GAME-THEORETIC REGULARIZED SELF-PLAY ALIGNMENT OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-play alignment algorithms have been developed as effective methods for fine-tuning large language models (LLMs), formulating preference optimization as a two-player game. However, the regularization to the reference policy, which is crucial for mitigating over-optimization, has been insufficiently investigated in self-play alignment. In this paper, we show that our regularization method can improve the unregularized self-play significantly. To study the impact of different regularization in self-play alignment, we propose Regularized Self-Play Policy Optimization (RSPO), a generalized framework that allows for regularizing self-play by simply adding a chosen regularization term into the loss, while maintaining provable last-iterate convergence to the Nash Equilibrium of the corresponding regularized game. Surprisingly, empirical evaluations using the Mistral-7B-Instruct base model reveal that forward KL divergence regularization reduces response length in RSPO, whereas reverse KL divergence markedly improves raw win rates. RSPO with a linear combination of forward and reverse KL divergence regularization substantially increase the length-controlled win rate in AlpacaEval-2, elevating the unregularized self-play alignment method (SPPO) from 28.53% to 35.44%. Finally, we show that RSPO also improves the response diversity.

1 INTRODUCTION

Large Language Models (LLMs) recently have obtained remarkable capabilities to accomplish a range of tasks (Jiang et al., 2023a; Dubey et al., 2024; DeepSeek-AI et al., 2025), generating more desirable and helpful content following the user's intention. One of the most important methods to align LLMs with human intentions is Reinforcement Learning from Human Feedback (RLHF), maximizing a preference-based reward penalized by a reverse KL regularization term of LLM policy and a supervised fine-tuning (SFT) reference model (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2024; Azar et al., 2024; Xiong et al., 2024). This regularization is crucial in RLHF to prevent over-optimization, which has been extensively studied and even extended beyond KL divergence (Go et al., 2023; Huang et al., 2024).

Self-play is a general line of works conducting iterative self-competition of models, which has been demonstrated as an effective approach for improving AI systems (Goodfellow et al., 2020; Wang et al., 2022), particularly in strategic decision-making problems (Silver et al., 2016; Heinrich & Silver, 2016; Pinto et al., 2017; Brown & Sandholm, 2018). In the human alignment of LLMs, self-play recently started to be used and has shown superior empirical performance than other iterative RLHF methods on benchmarks like AlpacaEval and Arena-Hard Evaluation (Dubois et al., 2024; Jiang et al., 2024; Wu et al., 2024; Rosset et al., 2024). By formulating the preference optimization problem as a two-player game, self-play alignment methods seek to identify a Nash Equilibrium (NE) of the game in which utility is determined by a general preference

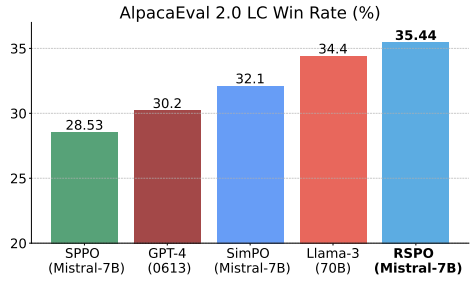

Figure 1: Our **Regularized** Self-Play Policy Optimization (**RSPO**) with base model Mistral-**7B**-Instruct outperforms Llama-3-**70B**, GPT-4 (0613), and (unregularized) Self-Play Policy Optimization (SPPO) (Wu et al., 2024) in AlpacaEval-2 LC win rate.

model (Munos et al., 2023; Calandriello et al., 2024; Azar et al., 2024). This NE is regarded as the most aligned LLM policy, achieved without Bradley-Terry (BT) assumption (David, 1963).

Despite the significant empirical improvements achieved through self-play, the impact of regularization to the reference policy—commonly used in RLHF to mitigate over-optimization—has received insufficient investigation in self-play alignment. Most existing self-play methods lack explicit regularization (Swamy et al., 2024; Rosset et al., 2024; Wu et al., 2024; Wang et al., 2024; Gao et al., 2024). In practice, unregularized self-play is also susceptible to over-optimization, particularly when the preference model is misspecified. While some approaches incorporate regularization, they are typically constrained to a reverse KL divergence penalty that restricts deviations from the reference policy (Munos et al., 2023; Zhang et al., 2024).

In this paper, we introduce a *generalized framework* for incorporating diverse regularization methods into self-play alignment, termed **Regularized Self-Play Policy Optimization (RSPO)**. RSPO offers a simple way to apply various regularization strategies in self-play by directly adding the regularization term to the loss function, while maintaining last-iterate convergence to the Nash Equilibrium of the corresponding regularized preference optimization game. Empirical analysis reveals distinct effects of different regularization methods: forward KL regularization reduces the response length in RSPO, whereas reverse KL regularization significantly enhances the raw win rate. Consequently, we adopt a linear combination of forward and reverse KL divergences, yielding a substantial improvement over the unregularized self-play alignment method, SPPO (Wu et al., 2024), on various benchmarks. Particularly on AlpacaEval-2, RSPO outperforms SPPO **with a** 6.9% **increase in length-controlled win rate (LCWR) and an** 18% **LCWR improvement over the base model**, Mistral-7B-Instruct. Furthermore, we offer an analysis of response diversity that regularization also promotes greater diversity. In summary, regularization plays a crucial role in self-play alignment, significantly improving both the quality and diversity of responses in previously unregularized self-play methods.

2 Related Work

Azar et al. (2024) presents the first work on optimizing general preference models. Nash-MD (Munos et al., 2023) is the first approach to address general preference optimization with self-play, by formulating preference optimization as a two-player game. Subsequent methods either aims to learn the Nash Equilibrium (NE) of the original unregularized game Swamy et al. (2024); Wu et al. (2024); Rosset et al. (2024); Wang et al. (2024), or seek to incorporate only reverse KL regularization and solving the NE of a reverse-KL-regularized preference optimization game Munos et al. (2023); Calandriello et al. (2024); Zhang et al. (2024). In contrast, we explore the broad class of divergence-based regularization techniques for self-play alignment.

We highlight the distinction between our self-play approach and the self-play methods based on pairwise comparisons, which construct loss functions by leveraging the difference in policy logits between preferred and rejected responses (Rafailov et al., 2024; Calandriello et al., 2024). Direct Nash Optimization (Rosset et al., 2024) and Iterative Nash Policy Optimization (INPO) (Zhang et al., 2024) follow Mirror Descent (MD) update (Beck & Teboulle, 2003) while indirectly compute loss with pairwise comparisons. This pairwise-comparison-based loss as in Direct Policy Optimization (DPO) has shown merely increasing relative likelihood gap, which may not elevate the probability of the preferred response (Pal et al., 2024). Our methods, instead approximate the MD update directly, by converting MD to an RL problem.

Online iterative RLHF, incorporating a trustworthy reward or preference model—including self-play—serves as a self-improving framework by iteratively generating new data using models and optimizing policies based on this data (Schulman et al., 2017; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023; Dong et al., 2024). Additionally, extending powerful offline methods such as Direct Preference Optimization (DPO) to iterative procedures has demonstrated remarkable performance improvements (Xu et al., 2023; Liu et al., 2023; Tran et al., 2023; Dong et al., 2024; Calandriello et al., 2024; Pang et al., 2024; Xiong et al., 2024; Guo et al., 2024; Tajwar et al., 2024; Cen et al., 2024; Xie et al., 2024). While in this work, we study the general preference optimization with self-play from a game-theoretic perspective.

3 PRELIMINARIES

We denote a prompt as x, a response as y, and a LLM policy as $\pi(y|x)$, where $\pi(\cdot|x) \in \Delta_{\mathcal{Y}}$, $\mathcal{X}$ is the set of all prompts and $\mathcal{Y} = \{y^0, y^1, \cdots\}$ is the set of all responses. We denote the probability simplex over the responses given a specific prompt, as $\Delta_{\mathcal{Y}}$. We parametrize the LLM policy π as π_θ. The reference policy is an LLM denoted as $\mu \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$. For notational brevity, we remove the dependence of policy π and loss functions on the prompt x throughout the paper.

3.1 GAME-THEORETIC PREFERENCE OPTIMIZATION

We study the preference optimization problem in an online setting by formulating it as a two-player max-min game, as studied in previous self-play works (Munos et al., 2023; Wu et al., 2024). The players are two LLMs whose strategies are LLM policies, denoted as max-player π and min-player π'. The utility of the max-player is the preference:

$$u(\pi; \pi') = \mathbb{P}(\pi \succ \pi') \stackrel{\text{def}}{=} \mathbb{E}_{y\sim\pi, y'\sim\pi'}[\mathbb{P}(y \succ y')], \tag{1}$$

where $u : \Delta_{\mathcal{Y}}^{\mathcal{X}} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \to \mathbb{R}$ is *linear* in π and π'; $\mathbb{P} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is a general preference model that quantifies the preference of y over y' given a prompt as in (Munos et al., 2023; Wu et al., 2024; Zhang et al., 2024). We extend the notation $\mathbb{P}(y \succ \pi') = \mathbb{E}_{y'\sim\pi'}[\mathbb{P}(y \succ y')]$ for any response y. The objective is finding a *Nash Equilibrium (NE)* policy π^* of the preference model:

$$(\pi^*, \pi^*) = \arg\max_{\pi} \min_{\pi'} \mathbb{P}(\pi \succ \pi'). \tag{2}$$

Therefore, an NE strategy π^* is an LLM that can generate *the most preferred responses in expectation*, thus achieving human alignment based on the preference model.

Existing game-theoretic self-play methods solve this NE following Algorithm 1 (Wu et al., 2024; Swamy et al., 2024; Zhang et al., 2024; Wang et al., 2024). Specifically, the policy is first initialized as $\pi_0 = \mu$. Then in each iteration t, the opponent is set to be the last-iterate policy π_t (the reason why it's called self-play), and the responses are sampled from π_t (Line 4). The pairwise preferences of the sampled responses are collected using the preference model $\mathbb{P}$ (Line 5). The policy parameters are updated by minimizing a specified loss function $\mathcal{L}(\theta; \mathbb{P})$ based on preferences over responses (Line 6). The loss function $\mathcal{L}(\theta; \mathbb{P})$ is dependent on the inherent online learning method.

3.2 PREFERENCE OPTIMIZATION VIA MULTIPLICATIVE WEIGHTS UPDATE

An effective self-play method to solve the preference optimization game in Equation (2) is Self-Play Policy Optimization (Wu et al., 2024). SPPO derives its loss function from the no-regret learning algorithm, Multiplicative Weights Update (MWU) (Freund & Schapire, 1997). Specifically in a game setting, denote learning rate as η, and normalization constant $Z(\pi_t)$. In iteration t, the policy update $\forall y \in \mathcal{Y}$ is

$$\pi_{t+1}(y) = \pi_t(y) \cdot \frac{\exp\Big(\eta \mathbb{E}_{y'\sim\pi_t}[u(y; y')]\Big)}{Z(\pi_t)}, \tag{3}$$

where $u(y; y')$ is the utility function defined in Equation equation 1.

The practical loss function of SPPO for policy update in each iteration t is the square error between LHS and RHS in Equation (3) at a logarithmic scale,

$$\mathcal{L}_{\text{SPPO}}(\theta) = \mathbb{E}_{y\sim\pi_t}\Big[\log \frac{\pi_\theta(y)}{\pi_t(y)} - \Big(\eta \mathbb{P}(y \succ \pi_t) - \log Z(\pi_t)\Big)\Big]^2. \tag{4}$$

SPPO converges to the NE of the preference optimization game. However, after multiple iterations training, the deviation of the policy π_θ from μ can be large. Such deviation is particularly problematic when the preference model is only accurate at evaluating responses sampled from the reference policy (Munos et al., 2023). Furthermore, in aligning LLMs in practice, the preference model is typically a surrogate $\hat{\mathbb{P}}$, such as PairRM (Jiang et al., 2023b), which may be misspecified at some out-of-distribution responses and inaccurate due to estimation error or limited model expressiveness (e.g., PairRM is only a 0.4B model), causing over-optimization problem. Regularizing the policy optimization to a reference SFT model, which is typically trained on high-quality data (Ouyang et al., 2022), can mitigate the problem. We provide a synthetic example in C.1 to demonstrate the problem.

3.3 Regularized Preference Optimization Game with Reference Policy

To address the regularization in self-play, we adopt the objective in Nash Learning from Human Feedback (Munos et al., 2023), and extend the KL divergence regularization to a general regularization function, to penalize the deviation from reference policy. We define a *convex* regularization function $R : \Delta_{\mathcal{Y}}^{\mathcal{X}} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \to (-\infty, \infty)$, where $R(\pi, \mu)$ measures the distance between π and the reference model μ, such as KL divergence $D_{\text{KL}}(\pi \| \mu)$. Denote regularization temperature as τ, the objective becomes to optimize a *regularized preference model* by solving the Nash Equilibrium (π^*, π^*) of the *regularized* game, where the utility of max player is still $u(\pi; \pi') = \mathbb{P}(\pi \succ \pi')$:

$$\arg\max_{\pi} \min_{\pi'} \mathbb{P}(\pi \succ \pi') - \tau R(\pi, \mu) + \tau R(\pi', \mu). \tag{5}$$

We provide the proof of the existence of this Nash Equilibrium in Appendix A.2. Various methods leverage Mirror Descent (MD) to find a regularized NE in Equation (5) (Munos et al., 2023; Calandriello et al., 2024; Zhang et al., 2024; Wang et al., 2024), based on its last-iterate convergence.

However, these MD-based methods have regularizer limited to a reverse KL divergence. Nash-MD[1] addresses the reverse KL regularization of π and μ using a geometric mixture policy π_t^μ (Munos et al., 2023):

$$\pi_{t+1} = \arg\min_{\pi} -\eta \langle \pi, \partial_\pi u(\pi_t; \pi_t^\mu) \rangle + D_{\text{KL}}(\pi, \pi_t^\mu). \tag{6}$$

While the LLMs optimized via self-play exhibit significant improvement (Wu et al., 2024; Wang et al., 2024; Zhang et al., 2024), they all have limited regularization of π and μ. They either completely lack explicit regularization, or only employing reverse KL divergence, imposing only a narrow form of regularization. The potential benefits of alternative regularization, such as adopting other f-divergences than reverse KL, remain unexplored.

4 Regularized Self-Play Policy Optimization

We propose a generalized framework of applying different regularization methods for self-play algorithms, called **Regularized Self-Play Policy Optimization (RSPO)**. In Section 4.1, we propose a novel no-regret learning method, Generalized Magnetic Mirror Descent, as the theoretical foundation of RSPO. In Section 4.2, we introduce our novel RSPO framework, and introduce our implementation of regularization methods in Section B.3. Finally, in Section 4.3, we demonstrate the novel connections of RSPO to existing self-play methods.

4.1 Generalized Magnetic Mirror Descent

We propose Generalized Magnetic Mirror Descent (GMMD) extended from Magnetic Mirror Descent Sokota et al. (2022), to solve a *regularized* max-min game. Denote the utility function of the game as U. We denote G as the element of the gradient vector of U:

$$\partial_\pi U(\pi; \pi') = \left(G(y^0; \pi'), \cdots, G(y^{|\mathcal{Y}|}; \pi')\right)^\top \in \mathbb{R}^{|\mathcal{Y}|}. \tag{7}$$

In iteration t, GMMD updates policy as

$$\pi_{t+1} = \arg\min_{\pi} -\eta \mathbb{E}_\pi[G(y; \pi_t)] + B_\psi(\pi; \pi_t) + \tau R(\pi, \mu), \tag{8}$$

where τ is regularization temperature, R is a general regularization function, serving as a "magnet" to attract π to μ during policy updating. B_ψ is the Bregman Divergence generated by a convex potential function ψ (Bregman, 1967).

Notably, the vanilla Magnetic Mirror Descent limits R to be the same regularization method of π and π_t, i.e., $B_\psi(\pi; \pi_t)$ (Sokota et al., 2022, Section 3.2); whereas in this paper we aim at a general regularizer of π and μ, which could be different from B_ψ, and study the effects of different regularizations methods.

[1] We call regularization by default meaning the one between π and μ, which is more important in preference optimization.

Proposition 4.1 (**Last-iterate Convergence**). *If $R(\cdot, \mu)$ is 1-strongly convex relative to ψ, $\eta \leq \tau$, and U is linear, then policy updated by GMMD in Equation* (8) *has last-iterate convergence to the following regularized Nash Equilibrium:*

$$\max_{\pi} \min_{\pi'} U(\pi; \pi') - \tau R(\pi, \mu) + \tau R(\pi', \mu). \tag{9}$$

Proposition 4.1 is a direct application of Theorem 3.4. by Sokota et al. (2022). We provide the proof in Appendix A.3. Proposition 4.1 guarantees the last-iterate convergence to the Nash Equilibrium of a regularized game.

To adapt GMMD to preference optimization problems, RL techniques are commonly employed as practical implementations of MD (Munos et al., 2023; Wang et al., 2024). Define the loss function of conducting GMMD in preference optimization as

$$\mathcal{L}_{\text{GMMD}}(\theta) \stackrel{\text{def}}{=} -\eta \mathbb{E}_{\pi_\theta}\big[G(y; \pi_t)\big] + D_{\text{KL}}(\pi_\theta || \pi_t) + \tau R(\pi_\theta, \mu). \tag{10}$$

Here, we set the Bregman divergence to Reverse KL in preference optimization as in previous works (Munos et al., 2023; Zhang et al., 2024). The gradient estimation of $\mathcal{L}_{\text{GMMD}}(\theta)$ for policy updates is required since the expectation in the first term is dependent on π_θ. Following Policy Gradient theorem (Sutton et al., 1999), then we have

$$\nabla_\theta \mathcal{L}_{\text{GMMD}}(\theta) = \mathbb{E}_{y \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(y) \left(- \eta G(y; \pi_t) + \log \frac{\pi_\theta(y)}{\pi_t(y)} + B \right)\right] + \tau \nabla_\theta R(\pi_\theta, \mu), \tag{11}$$

where B is a baseline function to reduce the variance as in REINFORCE (Williams, 1992). We set B independent to θ so that adding B won't change the value of Equation (10), due to $\mathbb{E}_{y \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(y) \cdot \eta B] = \eta B \nabla_\theta \mathbb{E}_{y \sim \pi_\theta}[1] = 0$.

We follow SPPO to replace the samples $y \sim \pi_\theta$ with $y \sim \pi_t$ directly since they are equivalent while computing the loss before updating, and rewrite the loss equivalent to GMMD:

$$\nabla_\theta \mathcal{L}_{\text{GMMD}}(\theta) = \nabla_\theta \left(\frac{1}{2} \mathbb{E}_{y \sim \pi_t}\Big[- \eta G(y; \pi_t) + \log \frac{\pi_\theta(y)}{\pi_t(y)} + \eta B \Big]^2 + \tau R(\pi_\theta, \mu) \right). \tag{12}$$

4.2 RSPO Framework

Based on the loss equivalent to Generalized Magnetic Mirror Descent (GMMD) in Equation (12), we propose **Regularized Self-Play Policy Optimization (RSPO)**. The loss function of RSPO $\mathcal{L}_{\text{RSPO}}(\theta; G, B, R)$ is defined as

$$\mathcal{L}_{\text{RSPO}}(\theta; G, B, R) = \mathbb{E}_{y \sim \pi_t}\Big[\log \frac{\pi_\theta(y)}{\pi_t(y)} - \eta \Big(G(y, \pi_t, \mu) - B(\pi_t, \mu) \Big) \Big]^2 + \lambda R(\pi_\theta, \mu) \,. \tag{13}$$

Here we call the first Mean Square Error term a self-play loss function $\mathcal{L}_{\text{SP}}$. $G : \mathcal{Y} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \to (-\infty, \infty)$ defines the *update direction* of π_θ, which can be set as the gradient of a utility function to guide the iterative optimization; The *baseline* function $B : \Delta_{\mathcal{Y}}^{\mathcal{X}} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \to (-\infty, \infty)$ is for variance-reduction for G similar to the baseline in REINFORCE; $R : \Delta_{\mathcal{Y}}^{\mathcal{X}} \times \Delta_{\mathcal{Y}}^{\mathcal{X}} \to \mathbb{R}$ is the regularization function, λ is the regularization temperature.

The loss function of RSPO comprises a quadratic self-play loss $\mathcal{L}_{\text{SP}}$ and an *external regularization* R. RSPO serves as a generalized framework, offering flexibility in incorporating different regularization methods into self-play-based preference optimization methods. The expectation term in Equation (13) can be interpreted as a loss function facilitating exponentiated gradient descent (Beck & Teboulle, 2003). The subsequent regularization term R provides a flexible mechanism for integrating different regularization functions by simply adding it to the self-play loss.

Besides the flexibility, by setting the update direction of RSPO as the gradient of the preference against π_t, $\forall y \in \mathcal{Y}$:

$$G(y, \pi_t, \mu) = \partial_{\pi(y)} \mathbb{P}(\pi \succ \pi_t) = \mathbb{P}(y \succ \pi_t), \tag{14}$$

RSPO is theoretically guaranteed to solve the regularized preference optimization game in Equation (5). Specifically, we execute Algorithm 1 by applying the following RSPO loss to approximate the GMMD:

$$\begin{aligned}&\mathcal{L}_{\text{RSPO}}\big(\theta; G=\mathbb{P}(y\succ\pi_t), B=\tfrac{1}{2}, R\big)\\ =&\mathbb{E}_{y\sim\pi_t}\Big[\log\frac{\pi_\theta(y)}{\pi_t(y)}-\eta\Big(\mathbb{P}(y\succ\pi_t)-\tfrac{1}{2}\Big)\Big]^2+\lambda R(\pi_\theta,\mu).\end{aligned}\tag{15}$$

Proposition 4.2. *Self-play following Algorithm 1 with the RSPO loss function in Equation* (15) *and regularizer R satisfying the assumption in Proposition 4.1, has last-iterate convergence to the Nash Equilibrium of the regularized preference optimization game, as described in Equation* (5)*.*

We provide the proof details in Appendix A.4. Here, we set $B=\frac{1}{2}$ following Nash-MD and SPPO. In theory, B helps minimize the variance of G the most when $B=\mathbb{E}_{y\sim\pi_t}[G(y,\pi_t,\mu)]$. But in preference optimization, due to the typically small minibatch size, estimation error of the mean of G could be large, leading to additional estimation error of the loss. Thus, we also set the baseline value for variance reduction to be a constant $\frac{1}{2}$, the mean value of G when the algorithm is converged.

Apart from the flexibility and simplicity of applying different regularization methods, RSPO can generalize existing self-play methods including the unregularized ones, which enables regularizing off-the-shelf self-play methods in practice with *no change* on their original loss functions or hyperparameters, directly adding external regularization term to their loss functions.

4.3 Generalizing Existing Self-Play Methods

We show that existing methods have loss functions equivalent to the special case of the quadratic self-play loss $\mathcal{L}_{\text{SP}}$, i.e., RSPO without external regularization: $\mathcal{L}_{\text{RSPO}}(R=0)$.

Unregularized self-paly method SPPO (Wu et al., 2024), has loss function in Equation (4) exactly in the form of $\mathcal{L}_{\text{SP}}$:

$$\mathcal{L}_{\text{SPPO}}(\theta)=\mathcal{L}_{\text{SP}}\bigg(\theta; G=\mathbb{P}(y\succ\pi_t), B=\frac{1}{2}\bigg).\tag{16}$$

Other unregularized self-play methods following the preference-based exponential update in Equation (3) can also be generalized by $\mathcal{L}_{\text{SP}}$, and thus can be regularized by simply adding regularization term to the loss functions. SPO (Swamy et al., 2024), based on the same exponential update rule as in SPPO, is equivalent to be updated via $\mathcal{L}_{\text{SP}}$ in Equation (16). Magnetic Policy Optimization (Wang et al., 2024) though has regularization in the policy update, periodically update $\mu=\pi_t$. Thus, it's inherently still conducting Equation (3) but incorporating multiple policy updates in each iteration following (Tomar et al., 2020).

In addition, even existing regularized methods can be generalized by $\mathcal{L}_{\text{SP}}$. Mirror Descent methods including Online Mirror Descent and Nash-MD have direct connection to RSPO and $\mathcal{L}_{\text{SP}}$ due to the same basic update rule (derivations provided in Appendix A.1).

$$\nabla_\theta\mathcal{L}_{\text{Nash-MD}}(\theta)=\nabla_\theta\mathcal{L}_{\text{SP}}\bigg(\theta; G=\mathbb{P}(y\succ\pi_t^\mu)-\tau\log\frac{\pi_t(y)}{\mu(y)}, B=\frac{1}{2}\bigg).\tag{17}$$

Therefore, our generalized loss framework RSPO enables to even add extra regularization to existing regularized self-play methods, while maintaining the convergence to Nash Equilibrium of the corresponding regularized game. We summarize how RSPO generalize existing self-play methods in Table 3.

Comparisons to Existing Methods. RSPO is more efficient for regularization in self-play, which requires *no change* on existing self-play loss nor their hyperparameters. RSPO is flexible for users to apply different divergences for regularization by simply changing an additive regularization term R to the loss function and tuning the single additional hyper-parameter λ. While existing regularized self-play methods are limited to the reverse KL divergence for regularization. Incorporating with regularization (e.g. from SPPO to Nash-MD) requires significant changes.

5 Experiments

In this section of experiments, we answer the following questions:

Methods (Base Model: Mistral-7B-Instruct)	**AlpacaEval-2 LCWR**	**Arena-Hard Auto-v0.1**	**MT-Bench**
Mistral-7B-Instruct	17.1	12.6	7.51
Snorkel (PairRM-Iterative-DPO)	26.4	20.7	7.58
SPPO Iter3	28.5	19.2	7.59
SimPO	32.1	21.0	7.60
RSPO (IS-For.+Rev.) Iter3	**35.4**	**22.9**	**7.75**

Table 1: Performance of existing methods, and our strongest model RSPO with Importance-Sampling-based Forward KL ($\lambda_1 = 0.1$) + Reverse KL ($\lambda_2 = 0.5$) divergence as regularization, on AlpacaEval-2 and Arena-Hard-Auto-v0.1.

- Does regularization improve the performance of self-play? (Sec. 5.1).
- Which regularization method is the most effective in self-play? (Sec. 5.2).
- What additional advantages can be derived from utilizing regularization in self-play? (Sec. 5.3).

Experiment Setup. We investigate our methods mainly on benchmarks AlpacaEval (Dubois et al., 2024), Arena-Hard (Li et al., 2024), and MT-Bench (Zheng et al., 2023). We follow the experiment setup of SPPO and Snorkel-Mistral-PairRM-DPO (Snorkel) (Tran et al., 2023) to examine our regularization methods, where Snorkel is based on iterative DPO and has achieved strong performance on AlpacaEval. Our *reference policy* model is Mistral-7B-Instruct-v0.2. Since iterative self-play methods require no response data for training, we only use the the *prompts of the Ultrafeedback dataset* (Cui et al., 2023), whose size is $\sim$ 60K. Following SPPO and Snorkel, we also split the prompts into three subsets and use only one subset per iteration to prevent over-fitting. To understand the later-iterate performance of self-play, in section 5.1, we also train on single fold of the prompts iteratively. We use a 0.4B response-pair-wise *preference model* PairRM (Jiang et al., 2023b), evaluated as comparable to $10\times$ larger reward/preference models (Cui et al., 2023). We investigate the effect of regularization mainly via AlpacaEval-2.0, where the main metric is length-controlled win rate (LCWR).

Implementations and Baselines. The implementation of self-play methods follows Algorithm 1. In each iteration, given response-pair-wise preference from PairRM and $K = 5$ number of response samples from the current policy, we estimate the policies' preference $\mathbb{P}(\pi \succ \pi_t)$ and regularization via Monte-Carlo estimation to compute the loss function. We replicate the SPPO with the default hyper-parameters and extend to 9 iterations. We implement RSPO as described in Theorem 4.2. The implementation of regularizations in RSPO are demonstrated in Appendix B.3 using the K samples. We report some of the baseline results from the previous papers, including SPPO, Snorkel (Mistral-PairRM-DPO) (Tran et al., 2023), Mistral-7B (Instruct-v0.2) (Jiang et al., 2023a), iterative DPO by Wu et al. (2024), and SimPO Meng et al. (2024). Since SPPO paper only provides results across 3 iterations (Wu et al., 2024), we replicate SPPO as an important baseline to study the performance across more than 3 iterations.

5.1 Effectiveness of Regularization

In this section, we assess the effectiveness of regularization primarily by comparing the performance of unregularized and regularized self-play methods. We first examine the over-optimization issue inherent in practical self-play preference optimization by extending the execution of SPPO to iteration 9. As depicted in Figure 2, a decline in performance appears during the later iterations of SPPO. We hypothesize that this behavior arises from the practical challenges associated with a misspecified preference model, as the signals driving policy updates in SPPO rely only on the preference model.

In Table 2, we further contrast the unregularized self-play method, SPPO and other iterative methods, with the best RSPO, namely RSPO (For.+Rev.). The regularization is a linear combination of Forward KL and Reverse KL divergence with coefficients 0.1 and 0.5, respectively. The comparative results reveal that regularization enhances the SPPO win rate from 31.02% to 38.31%, and the LC win rate increases from 28.53% to 35.44% in iteration 3. Notably, in the first iteration, reg. SPPO exhibits a slightly lower LC win rate, potentially attributable to the influence of strong reg-

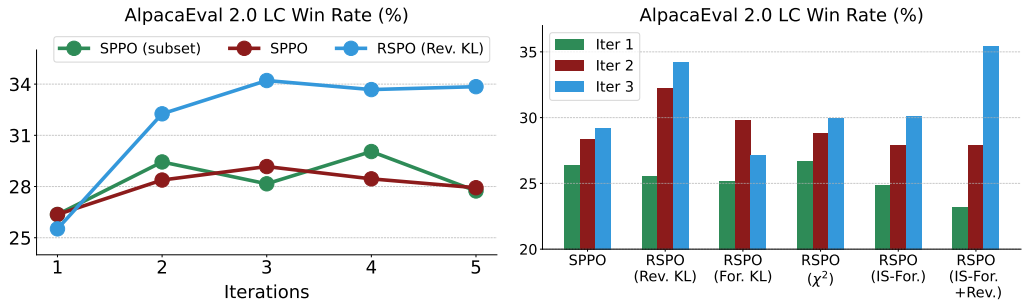

Figure 2: **Left:** LC win rate across iterations for standard SPPO, SPPO trained on a subset of the data (SPPO (subset)), and reverse-KL-regularized SPPO (SPPO (Rev. KL)). The base-model is Mistrial-7B. SPPO starts to degrade after 3 iterations. **Right:** LC win rate of SPPO and RSPO with different regularization methods. From left to right regularization methods: Reverse KL ($\lambda = 0.5$), Forward KL ($\lambda = 1.0$), Chi-Squared ($\lambda = 0.1$), Importance-Sampling Forward KL ($\lambda = 0.1$), Forward and Reverse KL linear combination ($\lambda_1 = 0.1$, $\lambda_2 = 0.5$).

Model	AlpacaEval 2.0 LC Win Rate	AlpacaEval 2.0 Win Rate	AlpacaEval 2.0 Avg. Len
Mistral-7B	17.11	14.72	1676
Snorkel	26.39	30.22	2736
SimPO	32.1	34.8	2193
DPO Iter1	23.81	20.44	1723
DPO Iter2	24.23	24.46	2028
DPO Iter3	22.30	23.39	2189
SPPO Iter1	24.79	23.51	1855
SPPO Iter2	26.89	27.62	2019
SPPO Iter3	28.53	31.02	2163
SPPO ≤ 9	29.17	29.75	2051
RSPO Iter1	23.16	21.06	1763
RSPO Iter2	27.91	27.38	1992
RSPO Iter3	**35.44**	**38.31**	2286

Regularization	Iteration	AlpacaEval 2.0 LCWR ↑	AlpacaEval 2.0 Self-BLEU ↓
×	1	24.79	0.751
	2	26.89	0.754
	3	28.53	0.758
IS-Forward KL + Reverse KL	1	23.16	0.747
	2	27.91	0.743
	3	**35.44**	0.714
Reverse KL	1	25.52	0.747
	2	32.26	0.730
	3	34.21	**0.691**
IS-Forward KL	1	24.88	0.756
	2	27.9	0.759
	3	30.09	0.760
χ^2	1	26.7	0.745
	2	28.78	0.740
	3	29.97	0.739

Table 2: **Left**: Comparisons of iterative methods with reference models Mistral-7B (Instruct-v0.2). SPPO ≤ 9 represents the best results among the 9 iterations of SPPO. Here the Regularized SPPO (RSPO) is regularized by the *linear combination of Forward KL and Reverse KL divergence*, i.e. RSPO (For. + Rev.), where the regularization temperatures are 0.1 and 0.5, respectively. **Right:** Response diversity of SPPO with different regularization methods using Self-BLEU score. The regularization temperatures are the same as in Figure 2 (Right). Lower Self-BLEU score means higher diversity of the sampled responses. Regularization methods involving Reverse KL resulted in higher diversity of the responses.

ularization. However, subsequent iterations show a marked improvement, with the LC win rate of reg. SPPO increasing by up to 7.53% within a single iteration. In summary, the findings in Table 2 underscore the effectiveness of regularization in self-play optimization.

In addition, to exclude the possibility of insufficient iterations, we report the the best result among 9 iterations of our replicated SPPO in Table 2, denoted as "SPPO $\leq$ 9". SPPO $\leq$ 9 consistently underperforms the RSPO result at iteration 3. These observations emphasize that even extended training under the unregularized framework fails to match the performance gains achieved through regularization, thereby affirming the critical role of regularization in self-play methodologies for preference optimization.

5.2 Impact of Different Regularizations

We then study the effect of applying different regularization R in RSPO. To obtain a well-regularized self-play, the tuning of regularization temperature λ is necessary. An ablation study of regularization temperature of different methods is shown in Figure 3. According to the figure, the response length is increased along with the temperature in reverse KL divergence and Chi-square divergence

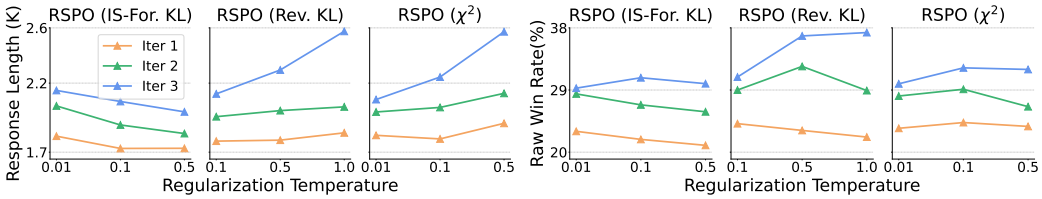

Figure 3: Ablation Study of regularization temperature λ conducted on AlpacaEval 2.0. We evaluate how the average response length and raw WR are affected by the regularization temperature.

regularized RSPO. While, the length is decreased with stronger regularization via Forward KL divergence, implemented using importance sampling. This result underscores the distinct effects of different regularization strategies. In particular, the raw win rate analysis highlights reverse KL divergence as a crucial factor in enhancing self-play performance. Given that forward KL divergence tends to reduce response length while reverse KL divergence yields significant improvements, we adopt a linear combination of both. This approach is designed to balance their complementary effects, ultimately optimizing for a higher LCWR (RSPO (IS-For. + Rev.) in Figure 2 RHS).

In Figure 2 (Right), we show the results of win rate and LCWR in AlpacaEval 2.0 of different regularizations. Only vanilla Forward KL decreases the win rate of SPPO. The regularizations that consists of Reverse KL including RSPO (Rev. KL) and RSPO (For.+Rev.) have shown significant improvement in win rates compared to vanilla SPPO. In particular, the results of RSPO (For.+Rev.) demonstrates the largest improvement between iterations, achieving the best LCWR.

We study the effect of applying different regularization R in RSPO. In Figure 2 (Right), we show the results of win rate and average response length on AlpacaEval 2.0. Among different regularizations, only Forward KL decreases the win rate of SPPO. The regularizations that consist of Reverse KL including RSPO (Rev. KL) and RSPO (For.+Rev.) have shown significant improvement in win rates compared to vanilla SPPO. In particular, the results of RSPO (For.+Rev.) demonstrates the largest improvement between iterations. We test the best RSPO model on different benchmarks [2] in Table 1.

5.3 Response Diversity

We demonstrate an additional advantage introduced by regularization, the diversity of the response. We provide a motivating example in Appendix C.2. We investigate it by estimating the diversity of the generations from trained models. We use Self-BLEU (Zhu et al., 2018) score to measure the diversity of the responses, where lower score implies higher response diversity. We take the first 200 tokens of each of the 16 generated responses using the prompts of AlpacaEval.

The trend of Self-BLEU scores presented in Table 2 (Right) show that applying Reverse KL to SPPO increases response diversity the most, as well as the LCWRs of AlpacaEval 2.0. Application of Forward KL results in slightly less generation diversity than vanilla SPPO, while they still achieve better win rates. The win rates are the highest when Forward KL and Reverse KL are linearly combined for regularization, while the Self-BLEU scores imply that the response diversity is lower than when only Reverse KL is applied. These results highlight that applying regularization in self-play methods can improve test performance and the diversity of the generations simultaneously.

6 Conclusion

In this paper, we study the regularization in self-play by proposing a framework, namely Regularized Self-Play Policy Optimization (RSPO). Based on RSPO, we can apply different regularization function for policy update by adding the regularization term to the loss functions, which is still guaranteed to converge to the Nash Equilibrium of the regularized Preference Optimization Game. In the empirical assessments, we achieve significant improvement over the base model and unregularized self-play method, SPPO. We also empirically demonstrate that regularization promotes the response diversity. These findings underscore the critical role of regularization as a fundamental component in optimizing self-play alignment.

[2] We report our replicated testing of SPPO Iter3 (https://huggingface.co/UCLA-AGI/Mistral7B-PairRM-SPPO-Iter3) on Arena-Hard, so it can be different from the result presented in SPPO (Wu et al., 2024).

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

Shicong Cen, Jincheng Mei, Katayoon Goshvadi, Hanjun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yuejie Chi, and Bo Dai. Value-incentivized preference optimization: A unified approach to online and offline rlhf. *arXiv preprint arXiv:2405.19320*, 2024.

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

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

Herbert Aron David. *The method of paired comparisons*, volume 12. London, 1963.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,

Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

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

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

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

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

Zhaolin Gao, Jonathan D Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J Andrew Bagnell, Jason D Lee, and Wen Sun. Rebel: Reinforcement learning via regressing relative rewards. *arXiv preprint arXiv:2404.16767*, 2024.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*, 2016.

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J Foster. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *arXiv preprint arXiv:2407.13399*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023b.

Lingjie Jiang, Shaohan Huang, Xun Wu, and Furu Wei. Textual aesthetics in large language models. *arXiv preprint arXiv:2411.02930*, 2024.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

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

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

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

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

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

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

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Paul Rubenstein, Olivier Bousquet, Josip Djolonga, Carlos Riquelme, and Ilya O Tolstikhin. Practical and consistent estimation of f-divergences. *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Maurice Sion. On general minimax theorems. 1958.

Samuel Sokota, Ryan D'Orazio, J Zico Kolter, Nicolas Loizou, Marc Lanctot, Ioannis Mitliagkas, Noam Brown, and Christian Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. *arXiv preprint arXiv:2206.05825*, 2022.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

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

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

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Hoang Tran, Chris Glaze, and Braden Hancock. Iterative dpo alignment. Technical report, Snorkel AI, 2023.

Mingzhi Wang, Chengdong Ma, Qizhi Chen, Linjian Meng, Yang Han, Jiancong Xiao, Zhaowei Zhang, Jing Huo, Weijie J Su, and Yaodong Yang. Magnetic preference optimization: Achieving last-iterate convergence for language models alignment. *arXiv preprint arXiv:2410.16714*, 2024.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

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

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*, 2024.

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

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

Yuheng Zhang, Dian Yu, Baolin Peng, Linfeng Song, Ye Tian, Mingyue Huo, Nan Jiang, Haitao Mi, and Dong Yu. Iterative nash policy optimization: Aligning llms with general preferences via no-regret learning, 2024. URL https://arxiv.org/abs/2407.00617.

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

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 1097–1100, 2018.

A PROOFS

In this section, we provide detailed derivations, proofs of propositions, and corollaries.

A.1 PROOF OF EQUIVALENCE BETWEEN MD AND RSPO

In this section, we first provide derivations of Nash-MD and Online Mirror Descent (Munos et al., 2023) to $\mathcal{L}_{\text{SP}}$.

Nash-MD . Nash-MD practical loss satisfies that

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{Nash-MD}}(\theta) &\stackrel{\text{def}}{=} \mathbb{E}_{\substack{y\sim\pi_\theta,\\ y'\sim\pi_t^\mu}}\Big[\nabla_\theta \log \pi_\theta(y)\Big(\mathbb{P}(y \succ y') - \frac{1}{2} - \tau \log \frac{\pi_\theta(y)}{\mu(y)}\Big)\Big], \\
&= \mathbb{E}_{\substack{y\sim\pi_\theta,\\ y'\sim\pi_t^\mu}}\Big[\nabla_\theta \log \pi_\theta(y)\Big(\mathbb{P}(y \succ y') - \frac{1}{2} - \tau \log \frac{\pi_\theta(y)}{\pi_t(y)} - \tau \log \frac{\pi_t(y)}{\mu(y)}\Big)\Big] \\
&= \mathbb{E}_{y\sim\pi_\theta}\Big[\nabla_\theta \log \pi_\theta(y)\Big(\mathbb{P}(y \succ \pi_t^\mu) - \frac{1}{2} - \tau \log \frac{\pi_\theta(y)}{\pi_t(y)} - \tau \log \frac{\pi_t(y)}{\mu(y)}\Big)\Big] \\
&= \mathbb{E}_{y\sim\pi_t}\Big[\nabla_\theta \log \pi_\theta(y)\Big(\mathbb{P}(y \succ \pi_t^\mu) - \frac{1}{2} - \tau \log \frac{\pi_\theta(y)}{\pi_t(y)} - \tau \log \frac{\pi_t(y)}{\mu(y)}\Big)\Big] \\
&= \nabla_\theta \mathbb{E}_{y\sim\pi_t}\Big[\tau \log \frac{\pi_\theta(y)}{\pi_t(y)} - \Big(\mathbb{P}(y \succ \pi_t^\mu) - \tau \log \frac{\pi_t(y)}{\mu(y)} - \frac{1}{2}\Big)\Big]^2/2 \\
&= \tau^2 \nabla_\theta \mathbb{E}_{y\sim\pi_t}\Big[\log \frac{\pi_\theta(y)}{\pi_t(y)} - \frac{1}{\tau}\Big(\mathbb{P}(y \succ \pi_t^\mu) - \tau \log \frac{\pi_t(y)}{\mu(y)} - \frac{1}{2}\Big)\Big]^2/2. \qquad (18)
\end{aligned}$$

The first equation is according to Section 7 in Munos et al. (2023). The second equation holds by adding an subtracting the same element $\log \pi_t(y)$. The third equation holds due to $\mathbb{E}_{y'\sim\pi_t^\mu}[\mathbb{P}(y \succ y')] = \mathbb{P}(y \succ \pi_t^\mu)$. The fourth equation holds since in each iteration before updating while computing the loss, $y \sim \pi_\theta$ is equivalent to $y \sim \pi_t$.

The learning rate η is originally omitted in the paper (Munos et al., 2023). Here Nash-MD is generalized by $\mathcal{L}_{\text{SP}}$ with $\eta = \frac{1}{\tau}$.

Online Mirror Descent. OMD is to execute $\arg\max_\pi \eta \mathbb{E}_{y\sim\pi}\Big[\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}\Big] - \text{KL}(\pi, \pi_t)$. Therefore, the parametrized policy is updated by descending the negative gradient

$$\begin{aligned}
&- \nabla_\theta \eta \mathbb{E}_{y\sim\pi_\theta}\left[\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}\right] + D_{\text{KL}}(\pi_\theta, \pi_t) \\
&= -\nabla_\theta \eta \mathbb{E}_{y\sim\pi_\theta}\left[\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)} - \log \frac{\pi_\theta}{\pi_t}\right] \\
&= \eta \mathbb{E}_{y\sim\pi_\theta}\left[-\nabla_\theta \log \pi_\theta\Big(\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)} - \log \frac{\pi_\theta}{\pi_t}\Big)\right] \\
&= \frac{\eta}{2} \cdot \mathbb{E}_{y\sim\pi_\theta}\left[\nabla_\theta \Big(\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)} - \log \frac{\pi_\theta(y)}{\pi_t(y)}\Big)^2\right] \\
&= \frac{\eta}{2} \cdot \mathbb{E}_{y\sim\pi_t}\left[\nabla_\theta \log \frac{\pi_\theta(y)}{\pi_t(y)} - \Big(\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}\Big)\right]^2. \qquad (19)
\end{aligned}$$

The first equation holds due to the definition of D_{KL}. The second equation holds due to importance sampling.

Therefore, OMD can also be generalized by RSPO with $G = \mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}$ and without external regularization.

A.2 PROOF OF THE EXISTENCE OF NASH EQUILIBRIUM IN EQUATION (5)

We prove the existence of Nash Equilibrium in the regularized game in this section, largely following idea of proving the existence of KL regularized Nash Equilibrium by Munos et al. (2023).

Proof. Since the utility $u(\pi, \pi')$ is linear in π and π', and the regularization function is assumed to be convex (Assumption A.1), the regularized preference is concave in π and convex in π'. Therefore, the existence and the uniqueness of a regularized Nash Equilibrium in Equation (5) can be directly derived from minimax theorem (Sion, 1958). □

A.3 Proof of Proposition 4.1

Assumption A.1 (Relative Convexity w.r.t. ψ). *We assume the regularization function R of policy π is a 1-strongly convex relative to negative entropy function $\psi(\pi)$. In other words, $\forall \pi, \pi' \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$,*

$$\langle \partial_\pi R(\pi) - \partial_\pi R(\pi'), \pi - \pi' \rangle \geq \langle \partial_\pi \psi(\pi) - \partial_\pi \psi(\pi'), \pi - \pi' \rangle \tag{20}$$

If $R(\cdot, \mu)$ is 1-strongly convex relative to ψ, policy updated by GMMD in Equation (8) has last-iterate convergence to the following Nash Equilibrium of a regularized game:

$$\max_{\pi} \min_{\pi'} U(\pi; \pi') - \tau R(\pi, \mu) + \tau R(\pi', \mu). \tag{21}$$

Proof. According to Equation (8), GMMD is equivalent to the Algorithm 3.1 in Sokota et al. (2022):

$$z_{t+1} = \arg\min_{z \in \mathcal{Z}} \eta \left(\langle F(z_t), z \rangle + \alpha g(z) \right) + B_\psi(z; z_t), \tag{22}$$

where in our setting, $z = \pi$ is the LLM policy, $F(z_t) = -\nabla_\pi U(\pi; \pi_t)$ is the vector of negative partial derivatives of preference w.r.t. each component of π, $\alpha = \tau$, $g(z)$ is the regularizer $R(\pi)$, and we set $\psi(z) = z \log z$ to convert the Bregman divergence B_ψ to KL divergence. Here $U(\pi; \pi_t)$ is treated as a function of vector form of π, i.e., $[\pi^0 \ \pi^1 \ \cdots \ \pi^{|\mathcal{Y}|}]$, thus the gradient is a vector gradient where $\nabla_\pi U(\pi; \pi_t) = [\partial U/\partial \pi^0 \ \ \partial U/\partial \pi^1 \ \ \cdots \ \ \partial U/\partial \pi^{|\mathcal{Y}|}]$. We then show that in our setting the following assumptions are satisfied:

F satisfies that for $\mu > 0$ and any z, z', $\langle F(z) - F(z'), z - z' \rangle = 0$ since U is linear in π, and $F(z) - F(z') = -\nabla_\pi U(\pi; \pi_t) + \nabla_\pi U(\pi'; \pi_t) = 0$. Therefore, F is Monotone and L-smooth. According to Assumption A.1, g is 1-strongly convex relative to ψ, i.e., $g(z) \geq g(z') + \frac{g'(z)}{\psi'(z)}(\psi(z) - \psi(z'))$.

Given the assumptions above, according to the Theorem 3.4. in Sokota et al. (2022), the update rule defined in Equation (22) has a last-iterate convergence guarantee to a policy π^*, which is the solution to the variational inequality problem $\text{VI}(\Delta_{\mathcal{Y}}^{\mathcal{X}}, F + \alpha \nabla g)$, i.e., π^* satisfies

$$\begin{aligned} &\langle \nabla \big(-U(\pi; \pi^*) + \tau R(\pi, \mu) \big) \mid_{\pi = \pi^*}, \pi - \pi^* \rangle \geq 0, \quad \forall \pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}} \\ \Leftrightarrow \ &\langle \nabla \big(-U(\pi; \pi^*) + \tau R(\pi, \mu) - \tau R(\pi^*, \mu) \big) \mid_{\pi = \pi^*}, \pi - \pi^* \rangle \geq 0, \quad \forall \pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}}. \end{aligned} \tag{23}$$

Equation (23) indicates that moving from π^* towards any direction $\pi - \pi^*$ can not increase the value of the objective preference model $U(\pi; \pi^*) - \tau R(\pi, \mu) + \tau R(\pi^*, \mu)$ at the point of $\pi = \pi^*$, given the opponent is π^*. Therefore, by symmetry, π^* is the Nash Equilibrium of the regularized preference model:

$$\max_{\pi} \min_{\pi'} U(\pi; \pi') - \tau R(\pi, \mu) + \tau R(\pi', \mu). \tag{24}$$

□

A.4 Proof of Proposition 4.2

Proof. We prove that RSPO in Equation (15) is equivalent to GMMD up to multiplying a constant to the gradient, leading to a regularized Nash Equilibrium.

$$\nabla_\theta \mathcal{L}_{\text{RSPO}}(\theta; G = \mathbb{P}(y \succ \pi_t), B = \frac{1}{2}) \tag{25}$$

$$= \nabla_\theta \Big(\mathbb{E}_{y\sim\pi_t} \Big[\log \frac{\pi_\theta(y)}{\pi_t(y)} - \eta \Big(\mathbb{P}(y \succ \pi_t) - \frac{1}{2} \Big) \Big]^2 + \lambda R(\pi_\theta, \mu) \Big) \tag{26}$$

$$= \nabla_\theta \Big(\langle \pi_t, , \big(-\eta \partial_\pi \mathbb{P}(\pi \succ \pi_t) + \log \frac{\pi_\theta}{\pi_t} + B \big)^2 \rangle + \lambda R(\pi_\theta, \mu) \Big) \tag{27}$$

$$= 2 \Big(\nabla_\theta \mathbb{E}_{y\sim\pi_t} [\big(-\eta G(y, \pi_t) + \log \frac{\pi_\theta(y)}{\pi_t(y)} + B \big)^2] \cdot \frac{1}{2} + \tau \nabla_\theta R(\pi_\theta, \mu) \Big) \tag{28}$$

$$= 2 \nabla_\theta \mathcal{L}_{\text{GMMD}}(\theta). \tag{29}$$

- Equation (26) holds due to definition.
- Equation (27) holds by treating policy as a vector and rewrite the expectation in vector product form, and $\nabla_\pi \mathbb{P}(\pi \succ \pi_t) \mid_{\pi=\pi_t} \mid_{\pi=\pi_t} = [\mathbb{P}(y^0 \succ \pi_t) \quad \mathbb{P}(y^1 \succ \pi_t) \quad \cdots \quad \mathbb{P}(y^{|\mathcal{Y}|} \succ \pi_t)]^T$, where $y^0, y^1, \cdots, y^{\mathcal{Y}}$ represent all possible values of y.
- Equation (28) holds by rewriting the form of dot product as expectation.
- Equation (29) holds due to the equivalent loss form of GMMD in Equation (12).

Thus, according to Proposition 4.1, update following Algorithm 1 with the above loss function has last-iterate convergence to the Nash Equilibrium of the regularized preference optimization game in Equation (5) by setting $u(\pi; \pi') = \mathbb{P}(\pi \succ \pi')$.

□

A.5 Proof of Proposition B.1

Proof. π is parametrized by θ, $\nabla_\theta D_{\text{KL}}(\pi||\mu) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(y) - \log \mu(y)]^2/2$. This is because

$$\begin{aligned}
\nabla_\theta D_{\text{KL}}(\pi||\mu) &= \nabla_\theta \sum_y \pi_\theta(y) \cdot (\log \pi_\theta(y) - \log \mu(y)) \\
&= \sum_y \nabla_\theta \pi_\theta(y) \cdot (\log \pi_\theta(y) - \log \mu(y)) + \sum_y \nabla_\theta \pi_\theta(y) \\
&= \sum_y \pi_\theta(y) \frac{\nabla_\theta \pi_\theta(y)}{\pi_\theta(y)} \cdot (\log \pi_\theta(y) - \log \mu(y)) + \nabla_\theta \sum_y \pi_\theta(y) \\
&= \mathbb{E}_{\pi_\theta}[(\log \pi_\theta(y) - \log \mu(y)) \cdot \nabla_\theta (\log \pi_\theta(y) - \log \mu(y))] \\
&= \mathbb{E}_{\pi_\theta}[\nabla_\theta (\log \pi_\theta(y) - \log \mu(y))^2]/2.
\end{aligned} \tag{30}$$

□

A.6 Proof of Proposition B.2

Proof. π is parametrized by θ, then $\nabla_\theta D_{\text{KL}}(\mu||\pi) = \mathbb{E}_\mu[\nabla_\theta \frac{\mu(y)}{\pi_\theta(y)}]$ because

$$\begin{aligned}
\nabla_\theta D_{\text{KL}}(\mu||\pi) &= \nabla_\theta \sum_y \mu(y) \cdot (\log \mu(y) - \log \pi_\theta(y)) \\
&= -\sum_y \mu(y) \nabla_\theta \log \pi_\theta(y) \\
&= -\sum_y \pi_\theta(y) \frac{\mu(y)}{\pi_\theta(y)} \nabla_\theta \log \pi_\theta(y) \\
&= -\mathbb{E}_{\pi_\theta}\left[\frac{\mu(y)\nabla_\theta \log \pi_\theta(y)}{\pi_\theta(y)}\right] \\
&= -\mathbb{E}_{\pi_\theta}\left[\frac{\mu(y)\nabla_\theta \pi_\theta(y)}{\pi_\theta(y)^2}\right] \\
&= \nabla_\theta \mathbb{E}_{\pi_\theta}\left[\frac{\mu(y)}{\pi_\theta(y)}\right]. \qquad (31)
\end{aligned}$$

□

A.7 Proof of Proposition B.3

Proof. π is parametrized by θ, $\nabla_\theta D_{\chi^2}(\pi_\theta(y)||\mu(y)) = \mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(y)}{\mu(y)}\right]$ since

$$\begin{aligned}
D_{\chi^2}(\pi_\theta(y)||\mu(y)) &= \frac{1}{2}\sum \left(\frac{\pi_\theta(y)}{\mu(y)} - 1\right)^2 \mu(y) dy \\
&= \frac{1}{2}\sum \frac{\pi_\theta(y)^2 - 2\pi_\theta(y)\mu(y) + \mu(y)^2}{\mu(y)} dy \\
&= \frac{1}{2}\sum \frac{\pi_\theta(y)^2}{\mu(y)} dy + C \\
&= \frac{1}{2}\mathbb{E}_{\pi_\theta(y)}\left[\frac{\pi_\theta(y)}{\mu(y)}\right] + C, \qquad (32)
\end{aligned}$$

where C is independent to θ. □

B Additional Details

In this section, we provide additional details of this paper, including the algorithm describtions of self-play algignment methods, a summarizing table for generalizing existing methods, and our implementation of regularizations.

B.1 Self-Play Alignment Algorithm

Algorithm 1 Self-Play Alignment

1: **Input:** LLM π_θ, preference model $\mathbb{P}$, number of iterations T, reference policy μ, loss function for policy update $\mathcal{L}(\theta; \mathbb{P})$, sample size K.
2: **Initialize:** $\pi_0 = \mu$.
3: **for** $t \in [T]$ **do**
4: Sample prompts and responses: $x \sim \mathcal{X}$, $y_{1:K} \sim \pi_t$
5: Get pair-wise preferences $\mathbb{P}(y_i \succ y_j)$, $\forall i, j \in [K]$
6: Update policy parameters $\theta = \arg\min_\theta \mathcal{L}(\theta; \mathbb{P})$
7: $\pi_{t+1} = \pi_\theta$
8: **end for**
9: **Output:** Last-iterate policy π_T.

B.2 Generalizing Existing Methods

Loss	Update Direction (G)	Baseline (B)	Preference Model
$\mathcal{L}_{\text{SPPO}}$ (Wu et al., 2024)	$\mathbb{P}(y \succ \pi_t)$	0.5	$\mathbb{P}(y \succ y')$
$\mathcal{L}_{\text{OMD}}$ (Munos et al., 2023)	$\mathbb{P}(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}$	Est.	$\mathbb{P}_\tau(y \succ y')$
$\mathcal{L}_{\text{Nash-MD}}$ (Munos et al., 2023)	$\mathbb{P}^\mu(y \succ \pi_t) - \tau \log \frac{\pi_t(y)}{\mu(y)}$	0.5	$\mathbb{P}_\tau(y \succ y')$

Table 3: Self-play losses $\mathcal{L}_{\text{SP}}$ with different game-theoretic RLHF policy optimization methods. $\mathbb{P}^\mu(y \succ \pi_t) = \mathbb{P}(y \succ \pi_t^\mu)$, π_t^μ is the geometric mixture of π_t and μ. We abbreviate the estimated baseline that reduce the variance of G the most as est.. $\mathbb{P}_\tau(y \succ y') = \mathbb{P}(y \succ y') - \tau \log \frac{\pi_\theta(y)}{\mu(y)} + \tau \log \frac{\pi'(y')}{\mu(y')}$ is the regularized preference model.

B.3 Implementation of Regularization

In practice, accurately estimating the gradient of the regularizer is essential, as many commonly used divergence measures are defined as expectations over π_θ. The estimation of divergences has been extensively studied and widely applied in various domains (Rubenstein et al., 2019). While for completeness, in this section, we introduce the regularization methods investigated in this study, including Reverse KL, Forward KL, and Chi-Square Divergence.

We begin by deriving the estimation of the Reverse KL divergence based on the following proposition.

Proposition B.1. *Reverse KL divergence satisfies:*

$$\nabla_\theta D_{KL}(\pi_\theta || \mu) = \mathbb{E}_{y \sim \pi_\theta}[\nabla_\theta (\log \pi_\theta(y) - \log \mu(y))^2]. \tag{33}$$

Due to the equivalent gradient in Proposition B.1, we can estimate the divergence with $\mathbb{E}_{y \sim \pi_\theta}[(\log \pi_\theta(y) - \log \mu(y))^2]$.

We employ two distinct approaches to estimate the forward KL divergence. The first method utilizes importance sampling, referred to as IS-For. KL, and is derived based on the following proposition.

Proposition B.2. *The gradient of forward KL divergence satisfies that*

$$\nabla_\theta D_{KL}(\mu || \pi_\theta) = \mathbb{E}_{y \sim \pi_\theta}[\nabla_\theta \mu(y) / \pi_\theta(y)]. \tag{34}$$

Therefore, we can estimate the forward KL divergence by leveraging the expectation $\mathbb{E}_{y \sim \pi_\theta}[\mu(y)/\pi_\theta(y)]$ to estimate the forward KL. Notably, to mitigate the risk of gradient explosion, we apply gradient clipping with a maximum value of 10.

The second method for forward KL is a direct estimation of $D_{\text{KL}}(\mu || \pi_\theta)$. To achieve this, we re-sample responses from the reference policy μ using the same prompts from the training dataset, constructing a reference dataset. The KL divergence is then estimated directly based on its definition by uniformly drawing samples from this reference dataset. A key advantage of this approach is that it eliminates the need for importance sampling, as each policy update iteration only requires samples from π_t.

Similarly, we estimate the Chi-Square divergence using $\mathbb{E}_{y \sim \pi_\theta}[\pi_\theta(y)/\mu(y)]$, based on the following proposition. Due to the presence of the ratio term, Chi-Square divergence estimation also necessitates gradient clipping to prevent instability, for which we set a clip value of 10.

Proposition B.3. *Chi-Square divergence has gradient*

$$\nabla_\theta D_{\chi^2}(\pi_\theta || \mu) = \mathbb{E}_{y \sim \pi_\theta}\left[\nabla_\theta \pi_\theta(y) / \mu(y)\right]. \tag{35}$$

We also explore the linear combination of different regularization functions to leverage their complementary effects, as in offline RLHF (Huang et al., 2024). The previously established propositions for estimating divergences can still be used in the combined regularization method.

Apart from the flexibility and simplicity of applying different regularization methods, RSPO can generalize existing self-play methods including the unregularized ones, which enables regularizing off-the-shelf self-play methods in practice with *no change* on their original loss functions or hyperparameters, directly adding external regularization term to their loss functions.

C Additional Experiments

In this section, we provide additional experiments, including two synthetic motivating examples and additional results on language tasks.

C.1 Regularization in Game Solving

The regularization in preference model is not used in all game-theoretic self-play methods. Here we investigate the necessity of regularization and offer a motivating example in Figure 4, a saddle point solving problem $\min_x \max_y \frac{\alpha}{2}x^2 + (x-1)(y-1) - \frac{\alpha}{2}y^2$. There exists a reference point as the initial values of x and y. We assume that both reference point and the Nash Equilibrium (NE) of the surrogate preference model (Surrogate NE) are close to the original NE but on different sides of the original NE.

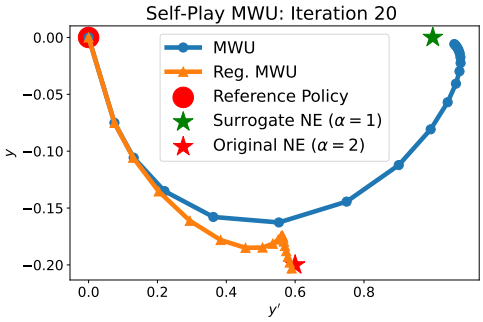

Figure 4: Motivating Example: 20 iterations of MWU and regularized MWU with the same learning rate to solve saddle point problem $\max_y \min_{y'} f(y, y', \alpha)$, where $f(y, y'; \alpha) = \frac{\alpha}{2}y'^2 + (y'-1)(y-1) - \frac{\alpha}{2}y^2$, first introduced in (Sokota et al., 2022). We assume that we only have access to a misspecified (surrogate) preference $f(y, y'; \alpha = 1)$, while the ground truth human preference is $f(y, y'; \alpha = 2)$. The dynamics show that regularization can be efficient to prevent over-optimization in self-play.

Typically, the surrogate preference/reward models are not positive related to the reference policy. Thus, it is a reasonable abstracted example of NLHF by treating reference point as reference policy and surrogate NE as the optimal policy obtained by optimizing the surrogate preference/reward. The results of the 20 iterations self-play MWU with early stopping show that regularization can be used to prevent reward over-optimization (reaching surrogate NE). A well-tuned regularization leads to faster convergence to the unknown original NE.

C.2 Diversity on 2D Example

We offer an analysis of our method compared to unregularized self-play (SPPO) on a 2D example in Figure 5. The area with darker color is assigned higher reward value. We use the preference defined by the L^2 norm between two actions. We also set the reference policy to be a uniform policy. According to the figure, unregularized method tends to converge to a single point on the manifold of the large reward. While regularized method have diverse sampled actions.

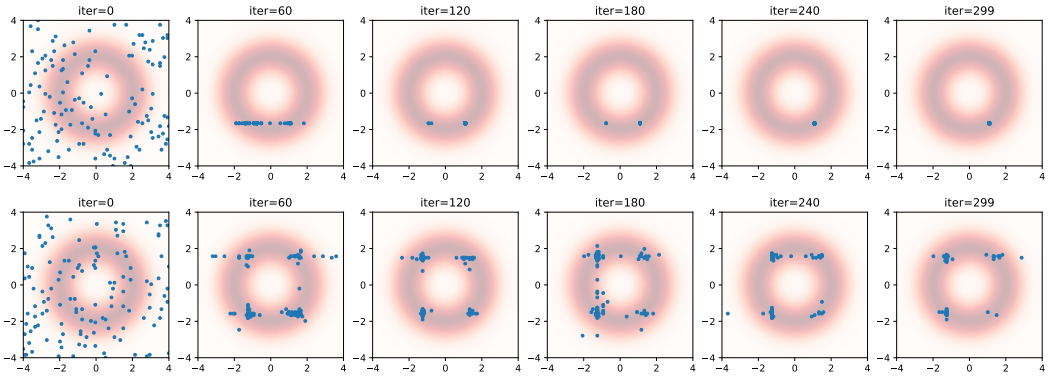

Figure 5: Samples in a 2D example of different iterations of SPPO (top) and RSPO (bottom) with external forward KL regularization. SPPO added simple external regularization can generate multi-modal policies.

C.3 More Results on AlpacaEval-2.0

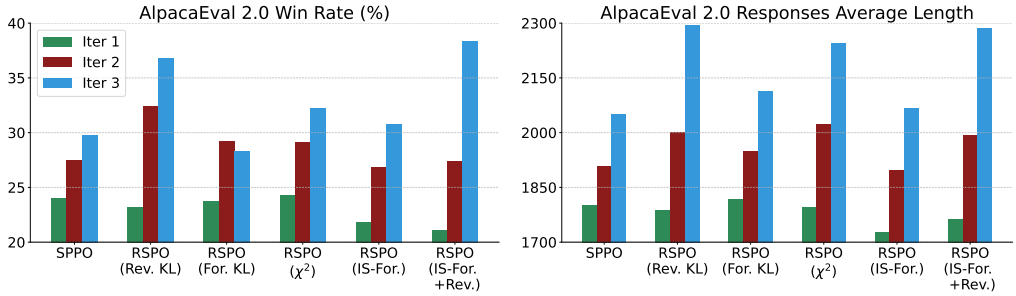

Figure 6: Win rates and average length of SPPO and RSPO with different regularization methods. From left to right regularization methods: Reverse KL, Forward KL, Chi-Squared, Importance-Sampling Forward KL, Importance-Sampling Forward and Reverse KL linear combination.