingFace Open LLM benchmark datasets. SFT fails to further leverage the fine-tuning data for performance enhancement and even results in degraded performance.

Model	Arc	TruthfulQA	Winogrande	GSM8k	HellaSwag	MMLU	Average
`zephyr-7b-sft-full`	60.41	43.73	74.19	26.76	82.85	60.92	58.14
SFT epoch 1	57.76	44.39	75.77	25.85	81.69	57.89	57.23

tasks, including MT-Bench (Zheng et al., 2023), Big-Bench (bench authors, 2023) and OpenBookQA (Mihaylov et al., 2018) in addition to the Open LLM Leaderboard tasks. Specifically, we use the following tasks from Big-Bench-Hard for a more comprehensive evaluation, including Causal Judgment (causal reasoning), Sports Understanding (commonsense reasoning) and Formal Fallacies (logical reasoning). In Table 6, we show the resulting scores of SPIN on MT-Bench as well as those tasks from Big-Bench. In Figure 6, we detail the model performances on MT-Bench with regard to different types of questions. We can see a notably robust improvement in the performance of SPIN on various tasks besides the HuggingFace Benchmark, without major degradation. Notably, on MT-Bench, the model fine-tuned by SPIN has surpassed the performance of `vicuna-13b-v1.5` (Chiang et al., 2023) with a score of 6.57.

Table 6. Test performance on other reasoning benchmark datasets for SPIN at different iterations and `zephyr-7b-sft-full`. We report the average score for MT-Bench and the accuracy score for Big Bench datasets under standard few-shot CoT evaluation. On OpenBookQA, we report `acc_norm` with 1-shot example as used in Anil et al. (2023). As similar to Open LLM Leaderboard evaluation, we observe a steady improvement in performance on the other benchmark tasks, with no significant degradation.

Model	MT-Bench	BB-causal	BB-formal	BB-sports	OpenBookQA
`zephyr-7b-sft-full`	5.94	56.15	49.6	96.0	45.4
SPIN iteration 0	$6.46_{(+0.52)}$	57.75	51.6	95.2	46.8
SPIN iteration 1	$6.65_{(+0.19)}$	58.82	51.2	95.2	47.2
SPIN iteration 2	$6.78_{(+0.13)}$	59.36	51.2	94.4	47.6

B.5 Generation Examples

In Tables 7 and 8, we further provide the generation examples of our fine-tuned model by SPIN from different iterations. We can observe an improvement in response quality as compared to the generation of the SFT checkpoint. Meanwhile, the model generations at higher iterations typically becomes more concise than iteration 0 and resemble the ground truth completion better.
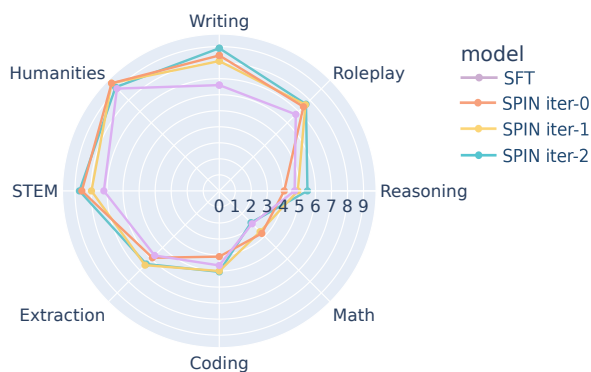
Figure 6. Model performance on MT-Bench. We compare `SPIN` across different iterations with the base SFT model. Starting from iteration 1, our fine-tuned model by `SPIN` robustly outperforms the SFT checkpoint on all evaluation aspects.

Table 7. Generation example of our fine-tuned model by `SPIN` at different iterations.

Prompt	How does the economic growth of Horsham compare to nearby towns?
Ground truth	I do not have access to the latest economic data on horsham and nearby towns. However, you can check the latest economic statistics and reports provided by the local government or the official websites of horsham and nearby towns for a detailed comparison of their economic growth.
`zephyr-7b=sft-full`	According to the given material, Horsham was growing "significantly" faster than nearby towns, indicating that it was economically thriving.\n However, the given material does not specify the exact rates or levels of economic growth of Horsham compared to nearby towns. Without this information, it is challenging to determine whether the differences in economic growth between Horsham and nearby towns were significant or minimal.
Iteration 0	I do not have access to the latest economic data for horsham and nearby towns. However, horsham is known to be a prosperous town in west sussex, uk, with a diverse economy that includes agriculture, manufacturing, and services. The town has a strong retail sector, and it is also home to several large companies, including the global pharmaceutical company novartis. In terms of comparison with nearby towns, horsham is generally considered to be one of the more affluent towns in the region. However, the economic growth of horsham and its neighboring towns may vary depending on various factors such as industry, investment, and government policies.
Iteration 1	I do not have access to real-time data or information about the most recent economic growth of horsesham and nearby towns. However, according to the recent reports from various sources such as the office for national statistics, horsesham's economy primarily relies on retail, healthcare, education, and public administration industries. Compared to nearby towns such as lewes, chichester, and bognor regis, horsesham's economy may have a different composition and growth rate based on their specific industries and economic policies.
Iteration 2	I don't have access to the most recent economic data. However, according to a 2019 report by pwc, horsham's expected economic growth up to 2025 was projected to be around 2.3% per annum, which is slightly below the average for the south east of england (2.8%). Neighbouring towns like crawley and bognor regis were projected to experience similar growth rates. note: this may not reflect the impact of the covid-19 pandemic that occurred in 2020.

Table 8. Another generation example of our fine-tuned model by `SPIN` at different iterations.

Prompt	Can you clarify if Poppelsdorfer Allee 45 and the old site of the Mathematical Institute are the same location? Answer according to: Go straight to Poppelsdorfer Allee 45 and sign-in.\n There is a big lecture hall at the old site of the Mathematical Institute (Wegelerstr. 10, Bonn).
Ground truth	No, Poppelsdorfer Allee 45 and the old site of the Mathematical Institute (Wegelerstr. 10, Bonn) are not the same location. To sign-in, one should go straight to Poppelsdorfer Allee 45. However, there is a big lecture hall at the old site of the Mathematical Institute.
`zephyr-7b=sft-full`	Yes, Poppelsdorfer Allee 45 and the old site of the Mathematical Institute are the same location.
Iteration 0	Yes, Poppelsdorfer Allee 45 and the old site of the Mathematical Institute (Wegelerstr. 10, Bonn) are the same location. The Mathematical Institute has been relocated and renamed as the Institute for Mathematics and Theoretical Physics, but the building where it used to be is still referred to as the old site. The big lecture hall at the old site still exists and may be used for events or classes.
Iteration 1	No, Poppelsdorfer Allee 45 and the old site of the Mathematical Institute are not the same location. While Poppelsdorfer Allee 45 is where attendees should sign-in, the old site of the Mathematical Institute (located at Wegelerstr. 10, Bonn) has a large lecture hall mentioned in the statement. Therefore, it can be assumed that the two locations are separate and not the same.
Iteration 2	Poppelsdorfer Allee 45 and the old site of the Mathematical Institute (Wegelerstr. 10, Bonn) are two different locations. As per the instruction, you need to go straight to Poppelsdorfer Allee 45 and sign-in. However, there is a lecture hall mentioned at the old site of the Mathematical Institute (Wegelerstr. 10, Bonn).

C   Proof of Theorems in Section 5

C.1   Proof of Theorem 5.2

Proof of Theorem 5.2. To begin with, we prove the "Sufficiency" in Theorem 5.2. Since $p_{\text{data}}(\cdot|\mathbf{x}) = p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})$, by symmetry property of $\mathbf{y}$ and $\mathbf{y}'$, we have for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ that

$$
\begin{aligned}
2L_{\text{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) &= \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right] \\
&\quad + \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}'\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right] \\
&= \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right. \\
&\quad \left. + \ell\left(\gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})} - \gamma\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)\right] \\
&\geq 2\mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\frac{\gamma}{2}\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \frac{\gamma}{2}\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right.\right. \\
&\quad \left.\left. + \frac{\gamma}{2}\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})} - \frac{\gamma}{2}\log\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)\right] \\
&= 2\ell(0),
\end{aligned}
$$

where the inequality is due to Jensen's inequality (recalling that ℓ is convex in Assumption 5.1). Therefore, we have that $L_{\text{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) \geq \ell(0) = L_{\text{SPIN}}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_t)$, which means that $\boldsymbol{\theta}_t$ is the global optimum of (4.7). As a consequence, the gradient at the point $\boldsymbol{\theta}_t$ is zero, which concludes $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$.

Next, we prove the "Necessity". Define $g(\lambda)$ as follows:

$$g(\lambda) = \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\lambda\log\frac{p_{\text{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \lambda\log\frac{p_{\text{data}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right].$$

Then we have $g(0) = \ell(0)$ and

$$\begin{aligned}
g'(0) &= \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell'(0)\left(\log\frac{p_{\text{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \log\frac{p_{\text{data}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right] \\
&= \ell'(0)\left(\mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x})}\left[\log\frac{p_{\text{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right] - \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\log\frac{p_{\text{data}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right]\right) \\
&= \ell'(0)\left[\text{KL}\big(p_{\text{data}}(\cdot|\mathbf{x})\big\|p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})\big) + \text{KL}\big(p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})\big\|p_{\text{data}}(\cdot|\mathbf{x})\big)\right] \\
&< 0,
\end{aligned}$$

where the last inequality is due to the condition that $\ell'(0) < 0$. Therefore, there exist a λ_0 such that for all $0 < \lambda < \lambda_0$, we have $g(\lambda) < \ell(0)$. Choose $\boldsymbol{\theta}^*$ such that $p_{\boldsymbol{\theta}^*}(\mathbf{y}|\mathbf{x}) = p_{\text{data}}(\mathbf{y}|\mathbf{x})$. For those $0 < \lambda < \lambda_0$, we have that

$$\begin{aligned}
L_{\text{SPIN}}(\boldsymbol{\theta}^*,\boldsymbol{\theta}_t) &= \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\boldsymbol{\theta}^*}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\lambda\log\frac{p_{\boldsymbol{\theta}^*}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \lambda\log\frac{p_{\boldsymbol{\theta}^*}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right] \\
&= \mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\left(\lambda\log\frac{p_{\text{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})} - \lambda\log\frac{p_{\text{data}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)\right] \\
&= g(\lambda) \\
&< g(0) \\
&= L_{\text{SPIN}}(\boldsymbol{\theta}_t,\boldsymbol{\theta}_t),
\end{aligned}$$

where the second equality holds by the choice of $p_{\boldsymbol{\theta}^*}(\cdot|\mathbf{x})$, and the inequality holds due to the choice of λ. Therefore, we conclude that $\boldsymbol{\theta}_t$ is not the global optimum of (4.7) if $p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x}) \neq p_{\text{data}}(\cdot|\mathbf{x})$. $\qquad\square$

C.2 Proof Theorem 5.4

We need the following auxiliary lemma before we prove Theorem 5.4.

Lemma C.1. Suppose that $\ell(t) = \log(1 + \exp(-t))$ and for $a, b > 0$, the following inequality holds

$$a\ell(t) + b\ell(-t) \geq a\log(1 + b/a) + b\log(1 + a/b),$$

the equality holds if and only if $t = \log(a/b)$.

Proof of Lemma C.1. Define $g(t) = a\ell(t) + b\ell(-t) = a\log(1 + \exp(-t)) + b\log(1 + \exp(t))$, then we have

$$g'(t) = -\frac{a\exp(-t)}{1 + \exp(-t)} + \frac{b\exp(t)}{1 + \exp(t)} = \frac{-a + b\exp(t)}{1 + \exp(t)}.$$

Therefore, $g'(t) < 0$ when $t < \log(a/b)$, $g'(t) > 0$ when $t > \log(a/b)$, which indicates that g achieves it minimum at $t = \log(a/b)$ which concludes the proof. $\qquad\square$

Lemma C.1 shows that the global minimum of $a\ell(t) + b\ell(-t)$ is achieved when $t = \log(a/b)$. Based on Lemma C.1, we can further prove that (4.2) with the logistic loss function has a closed-form solution if we ignore the constraint set $\mathcal{F}_t$.

Lemma C.2. Denote $p_+(\mathbf{y}, \mathbf{y}', \mathbf{x}) = q(\mathbf{x}) \cdot p_{\text{data}}(\mathbf{y}|\mathbf{x}) \cdot p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})$ and $p_-(\mathbf{y}, \mathbf{y}', \mathbf{x}) = q(\mathbf{x}) \cdot p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x}) \cdot p_{\text{data}}(\mathbf{y}|\mathbf{x})$,

$$\mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\big[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\big] \geq \log 2 - \text{JSD}(p_+\|p_-),$$

where $\text{JSD}(p_+\|p_-)$ represents the Jensen–Shannon divergence which is defined as follows

$$\text{JSD}\left(p\big\|q\right) = \frac{1}{2}\text{KL}\left(p\big\|\frac{p+q}{2}\right) + \frac{1}{2}\text{KL}\left(q\big\|\frac{p+q}{2}\right),$$

where $\mathrm{KL}(\cdot\|\cdot)$ is KL-divergence. JSD is always non-negative and equals zero if and only if p_+ and p_- are identical. Moreover, the global minimum value $\log 2 - \mathrm{JSD}(p_+\|p_-)$ is achieved by f^* if and only if,

$$f^*(\mathbf{x}, \mathbf{y}) = Z(\mathbf{x}) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right),$$

where $Z(\mathbf{x})$ is any function that is possibly dependent on $\mathbf{x}$.

Proof of Lemma C.2. We rewrite the objective function in the following formula,

$$2\mathbb{E}_{\mathbf{x}\sim q(\cdot),\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x}),\mathbf{y}'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\left[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\right]$$

$$= \int q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})\left[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\right]d\mathbf{y}d\mathbf{y}'$$

$$+ \int q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\left[\ell\big(f(\mathbf{x},\mathbf{y}') - f(\mathbf{x},\mathbf{y})\big)\right]d\mathbf{y}d\mathbf{y}'$$

$$= \int q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)$$

$$+ q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\ell\big(f(\mathbf{x},\mathbf{y}') - f(\mathbf{x},\mathbf{y})\big)d\mathbf{y}d\mathbf{y}'$$

$$\overset{(i)}{\geq} \int q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})\log\left(1 + \frac{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)$$

$$+ q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\log\left(1 + \frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)d\mathbf{y}d\mathbf{y}',$$

where the inequality is due to $a\ell(t) + b\ell(-t) \geq a\log(1 + b/a) + b\log(1 + a/b)$ in Lemma C.1 with $a = q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x}), b = q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x}), t = f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')$. The equality (i) holds if and only if the following equation holds almost surely for any $\mathbf{x},\mathbf{y},\mathbf{y}'$,

$$f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}') = \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right). \tag{C.1}$$

Equation (C.1) is equivalent to

$$f(\mathbf{x},\mathbf{y}) - \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right) = f(\mathbf{x},\mathbf{y}') - \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)$$

holds almost surely for any $\mathbf{x},\mathbf{y},\mathbf{y}'$. Therefore, the equality (i) holds if and only if there exists some $Z(\mathbf{x})$ such that

$$f(\mathbf{x},\mathbf{y}) = Z(\mathbf{x}) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right).$$

Recall that $p_+(\mathbf{y},\mathbf{y}'|\mathbf{x}) = p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})\cdot p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})$ and $p_-(\mathbf{y},\mathbf{y}'|\mathbf{x}) = p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\cdot p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})$. Then, the right-hand side of (i) can be written as

$$\int q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})\log\left(1 + \frac{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}\right)$$

$$+ q(\mathbf{x})p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\log\left(1 + \frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}'|\mathbf{x})}{p_{\mathrm{data}}(\mathbf{y}'|\mathbf{x})p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)d\mathbf{y}d\mathbf{y}'$$

$$= \int p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})\log\left(1 + \frac{p_-(\mathbf{y},\mathbf{y}'|\mathbf{x})}{p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})}\right) + p_-(\mathbf{y},\mathbf{y}'|\mathbf{x})\log\left(1 + \frac{p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})}{p_-(\mathbf{y},\mathbf{y}'|\mathbf{x})}\right)d\mathbf{y}d\mathbf{y}'$$

$$= 2\log 2 + \int p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})\log\left(\frac{1/2[p_-(\mathbf{y},\mathbf{y}'|\mathbf{x}) + p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})]}{p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})}\right)$$

$$+ p_-(\mathbf{y},\mathbf{y}'|\mathbf{x})\log\left(\frac{1/2[p_-(\mathbf{y},\mathbf{y}'|\mathbf{x}) + p_+(\mathbf{y},\mathbf{y}'|\mathbf{x})]}{p_-(\mathbf{y},\mathbf{y}'|\mathbf{x})}\right)d\mathbf{y}d\mathbf{y}'$$

$$= 2\log 2 - \mathrm{KL}\left(p_+\left\|\frac{p_+ + p_-}{2}\right.\right) - \mathrm{KL}\left(p_-\left\|\frac{p_+ + p_-}{2}\right.\right)$$

$$= 2\log 2 - 2 \cdot \mathrm{JSD}(p_+\|p_-),$$

where the last equality is by the definition of JSD. This concludes the proof. $\square$

Lemma C.2 provides a closed-form solution to (4.2) if we ignore the constraint set $\mathcal{F}_t$. If this closed-form solution belongs to $\mathcal{F}_t$, then it should also be the solution to (4.2). This observation is the key to the proof of Theorem 5.4.

Proof of Theorem 5.4. Under the condition of Theorem 5.4, there exists a $p_{\boldsymbol{\theta}}$ such that

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big)^{1/\lambda}.$$

Therefore, there exists a function $\widehat{Z}(\mathbf{x})$ such that

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \widehat{Z}(\mathbf{x}) \cdot p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big)^{1/\lambda}. \tag{C.2}$$

Applying logarithm function on both side of (C.2) yields

$$\lambda \log(\widehat{Z}(\mathbf{x})) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right) = \lambda \log\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right) \in \mathcal{F}_t.$$

By Lemma C.2, $f^*(\mathbf{x},\mathbf{y}) = \lambda\log(\widehat{Z}(\mathbf{x})) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)$ is the global minimum of the following minimization problem,

$$\underset{f}{\mathrm{argmin}}\, \mathbb{E}_{\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x}),y'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\big[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\big]. \tag{C.3}$$

Since $f^* \in \mathcal{F}_t$, $f^*(\mathbf{x},\mathbf{y}) = \lambda\log(\widehat{Z}(\mathbf{x})) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)$ is also the global optimum of the optimization problem (4.2),

$$\underset{f\in\mathcal{F}_t}{\mathrm{argmin}}\, \mathbb{E}_{\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x}),y'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\big[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\big].$$

Therefore, we have proved that

$$\min_{f}\, \mathbb{E}_{\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x}),y'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\big[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\big]$$

$$= \min_{f\in\mathcal{F}_t}\, \mathbb{E}_{\mathbf{y}\sim p_{\mathrm{data}}(\cdot|\mathbf{x}),y'\sim p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{x})}\big[\ell\big(f(\mathbf{x},\mathbf{y}) - f(\mathbf{x},\mathbf{y}')\big)\big]$$

$$= \min_{\boldsymbol{\theta}\in\Theta}\, L_{\mathrm{SPIN}}(\boldsymbol{\theta},\boldsymbol{\theta}_t). \tag{C.4}$$

Since $\boldsymbol{\theta}_{t+1}$ is the global minimum of $L_{\mathrm{SPIN}}(\boldsymbol{\theta},\boldsymbol{\theta}_t)$. Then by (C.4), $\lambda\log\left(\frac{p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right)$ should be the global minimum of problem (C.3). By Lemma C.2, there exists $Z(\mathbf{x})$ such that

$$\lambda\log\left(\frac{p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right) = Z(\mathbf{x}) + \log\left(\frac{p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})}{p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})}\right),$$

which leads to the result that $p_{\boldsymbol{\theta}_{t+1}}(\mathbf{y}|\mathbf{x}) \propto p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big(p_{\mathrm{data}}(\mathbf{y}|\mathbf{x})/p_{\boldsymbol{\theta}_t}(\mathbf{y}|\mathbf{x})\big)^{1/\lambda}$.

$\square$