IRIS: Interpolative Rényi Iterative Self-play for Large Language Model Fine-Tuning

Wenjie Liao[1]* **Like Wu**[1] **Liangjie Zhao**[2] **Shihui Xu**[1] **Shigeru Fujimura**[1]

[1]Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan
[2]Institute of Computing Technology, University of the Chinese Academy of Sciences

{jie3040@aoni, wulike@fuji}.waseda.jp, zhaoliangjie55@gmail.com
shxu@toki.waseda.jp, fujimura@waseda.jp

Abstract

Self-play fine-tuning enables large language models to improve beyond supervised fine-tuning without additional human annotations by contrasting annotated responses with self-generated ones. Many existing methods rely on a fixed divergence regime. SPIN is closely related to a KL-based regime, SPACE to a Jensen-Shannon-style objective via noise contrastive estimation, and SPIF to χ^2-regularized self-play. Since these divergences exhibit different strengths depending on the distributional gap between model and target, no single choice appears to provide favorable learning dynamics across training stages. We propose IRIS (Interpolative Rényi Iterative Self-play), a Rényi-based self-play fine-tuning framework with a continuously adjustable objective. IRIS decomposes into two independent tilted risk terms over annotated and synthetic data, with exponential importance weights controlled by the order parameter α. We show that several self-play objectives can be interpreted as limiting or representative regimes at particular values of α, providing a unified theoretical perspective on these methods. An adaptive order schedule further adjusts α to the distributional gap, shifting from sharper importance weighting early in training to smoother refinement near convergence. Theoretically, we establish the fixed-point property of IRIS and analyze how α controls gradient concentration. Experiments on Zephyr-7B and Qwen2.5-3B across ten benchmarks show that IRIS improves upon baselines, reaching 44.57% average score with gains across iterations. In our setting, IRIS with only $26k$ annotated samples surpasses standard supervised fine-tuning trained on the full $200k$ dataset.

1 Introduction

Large language models (LLMs) have achieved remarkable capabilities through post-training alignment with human preferences[47, 68, 13, 61, 52, 51, 4, 28]. The prevailing approaches, including reinforcement learning from human feedback (RLHF)[12, 5, 82, 6] and direct preference optimization (DPO)[56, 69, 80], rely on curated preference datasets that are expensive to acquire and difficult to scale[20, 44, 39, 87]. This practical constraint motivates a natural question: can an LLM continue to improve beyond the ceiling of supervised fine-tuning (SFT)[52, 71] without any additional human-annotated data? Self-play fine-tuning (SPIN)[11] provides an affirmative answer. Drawing inspiration from the self-play mechanism in AlphaGo Zero[63], SPIN frames fine-tuning as a two-player game in which the current model learns to distinguish human-written responses from those generated by its previous iteration. Empirically, SPIN enables a weak SFT model to match models trained with DPO without acquiring new human annotations, and has been applied to various LLM fields[86, 80, 24].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

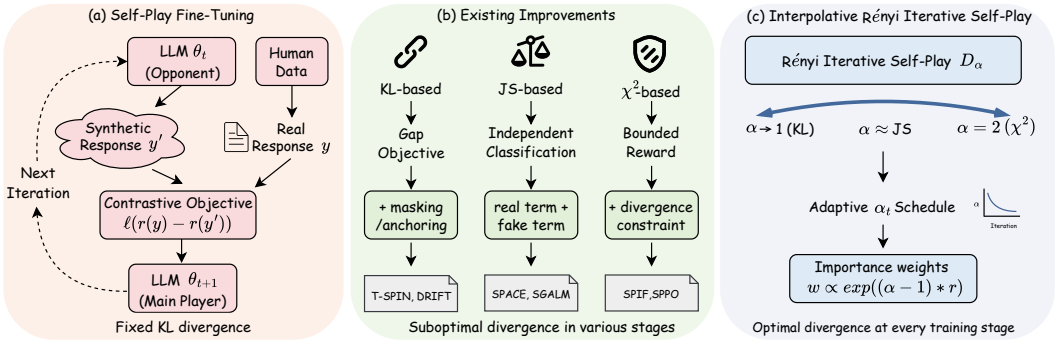


Figure 1: Overview of self-play fine-tuning paradigms: (a) the standard framework trains $\pi_{\theta_{t+1}}$ to distinguish human from synthetic responses via a contrastive objective under a fixed KL divergence; (b) existing improvements adopt KL-based, JS-based, or χ^2-based objectives, each suboptimal outside its preferred training regime; (c) IRIS places these methods within a Rényi-based perspective, with an adaptive schedule α_t that interpolates among corresponding divergence regimes during training.

Despite this conceptual elegance, SPIN exhibits several practical limitations that have motivated a growing body of follow-up work[43, 75, 76, 79, 40, 85, 38, 18, 84]. The gap-based objective, which contrasts real and synthetic responses through their reward difference, degenerates to a trivial constant as the two distributions converge during training. This gradient vanishing phenomenon causes instability and performance degradation in later iterations. SPACE[75] addresses this problem by reformulating the objective via noise contrastive estimation (NCE)[30, 31], replacing the gap-based loss with independent classification losses for real and synthetic samples. From a complementary perspective, SGALM[79] constructs a GAN-style adversarial game[27] within a single LLM, arriving at a structurally similar binary classification framework. Meanwhile, T-SPIN[76] tackles the vanishing reward advantage across iterations by introducing triplets that incorporate historical model outputs alongside current ones, maintaining a persistent learning signal through the entire training process. DRIFT[43] observes that SPIN applies uniform training pressure across all tokens, including those the model already predicts correctly, and proposes difference-aware masking to concentrate the gradient on error regions. Most recently, SPIF[40] reveals that SPIN's implicit reward function can grow unboundedly, and introduces χ^2 divergence[54, 48] regularization to constrain reward magnitudes.

Each of these methods remedies a specific failure mode, but we observe that they share a common root cause. Many recent self-play fine-tuning methods[11, 75, 76, 43, 79, 40, 1, 80, 67, 87] can be interpreted through the lens of f-divergence optimization, although they differ in data construction, optimization parameterization, and auxiliary design. The existing strategies fall into three categories accordingly. The first retains the gap-based contrastive structure under the KL divergence, strengthening it via historical anchoring, token-level masking, or opponent regularization[11, 76, 43, 1]. The second replaces the gap with independent binary classification of real and synthetic samples under the Jensen-Shannon divergence, encompassing noise contrastive estimation, adversarial, and self-rewarding approaches[75, 79, 87]. The third enforces explicit divergence constraints or game-theoretic equilibrium formulations grounded in χ^2 regularization or Nash convergence[40, 80, 67]. This taxonomy suggests a practical limitation of existing methods: a single fixed divergence may not provide equally favorable learning dynamics across all stages of training. The KL divergence provides smooth gradients when the model is far from the target but loses sensitivity as the distributions converge. The χ^2 divergence offers bounded rewards and fine-grained discrimination near convergence, yet over-penalizes outliers early on. The JS divergence ensures stability through independent optimization but is less sensitive to differences in later stages. This explains why each category excels in certain regimes while poorly in others. The comparisons are shown in Figure 1.

Building on this insight, we propose **IRIS** (**I**nterpolative **R**ényi **I**terative **S**elf-play), a unified framework that resolves the divergence selection problem through the Rényi divergence[60]. The Rényi divergence $D_\alpha(p\|q)$ is a one-parameter generalization that continuously interpolates among the KL divergence ($\alpha \to 1$), the χ^2 divergence ($\alpha = 2$), and the Hellinger distance ($\alpha = 1/2$) [72, 14], covering several divergence regimes closely related to those underlying existing self-play objectives. Exploiting the variational representation of this family [42, 9], we derive the IRIS objective as a loss

that decomposes into two independent terms over real and synthetic data, inheriting the stability of independent classification. The gradient features exponential importance weights $\tilde{w} \propto \exp((\alpha-1)\cdot r)$ that continuously concentrate learning on informative samples through a principled, tunable mechanism. We further introduce an adaptive order schedule α_t that adjusts the divergence geometry to the distributional gap: larger α encourages mode-covering exploration early in training, while smaller α enables mode-seeking refinement near convergence, selecting the optimal divergence at every stage.

Theoretically, we establish the fixed-point property of IRIS, analyze its gradient structure, and show that several self-play objectives can be interpreted as representative or limiting regimes under particular choices of α. We further introduce an adaptive schedule to adjust the divergence geometry during training. Empirically, experiments on Zephyr-7B[71] and Qwen2.5-3B-Instruct[83] across ten tasks from the Open LLM Leaderboard[7] show that IRIS reaches 44.57% average score at Iter 4, with up to 10.75-point gains over SFT on tasks such as IFEval[90]. IRIS maintains stable improvements across all four iterations, whereas several strong baselines plateau or show less stable behavior later.

Our contributions. (i) We present a Rényi-based perspective that connects representative self-play objectives through distinct divergence regimes; (ii) We propose IRIS, a self-play fine-tuning framework with a continuously adjustable objective and an adaptive order schedule; (iii) We analyze the fixed-point and gradient properties of IRIS; (iv) Experiments across ten benchmarks show that IRIS outperforms other self-play baselines, while maintaining stable gains across iterations.

2 Preliminaries

In this section, we introduce the necessary background on supervised fine-tuning and self-play fine-tuning, and examine existing improvements through the unifying lens of f-divergence theory.

Supervised Fine-Tuning. Supervised fine-tuning (SFT) is a straightforward yet powerful approach for large language models to achieve effective task adaptation using labeled data[20, 77, 66]. Given a prompt $\mathbf{x}$ sampled from a task-specific distribution $q(\cdot)$ and a corresponding high-quality response $\mathbf{y}$ from the target distribution $p_{\text{data}}(\cdot|\mathbf{x})$, supervised fine-tuning (SFT) learns a policy p_θ by maximizing the log-likelihood of annotated data:

$$\mathcal{L}_{\text{SFT}}(\theta) = \mathbb{E}_{\mathbf{x}\sim q(\cdot),\, \mathbf{y}\sim p_{\text{data}}(\cdot|\mathbf{x})}\big[\log p_\theta(\mathbf{y}|\mathbf{x})\big]. \tag{1}$$

Maximizing (1) is equivalent to minimizing the KL divergence $\mathrm{KL}(p_{\text{data}}(\cdot|\mathbf{x}) \,\|\, p_\theta(\cdot|\mathbf{x}))$. By the non-negativity of the KL divergence, the optimal solution satisfies $p_{\theta^*}(\cdot|\mathbf{x}) = p_{\text{data}}(\cdot|\mathbf{x})$. However, approaching this optimum requires extensive annotated data, limiting practical applicability of SFT.

Self-Play Fine-Tuning. Self-play fine-tuning[11, 24, 59] addresses the data scarcity limitation of SFT by augmenting training with model-generated responses. The framework consists of a main player and an opponent player, both instantiated from the same LLM with different parameters. At each iteration $t+1$, the opponent player p_{θ_t} generates a synthetic response $\mathbf{y}' \sim p_{\theta_t}(\cdot|\mathbf{x})$ for each prompt $\mathbf{x}$. The main player then updates its parameters to distinguish the annotated response $\mathbf{y}$ from the synthetic one $\mathbf{y}'$. Defining the log-ratio reward and objectives as

$$r(\mathbf{x}, \hat{\mathbf{y}}) = \log p_\theta(\hat{\mathbf{y}}|\mathbf{x}) - \log p_{\theta_t}(\hat{\mathbf{y}}|\mathbf{x}), \tag{2}$$

$$\mathcal{L}_{\text{SPIN}}(\theta) = \mathbb{E}[\ell(r(\mathbf{x}, \mathbf{y}) - r(\mathbf{x}, \mathbf{y}'))]\,. \tag{3}$$

where $\ell(\cdot)$ is a monotonically decreasing convex function, typically chosen as the logistic loss $\ell(t) = \log(1 + e^{-t})$, and the expectation is taken over $\mathbf{x} \sim q(\cdot)$, $\mathbf{y} \sim p_{\text{data}}(\cdot|\mathbf{x})$, and $\mathbf{y}' \sim p_{\theta_t}(\cdot|\mathbf{x})$. The global optimum of (3) is attained when $p_\theta = p_{\text{data}}$, at which point the reward gap vanishes for all prompt-response pairs and the objective degenerates into a constant independent of θ. This fixed-point property guarantees convergence in principle, but it also implies that the training signal weakens as p_{θ_t} approaches p_{data}, causing optimization instability in later iterations.

Self-Play Improvements as f-Divergence Minimization. The instability described above has motivated a growing body of improvements to the self-play framework [75, 76, 43, 79, 40, 1, 80, 67].A closer examination reveals that these methods, despite their diverse motivations, share a common mathematical structure rooted in f-divergence theory [2, 16]. Recall that an f-divergence $D_f(p\|q) = \mathbb{E}_q[f(p/q)]$ is uniquely determined by its convex generator f, and admits a variational representation via the Fenchel conjugate [49] that enables tractable optimization through neural network parameterization [50]. Through this lens, existing self-play methods cluster into three

families according to the divergence they implicitly minimize. The first family operates under the KL divergence, preserving the gap-based contrastive structure of SPIN while strengthening it through historical anchoring [76], token-level masking [43], or opponent regularization [1]. The second family corresponds to the Jensen-Shannon divergence [27], replacing the reward gap with independent binary classification of real and synthetic samples via noise contrastive estimation [75] or adversarial training [79]. The third family employs the χ^2 divergence, enforcing explicit divergence constraints [40] or game-theoretic equilibrium formulations [80, 67] that bound the reward magnitude.

This classification exposes a fundamental limitation shared by all existing approaches: each operates under a single fixed divergence throughout training. As discussed in Section 1, no single f-divergence provides uniformly optimal learning dynamics across all stages, since different divergences exhibit complementary strengths depending on the distributional gap between the model and the target. This observation motivates a framework capable of adapting its divergence behavior during training, which we develop in the next section using the Rényi divergence family[60, 72].

3 Method

In this section, we present IRIS, a unified self-play fine-tuning framework built on the Rényi divergence. We construct the IRIS objective, analyze its gradient structure, and introduce an adaptive order schedule throughout training. Additional proofs and derivatives are provided in Appendix B.

3.1 The IRIS Objective

The analysis in Section II reveals that each existing self-play method commits to a single f-divergence throughout training, and no fixed divergence provides uniformly optimal learning dynamics across all stages. To address this limitation, we seek a parameterized objective that continuously interpolates among different divergence behaviors while preserving the desirable properties of self-play fine-tuning: a fixed point at $p_\theta = p_{\text{data}}$, independent optimization over real and synthetic data, and tractable gradient computation. Our method builds on the *tilted risk* principle [41]. For a random variable Z under distribution p, the tilted risk of order $s \neq 0$ is defined as

$$\mathcal{R}_s(Z;\, p) = \frac{1}{s} \log \mathbb{E}_p\big[e^{sZ}\big]\,. \tag{4}$$

This quantity is the scaled cumulant generating function of Z, interpolating between the expectation ($s \to 0$) and the essential supremum ($s \to +\infty$). The tilted risk is connected to the Rényi divergence: for any distributions p and q, the Rényi divergence of order α satisfies $D_\alpha(p\|q) = \mathcal{R}_\alpha\Big(\log \frac{p}{q};\, q\Big)$ [72], establishing the tilted risk as a building block for Rényi-based objectives. Using the log-ratio reward $r(\mathbf{x}, \hat{\mathbf{y}}) = \log p_\theta(\hat{\mathbf{y}}|\mathbf{x}) - \log p_{\theta_t}(\hat{\mathbf{y}}|\mathbf{x})$ defined in (2), we construct the IRIS objective for the main player by applying tilted risks of complementary orders to the real and synthetic data:

$$\mathcal{L}_{\text{IRIS}}(\theta;\, \alpha) = \underbrace{-\frac{1}{\alpha-1}\, \log \mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}\Big[e^{(\alpha-1)\, r(\mathbf{x},\mathbf{y})}\Big]}_{\text{tilted reward on real data}} + \underbrace{\frac{1}{\alpha}\, \log \mathbb{E}_{\mathbf{y}'\sim p_{\theta_t}}\Big[e^{\alpha\, r(\mathbf{x},\mathbf{y}')}\Big]}_{\text{tilted penalty on synthetic data}}, \tag{5}$$

where the outer expectation over prompts $\mathbf{x} \sim q(\cdot)$ is omitted for notational clarity, and $\alpha > 0$ with $\alpha \neq 1$ is the Rényi order parameter. The restriction at $\alpha = 1$ applies to this closed form, whose continuous KL limit is given in (24). The first term is the negated tilted risk of order $(\alpha-1)$ evaluated on real data, which encourages the model to assign high reward to human-annotated responses with sensitivity controlled by α. The second term is the tilted risk of order α on synthetic data, which penalizes high reward assigned to model-generated responses. The two terms operate independently on real and synthetic samples, inheriting the structural stability of independent classification [75, 79]. The asymmetry in orders $(\alpha-1)$ and α arises naturally from the Rényi variational principle [9] and ensures that the two terms remain balanced across the full range of α. A requirement for any self-play objective is that the target distribution p_{data} constitutes a fixed point. We establish this for IRIS.

Proposition 1 (Fixed Point)**.** *For any $\alpha > 0$ with $\alpha \neq 1$, the global minimum of $\mathcal{L}_{\text{IRIS}}(\theta;\, \alpha)$ over θ is attained at $p_\theta = p_{\text{data}}$. At this optimum,*

$$\mathcal{L}_{\text{IRIS}}(\theta^*;\, \alpha) = -\frac{1}{\alpha}\, D_\alpha(p_{\text{data}}\|p_{\theta_t}), \tag{6}$$

Table 1: Unification of self-play fine-tuning methods through the IRIS framework. Representative self-play objectives can be related to specific regimes within the Rényi divergence family.

Order α	Divergence	Weight profile	Corresponding methods
$\alpha \to 1$	KL	Uniform	SPIN[11], T-SPIN[76], DRIFT[43]
$\alpha \approx 1$	Approx. JS	Sigmoid	SPACE[75], SGALM[79]
$\alpha = 2$	χ^2	Quadratic	SPIF[40], SPPO[80]
$\alpha = 1/2$	Hellinger	Inverse-exponential	(new regime via IRIS)

where D_α denotes the Rényi divergence of order α. Furthermore, if $p_{\theta_t} = p_{\text{data}}$, then $\mathcal{L}_{\text{IRIS}}(\theta^*;\, \alpha) = 0$, confirming that the equilibrium $p_\theta = p_{\theta_t} = p_{\text{data}}$ is a fixed point of the iterative self-play process.

Gradient Analysis and Importance Weighting. To understand the optimization dynamics of IRIS, we derive the gradient of (5) with respect to θ and analyze how order α controls the model training.

Proposition 2 (Gradient Structure)**.** *For fixed α, the gradient of $\mathcal{L}_{\text{IRIS}}(\theta;\, \alpha)$ with respect to θ takes the form*

$$\nabla_\theta \mathcal{L}_{\text{IRIS}} = -\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}\left[\tilde{w}^+(\mathbf{y})\, \nabla_\theta \log p_\theta(\mathbf{y}|\mathbf{x})\right] + \mathbb{E}_{\mathbf{y}'\sim p_{\theta_t}}\left[\tilde{w}^-(\mathbf{y}')\, \nabla_\theta \log p_\theta(\mathbf{y}'|\mathbf{x})\right], \tag{7}$$

where the normalized importance weights are

$$\tilde{w}^+(\mathbf{y}) = \frac{e^{(\alpha-1)\, r(\mathbf{x},\mathbf{y})}}{\mathbb{E}_{p_{\text{data}}}\left[e^{(\alpha-1)\, r}\right]}, \qquad \tilde{w}^-(\mathbf{y}') = \frac{e^{\alpha\, r(\mathbf{x},\mathbf{y}')}}{\mathbb{E}_{p_{\theta_t}}[e^{\alpha\, r}]}. \tag{8}$$

The gradient in (7) admits a natural interpretation as importance-weighted likelihood updates: real responses are reinforced while synthetic ones are suppressed, with exponential weights acting as a differentiable attention mechanism over samples. For fixed rewards and $\alpha > 1$, increasing α makes the weights in both terms more concentrated on responses with high reward r. As α approaches 1 from above, the weights on annotated responses approach uniform weighting. The weights on synthetic responses instead approach values proportional to e^r. For $0 < \alpha < 1$, the weights on annotated responses favor responses with low reward, while those on synthetic responses still favor responses with high reward. This exponential tilting further connects IRIS to tilted empirical risk minimization [41], where α balances average-case and worst-case performance.

Unification of Existing Methods. A notable consequence of the tilted risk construction is that the IRIS objective recovers the divergence families underlying existing self-play methods at specific values of α, confirming that these methods were implicitly operating within the Rényi family. The independent treatment of real and synthetic terms in (5) further parallels the decoupled structure of SPACE [75] and SGALM [79] under the Jensen-Shannon divergence. The correspondences of various regimes are summarized in Table 1 and three informative regimes are listed below:

- **KL regime** ($\alpha \to 1$). Using the limit $\frac{1}{s}\log \mathbb{E}[e^{sZ}] \to \mathbb{E}[Z]$ as $s \to 0$, $\mathcal{L}_{\text{IRIS}}$ converges to $-\mathbb{E}_{p_{\text{data}}}[r] + \log \mathbb{E}_{p_{\theta_t}}[e^r]$, the Donsker-Varadhan form of the KL divergence [21], and the annotated-response weights become uniform. This recovers the divergence regime of SPIN [11] together with its gap-based variants [76, 43, 1], which approximate this KL objective via logistic losses.
- **χ^2 regime** ($\alpha = 2$). The identity $\mathbb{E}_{p_{\theta_t}}[e^{2r}] = 1 + \chi^2(p_\theta \| p_{\theta_t})$ implies that the second term directly penalizes the χ^2 divergence between the model and the opponent, which aligns with the bounded-reward formulations of SPIF [40], SPPO [80], and RSPO [67].
- **Hellinger regime** ($\alpha = 1/2$). Under the squared Hellinger distance, $\tilde{w}^+ \propto e^{-r/2}$ emphasizes underperforming real responses, producing mode-covering behavior absent from prior methods.

3.2 Adaptive Order Schedule

The unification established above motivates a natural question: rather than committing to a single α throughout training, can we adapt the order during training? This question is central to IRIS, because the appropriate divergence geometry depends on the current distributional gap between the policy and the target. We consider two complementary strategies for choosing the order during training.

Algorithm 1 IRIS: Interpolative Rényi Iterative Self-Play

Input: SFT model p_{θ_0}, dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, iterations T, constant c, clipping bounds $\alpha_{\min}$ and $\alpha_{\max}$
for $t = 0, 1, \ldots, T-1$ **do**
  Keep the opponent p_{θ_t} fixed
  Generate synthetic responses $\mathbf{y}'_i \sim p_{\theta_t}(\cdot|\mathbf{x}_i)$
  for each training mini-batch k in iteration t **do**
    Compute current log-ratio rewards via (2)
    Set $\alpha_t^{(k)}$ via (9)
    Update θ using (5) with $\alpha_t^{(k)}$ on the current mini-batch
  end for
  Set $\theta_{t+1} \leftarrow \theta$
end for
Output: Fine-tuned model p_{θ_T}

Distributional Gap Feedback. We update $\alpha_t^{(k)}$ at every mini-batch using the current reward gap.

$$\alpha_t^{(k)} = 1 + c \cdot \hat{D}_t^{(k)}, \qquad \hat{D}_t^{(k)} = \widehat{\mathbb{E}}_{\mathbf{y} \sim p_{\text{data}}}\left[r_t^{(k)}(\mathbf{x}, \mathbf{y})\right] - \widehat{\mathbb{E}}_{\mathbf{y}' \sim p_{\theta_t}}\left[r_t^{(k)}(\mathbf{x}, \mathbf{y}')\right], \tag{9}$$

where $c > 0$ is a scaling constant, and $\alpha_{\min}$ and $\alpha_{\max}$ are the clipping bounds. Here t indexes self-play iterations and k indexes mini-batches within iteration t. The empirical expectations are computed over the current mini-batch, giving the reward gap $\hat{D}_t^{(k)}$ as a proxy for the distributional gap [11]. The reward $r_t^{(k)}$ is evaluated as in (2) using the current policy p_θ. The opponent p_{θ_t} remains fixed throughout iteration t. This choice ties the schedule directly to the training signal already used by self-play, and therefore avoids introducing an additional external criterion for tuning $\alpha_t^{(k)}$. When the model is far from the target ($\hat{D}_t^{(k)}$ large), $\alpha_t^{(k)}$ is large, promoting mode-covering exploration through the χ^2-like behavior and concentrated importance weights. This places more emphasis on informative high-gap samples, which is beneficial when the model and data distributions remain substantially separated. As the model converges ($\hat{D}_t^{(k)} \to 0$), $\alpha_t^{(k)}$ decreases toward 1, transitioning to the KL regime that provides smoother, more uniform gradients near equilibrium.

Geometric Annealing. For simplicity and reproducibility, we also consider a predetermined schedule:

$$\alpha_t = \alpha_{\max} \cdot \left(\frac{\alpha_{\min}}{\alpha_{\max}}\right)^{t/T}, \tag{10}$$

which interpolates from $\alpha_{\max}$ to $\alpha_{\min}$ over T iterations. It remains fixed within each iteration. This schedule requires no statistics and is straightforward to implement. Both strategies use orders above 1 to emphasize high-reward responses. Near 1, annotated-response weights become more uniform, while orders below 1 favor low-reward annotated responses. In Section IV, we compare both strategies and find that distributional gap feedback outperforms geometric annealing, as it responds to dynamics rather than following a predetermined trajectory.

3.3 Algorithm Summary

We summarize IRIS in Algorithm 1. The procedure follows the standard self-play pipeline [11], with two modifications: (i) the loss function in (5) replaces the gap-based or fixed-divergence objectives of prior methods, and (ii) the order $\alpha_t^{(k)}$ is updated at every mini-batch based on the current reward statistics. In practice, the log-sum-exp operations in (5) are stabilized by subtracting the batch maximum before exponentiation [10], the importance weights $\tilde{w}^{\pm}$ in (8) are normalized within each mini-batch to prevent any single sample from dominating the gradient, and the reward gap $\hat{D}_t^{(k)}$ in (9) is recomputed for each mini-batch before setting $\alpha_t^{(k)}$. IRIS is fully compatible with existing self-play codebases [11, 75], requiring only a substitution of the loss computation module while leaving the generation, data processing, and distributed training components unchanged.

Table 2: Performance (%) comparisons on Zephyr-7B, where we report improvements (green) or degradations (red) over the previous iteration. **Bold**: best result across all iterations for each method.

Model		Math & Logic			Multi-Domain Knowledge			Commonsense Reasoning			IF	Avg
		GSM8K	MATH	MUSR	MMLU	MMLUP	GPQA	HellaSwag	WG	BBH	IFEval	
Zephyr-7B		25.73	1.71	38.07	56.44	29.19	28.80	81.89	74.20	44.54	2.97	38.35
SFT		41.06	3.15	39.79	57.22	29.23	28.27	83.40	73.08	44.61	20.07	41.99
SPIN	Iter0	30.25	4.30	41.42	56.43	28.24	29.31	84.17	73.63	44.53	8.47	40.08
	Iter1	32.68	2.87	38.79	56.51	27.72	29.35	83.63	73.84	44.72	7.63	39.77 (−0.31)
	Iter2	36.85	2.72	41.53	57.46	28.53	28.61	83.58	73.90	43.78	14.72	**41.17** (+1.40)
	Iter3	33.54	2.58	39.13	55.71	25.83	27.54	83.41	74.35	43.22	22.34	40.77 (−0.40)
	Iter4	31.63	2.43	38.58	54.82	25.17	26.93	83.20	74.21	42.96	21.87	40.19 (−0.58)
SPACE	Iter0	40.72	3.67	40.15	57.60	28.94	28.83	83.38	73.71	43.14	10.82	41.10
	Iter1	40.83	4.20	40.67	58.17	29.08	29.15	83.52	73.56	43.31	12.29	41.48 (+0.38)
	Iter2	41.15	4.48	41.23	58.24	29.15	29.27	83.47	73.82	43.62	14.36	41.88 (+0.40)
	Iter3	41.37	4.64	41.59	58.12	29.23	29.41	83.55	73.93	43.87	16.43	42.21 (+0.33)
	Iter4	41.82	4.73	41.86	58.07	29.30	29.58	83.61	73.67	44.18	17.12	**42.40** (+0.19)
T-SPIN	Iter0	36.47	3.40	37.28	56.53	29.07	28.23	82.82	73.61	44.28	7.31	39.90
	Iter1	39.82	3.53	38.01	57.02	29.28	29.14	83.08	73.85	44.63	26.73	42.51 (+2.61)
	Iter2	40.93	3.77	38.68	57.12	29.40	29.59	83.24	73.78	44.62	27.83	42.90 (+0.39)
	Iter3	40.85	3.96	39.32	57.83	29.52	30.27	83.42	73.86	45.14	29.39	43.36 (+0.46)
	Iter4	40.40	3.98	39.84	57.76	29.47	30.92	83.28	73.93	45.08	30.84	**43.56** (+0.20)
SPIF	Iter0	33.81	4.14	41.73	56.82	28.63	29.45	84.25	73.47	44.68	9.43	40.64
	Iter1	37.60	4.38	41.83	57.35	29.06	29.57	83.86	73.58	44.73	14.92	41.69 (+1.05)
	Iter2	39.42	4.53	42.19	57.68	29.36	29.84	83.67	73.74	44.76	18.83	**42.40** (+0.71)
	Iter3	38.85	4.47	41.58	57.26	29.10	29.53	83.54	73.62	44.38	19.42	42.18 (−0.22)
	Iter4	38.27	4.30	41.14	56.93	28.87	29.29	83.42	73.53	44.15	19.17	41.91 (−0.27)
IRIS	Iter0	35.18	3.63	41.87	56.78	29.12	29.52	84.13	73.68	44.81	10.29	40.90
	Iter1	41.53	4.17	42.08	57.86	29.67	30.14	83.95	73.92	45.23	22.61	43.12 (+2.22)
	Iter2	41.74	4.50	42.21	58.34	30.12	30.58	83.87	74.16	45.67	26.49	43.77 (+0.65)
	Iter3	41.90	4.83	42.78	58.72	30.43	30.70	83.93	74.28	45.82	29.36	44.28 (+0.51)
	Iter4	42.21	4.97	42.89	58.93	30.57	30.96	84.05	74.43	45.82	30.82	**44.57** (+0.29)

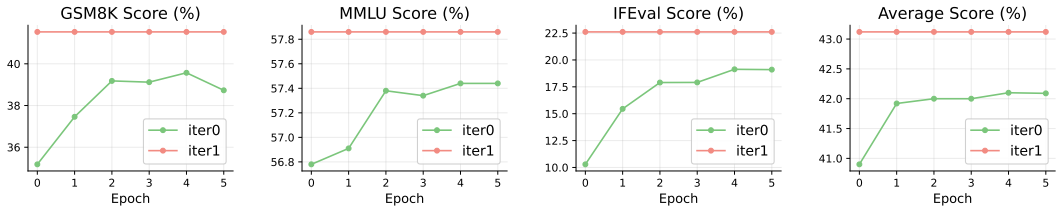


Figure 2: Performance comparison between training at iteration 0 for multiple epochs on fixed synthetic responses (green) and training at iteration 1 for two epochs on regenerated responses (red).

4 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of IRIS. We first describe the experimental settings, and then report the main results with analyses. Finally, we present ablation studies on the adaptive order schedule. More details are provided in Appendix C.

4.1 Experimental Setup

Models and training. Following SPIN [11] and subsequent works [75, 76, 40], we randomly sample $50k$ prompt-response pairs from the first round of interactions in Ultrachat200k [19] and evaluate on Zephyr-7B-SFT-Full [71] and Qwen2.5-3B-Instruct [83]. At each of five self-play iterations (Iter 0 through Iter 4), we generate one synthetic response per prompt and combine it with the annotated response for training. We use RMSProp [35] as optimizer with a global batch size of 64 and 2 epochs per iteration. For the adaptive schedule in (9), we set $c = 0.5$ with $\alpha_t^{(k)} \in [0.5, 3.0]$. All the experiments run on 8 NVIDIA H100 GPUs using the Alignment Handbook [70] and Accelerate [29].

Evaluation and baselines. We evaluate on ten diverse tasks from the HuggingFace Open LLM Leaderboard [7], grouped into mathematical and logical reasoning (GSM8K [15], MATH [33], MUSR [64]), multi-domain knowledge (MMLU [32], MMLU-Pro [78], GPQA [58]), commonsense

Table 3: Ablation on order schedule evaluated on Zephyr-7B. All variants use IRIS objective in (5) and differ in how α is determined. The Distributional Gap Feedback schedule achieves best performance.

Schedule strategy	α range	Iter0	Iter1	Iter2	Iter3	Iter4	Best Avg
Fixed $\alpha = 0.5$	$\{0.5\}$	40.21	41.08	41.39	41.53	41.67	41.67
Fixed $\alpha = 1.5$	$\{1.5\}$	41.17	42.46	43.03	43.14	42.91	43.14
Fixed $\alpha = 2.0$	$\{2.0\}$	41.58	42.73	42.86	42.51	42.18	42.86
Geometric Annealing	$[0.8, 2.5]$	41.36	42.89	43.47	43.78	43.92	43.92
Gap Feedback (Ours)	$[0.5, 3.0]$	40.90	43.12	43.77	44.28	44.57	**44.57**

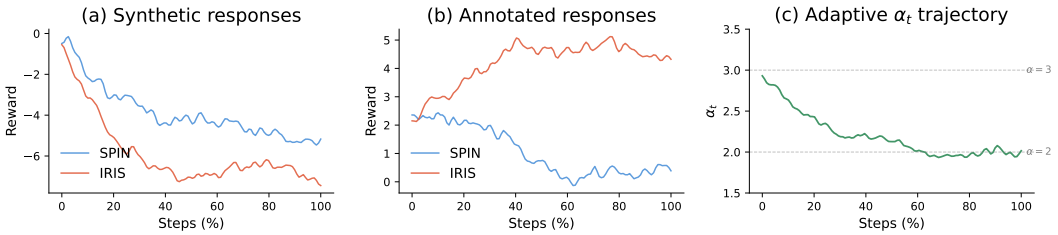


Figure 3: Training dynamics at Iter 0 on Zephyr-7B: (a) reward of synthetic responses; (b) reward of annotated responses; (c)trajectory of the adaptive order α_t under distributional gap feedback schedule.

reasoning (HellaSwag [88], WinoGrande [62], BBH [65]), and instruction following (IFEval [90]), using the Language Model Evaluation Harness [25]. We compare IRIS against SFT trained on the full $200k$ dataset, SPIN [11], SPACE [75], T-SPIN [76], and SPIF [40]. All self-play methods are trained on the same $50k$ subset with identical generation and training configurations for fair comparison.

4.2 Main Results

Comparison across methods. Table 2 presents the main results on Zephyr-7B across all ten benchmarks. Several observations emerge from these results. First, all self-play methods improve over the pretrained model. SPACE, T-SPIN, SPIF, and IRIS also surpass SFT at their best iterations. Second, IRIS consistently achieves the highest average score across all iterations, reaching 44.57% at Iter 4, compared to 43.56% for T-SPIN, 42.40% for SPACE and SPIF, and 41.17% for SPIN. Third, the improvements from IRIS are particularly pronounced on MATH, MUSR and IFEval tasks with 1.82-point, 3.10-point and 10.75-point increases respectively compared with SFT base, where the adaptive divergence schedule provides the greatest benefit. Results on Qwen2.5-3B-Instruct, which also exhibit similar overall trends, are provided in Appendix C.3.

Comparison to training with more epochs. We examine whether the iterative gains of self-play can be matched by simply training longer on fixed synthetic responses, extending iteration 0 to five epochs and comparing against iteration 1 with two epochs on regenerated responses (Figure 2). The average score plateaus near 42.1 after five epochs, remaining well below the 43.1 that iteration 1 achieves with two epochs alone, with especially large gaps on GSM8K and IFEval. The bottleneck lies in the stagnation of negative samples: as the policy improves, the fixed synthetic responses grow increasingly uninformative. Regenerating responses at iteration 1 supplies negatives that reflect the model's current weaknesses, and in IRIS additionally triggers a recalibration of $\alpha_t^{(k)}$ via the adaptive schedule (9), keeping the objective aligned with the evolving distributional gap. These results confirm that data regeneration, not prolonged training on static data, drives cross-iteration improvement.

Training dynamics. Reward decline on high-quality responses is a known issue in preference alignment [23, 53, 55, 81] and was first studied in self-play fine-tuning by SPACE [75]. Figure 3 compares IRIS and SPIN at Iter 0. Panels (a) and (b) show that SPIN enlarges the reward margin, but the reward on annotated responses decreases during training, while the reward on synthetic responses decreases faster. This is a limitation of gap-based optimization: SPIN improves the objective through the relative difference $r(\mathbf{x}, \mathbf{y}) - r(\mathbf{x}, \mathbf{y}')$, without directly increasing the absolute reward on real data. As a result, a larger margin does not necessarily mean better modeling of high-quality responses. In contrast, IRIS increases the reward on real data and decreases it on synthetic data, since its objective

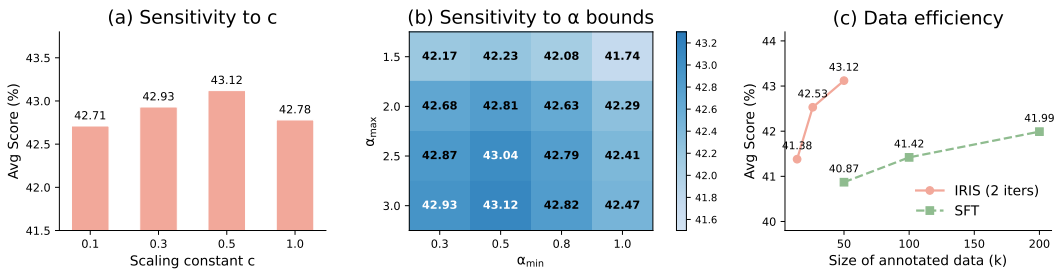


Figure 4: Ablation analysis on Zephyr-7B: (a) sensitivity to scaling constant c in the distributional gap feedback schedule, evaluated at Iter 1; (b) heatmap of average scores for different clipping bounds $\alpha_{\min}$ and $\alpha_{\max}$; (c) comparison between IRIS and SFT with varying amounts of annotated data.

optimizes both terms separately via a tilted reward on annotated data and a tilted penalty on synthetic ones. The margin in IRIS reflects progress toward the target distribution. Panel (c) shows that $\alpha_t^{(k)}$ starts near 3.0 and approaches 2.0 within Iter 0 under distributional gap feedback, indicating a shift from sharper reweighting early in training to smoother refinement.

4.3 Ablation Studies

Effect of the adaptive order schedule. Table 3 compares five strategies for setting the Rényi order α on Zephyr-7B. Fixing α at 0.5, 1.5, or 2.0 throughout training corresponds to the Hellinger, intermediate, and χ^2 regimes, respectively. The results reveal a clear tradeoff between stability and aggressiveness: $\alpha = 0.5$ produces the most stable trajectory with no degradation across iterations but converges slowly, $\alpha = 2.0$ delivers the strongest initial gains but becomes unstable after Iter 2, and $\alpha = 1.5$ balances the two yet still plateaus at a lower final performance level. Geometric annealing from $\alpha_{\max} = 2.5$ to $\alpha_{\min} = 0.8$ surpasses all fixed choices by combining early exploration with late refinement across self-play iterations. The distributional gap feedback schedule in (9) consistently achieves the highest score at every iteration beyond Iter 0, as it calibrates the order to the actual divergence between the policy and the data distribution rather than following a predetermined decay.

Sensitivity to hyperparameters. We examine the sensitivity of IRIS to the scaling constant c in (9) and the clipping bounds $[\alpha_{\min}, \alpha_{\max}]$, both evaluated at Iter 1 on Zephyr-7B. Figure 4(a) reports the average score for $c \in \{0.1, 0.3, 0.5, 1.0\}$: all four values fall between 42.71% and 43.12%, a range of less than 0.5 points, indicating that the feedback gain has minimal impact on final performance so long as it remains within a reasonable magnitude. Figure 4(b) presents a heatmap over $\alpha_{\max} \in \{1.5, 2.0, 2.5, 3.0\}$ and $\alpha_{\min} \in \{0.3, 0.5, 0.8, 1.0\}$. Performance exceeds 42.6% whenever $\alpha_{\max} \geq 2.0$ and $\alpha_{\min} \leq 0.8$, confirming that the schedule tolerates a broad operating region, while the combination $\alpha_{\max} = 3.0$, $\alpha_{\min} = 0.5$ achieves the peak of 43.12%. Based on these results, we adopt $c = 0.5$, $\alpha_{\max} = 3.0$, and $\alpha_{\min} = 0.5$ as default values throughout all other experiments.

Data efficiency. Following the protocol of SPACE [75] and T-SPIN [76], we compare IRIS with SFT under varying amounts of annotated data. We train IRIS on subsets of $14k$, $26k$, and $50k$ samples for two iterations, and SFT on $50k$, $100k$, and $200k$ samples. Figure 4(c) shows that IRIS with only $26k$ annotated responses reaches 42.53% after two iterations, surpassing SFT trained on the full $200k$ dataset (41.99%) by 0.54 points while using roughly one eighth of the annotated data. When the annotated set grows to $50k$, IRIS further improves to 43.12%, exceeding $200k$ SFT by 1.13 points. The gap widens as more annotated data becomes available, suggesting that the adaptive schedule extracts progressively richer training signal from larger annotation pools. These results confirm that iterative self-play amplifies the value of limited annotations, and that the adaptive divergence mechanism in IRIS provides complementary gains by tailoring the Rényi order to the current policy state at each iteration rather than relying on a fixed and potentially suboptimal divergence measure.

5 Conclusion and discussion

We have presented IRIS, a self-play fine-tuning framework grounded in the Rényi divergence family. The IRIS objective decomposes into two independent tilted risk terms over annotated and synthetic

data, yielding decoupled optimization and continuously adjustable importance weighting through the order parameter α. We have shown that several representative self-play objectives, including SPIN, SPACE, and SPIF, can be interpreted within the same Rényi-based perspective through particular divergence regimes. The adaptive order schedule shifts from sharper importance weighting early in training to smoother refinement near convergence, responding to the observed distributional gap rather than following a fixed trajectory. Experiments on Zephyr-7B and Qwen2.5-3B across ten benchmarks show that IRIS is competitive with and often improves upon strong baselines, while maintaining stable gains across iterations and ultimately reaching 44.57% average score at Iter 4.

Limitations and Future Work. Like all existing self-play methods, IRIS assumes a fixed target distribution throughout training. Extending the framework to non-stationary settings where the target evolves through online interaction is a natural direction, and recent advances in game-theoretic alignment via no-regret learning [89] and last-iterate convergence guarantees [74] provide promising theoretical foundations for such an extension. Our evaluation is restricted to single-turn generation on models up to 7B parameters; applying IRIS to multi-turn dialogue and scaling to larger architectures through parameter-efficient methods [37] remain unexplored. On the theoretical side, the adaptive schedule in (9) is empirically effective but lacks formal regret bounds characterizing the optimal trajectory of α_t on the Rényi manifold. Finally, combining the Rényi family with broader divergence classes [73] or extending the mechanism to token-level order adaptation may yield further gains.

Acknowledgments and Disclosures

This work received no external funding or support from third parties. The authors declare no competing interests.

References

[1] Reda Alami, Abdalgader Abubaker, Mastane Achab, Mohamed El Amine Seddik, and Salem Lahlou. Investigating regularization of self-play language models. *arXiv preprint arXiv:2404.04291*, 2024.

[2] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[3] Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, 2016.

[4] Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. Accessed: 2026-04-04.

[5] 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*, pages 4447–4455. PMLR, 2024.

[6] 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.

[7] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard (2023–2024). `https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard`, 2023. Accessed: 2026-04-04.

[8] Anil Kumar Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[9] Jeremiah Birrell, Paul Dupuis, Markos A Katsoulakis, Luc Rey-Bellet, and Jie Wang. Variational representations and neural network estimation of Rényi divergences. *SIAM Journal on Mathematics of Data Science*, 3(4):1093–1116, 2021.

[10] Pierre Blanchard, Desmond J Higham, and Nicholas J Higham. Accurately computing the log-sum-exp and softmax functions. *IMA Journal of Numerical Analysis*, 41(4):2311–2330, 2021.

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

[12] 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.

[13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[14] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

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

[16] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:299–318, 1967.

[17] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

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

[19] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[20] Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198, 2024.

[21] Monroe D Donsker and S R Srinivasa Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. IV. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

[22] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[23] Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. Towards analyzing and understanding the limitations of DPO: A theoretical perspective. *ArXiv e-prints, arXiv:2404.04626*, 2024.

[24] Chongming Gao, Ruijun Chen, Shuai Yuan, Kexin Huang, Yuanqing Yu, and Xiangnan He. SPRec: Leveraging self-play to debias preference alignment for large language model-based recommendations. *arXiv preprint arXiv:2412.09243v1*, 2024.

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

[26] Manuel Gil, Fady Alajaji, and Tamás Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

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

[28] Google. Gemini 2.5: Our most intelligent ai model. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`, 2025. Accessed: 2026-04-04.

[29] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`, 2022.

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

[31] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

[32] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *International Conference on Learning Representations*, 2021.

[33] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[34] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernández-Lobato, and Richard E. Turner. Black-box α-divergence minimization. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1511–1520, 2016.

[35] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning, lecture 6e, RMSProp: Divide the gradient by a running average of its recent magnitude. Technical report, University of Toronto, 2012.

[36] Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[37] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[38] Xiang Ji, Sanjeev Kulkarni, Mengdi Wang, and Tengyang Xie. Self-play with adversarial critic: Provable and scalable offline alignment for language models. *arXiv preprint arXiv:2406.04274*, 2024.

[39] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235. PMLR, 2024.

[40] Shangzhe Li, Xuchao Zhang, Chetan Bansal, and Weitong Zhang. Your self-play algorithm is secretly an adversarial imitator: Understanding llm self-play through the lens of imitation learning. *arXiv preprint arXiv:2602.01357*, 2026.

[41] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. In *International Conference on Learning Representations*, 2021.

[42] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[43] Wenjie Liao, Xiaohui Song, and Haonan Lu. Drift: Difference-aware reinforcement through iterative fine-tuning for language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31988–31996, 2026.

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

[45] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024.

[46] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[47] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, 2022.

[48] Jerzy Neyman. Contribution to the theory of the chi-square test. In *Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability*, pages 239–273. University of California Press, 1949.

[49] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[50] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[51] OpenAI. Gpt-4. `https://openai.com/index/gpt-4-research/`, 2023. Accessed: 2026-04-04.

[52] 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.

[53] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *ArXiv e-prints, arXiv:2402.13228*, 2024.

[54] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[55] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. In *First Conference on Language Modeling*, 2024.

[56] 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*, 36:53728–53741, 2023.

[57] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.

[58] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[59] Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. Learning or self-aligning? rethinking instruction fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6090–6105, 2024.

[60] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561. University of California Press, 1961.

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

[62] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740, 2020.

[63] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[64] Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, 2024.

[65] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, 2023.

[66] Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, 2024.

[67] Xiaohang Tang, Sangwoong Yoon, Seongho Son, Huizhuo Yuan, Quanquan Gu, and Ilija Bogunovic. RSPO: Regularized self-play alignment of large language models. *arXiv preprint arXiv:2503.00030*, 2025.

[68] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[69] Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, et al. Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation. *arXiv preprint arXiv:2503.12854*, 2025.

[70] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alvaro Bartolome, Alexander M. Rush, and Thomas Wolf. The alignment handbook. *GitHub repository*, 2024.

[71] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[72] Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[73] Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *The Twelfth International Conference on Learning Representations*, 2024.

[74] 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 model alignment. In *International Conference on Learning Representations*, 2025.

[75] Yibo Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Space: Noise contrastive estimation stabilizes self-play fine-tuning for large language models. *arXiv preprint arXiv:2512.07175*, 2025.

[76] Yibo Wang, Hai-Long Sun, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Triplets better than pairs: Towards stable and effective self-play fine-tuning for llms. *arXiv preprint arXiv:2601.08198*, 2026.

[77] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786, 2023.

[78] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024.

[79] Shiguang Wu, Yaqing Wang, and Quanming Yao. Self-generative adversarial fine-tuning for large language models. *arXiv preprint arXiv:2602.01137*, 2026.

[80] 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.

[81] Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. Cal-DPO: Calibrated direct preference optimization for language model alignment. In *Advances in Neural Information Processing Systems 37*, 2024.

[82] 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. *arXiv preprint arXiv:2312.11456*, 2023.

[83] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[84] Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. Online iterative reinforcement learning from human feedback with general preference model. In *Advances in Neural Information Processing Systems*, 2024.

[85] Yueqin Yin, Zhendong Wang, Yujia Xie, Weizhu Chen, and Mingyuan Zhou. Self-augmented preference optimization: Off-policy paradigms for language model alignment. *arXiv preprint arXiv:2405.20830*, 2024.

[86] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *Advances in Neural Information Processing Systems*, 37:73366–73398, 2024.

[87] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[88] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[89] 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. In *International Conference on Learning Representations*, 2025.

[90] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Additional Related Work

A.1 Rényi Divergence and Information Geometry

The Rényi divergence [60] belongs to the broader family of α-divergences studied in information geometry [3, 14], with van Erven and Harremoës [72] establishing key properties including monotonicity in the order α and continuity at the KL limit. Its variational representation [9, 42] enables tractable optimization in high-dimensional settings and has been applied to variational inference through black-box α-divergence minimization [34], where the order parameter provides a smooth interpolation between mass-covering and zero-forcing inference. Beyond generative modeling, Rényi divergence underpins Rényi differential privacy [46], which yields tighter composition bounds than standard (ε, δ)-DP through moment-based accounting, and connects to distributionally robust optimization via tilted empirical risk minimization [41], where the tilting parameter controls the tradeoff between average-case and worst-case objectives. Gil [26] and Wang et al. [73] offer comprehensive treatments of Rényi-based estimation and general divergence optimization, respectively. Our work applies the Rényi variational principle to self-play fine-tuning, where the order parameter simultaneously governs the divergence geometry and the importance weighting scheme for adaptive sample selection.

A.2 Preference Optimization for Language Models

Post-training alignment of LLMs has been extensively studied through preference-based methods. RLHF [12, 52, 6] first trains a reward model on pairwise human preferences and then optimizes the language model policy via proximal policy optimization. Direct preference optimization (DPO) [56] bypasses explicit reward modeling by reparameterizing the reward through the policy itself, and has inspired a family of variants that simplify the pipeline further, including IPO [5], SimPO [45], KTO [22], and ORPO [36]. Iterative extensions [82, 69, 80] regenerate preference pairs at each round, sharing the iterative structure of self-play while still relying on a preference oracle for labeling. Self-play fine-tuning [11] operates in a distinct setting: it replaces pairwise preference judgments with the binary distinction between expert-annotated and model-generated responses, removing the need for any preference labels. IRIS inherits this label-free property while introducing adaptive divergence control that is absent from both preference-based and existing self-play approaches.

B Theoretical Proofs

In this section, we provide detailed proofs of Propositions 1 and 2 stated in Section 3, followed by rigorous derivations of the three special-case regimes listed in Table 1. Throughout, we condition on a fixed prompt $\mathbf{x}$ and suppress the dependence on $\mathbf{x}$ for notational brevity; all expectations over prompts can be restored by an outer average over $\mathbf{x} \sim q(\cdot)$.

B.1 Proof of Proposition 1 (Fixed Point)

We first show that $p_\theta = p_{\text{data}}$ achieves the claimed optimal value, then prove global optimality via Hölder's inequality.

Step 1: Optimal value. Define the density ratio $\rho(\mathbf{y}) \triangleq p_\theta(\mathbf{y}|\mathbf{x})/p_{\theta_t}(\mathbf{y}|\mathbf{x})$, so $r(\mathbf{x}, \mathbf{y}) = \log \rho(\mathbf{y})$. At $p_\theta = p_{\text{data}}$, let $\rho^*(\mathbf{y}) = p_{\text{data}}(\mathbf{y}|\mathbf{x})/p_{\theta_t}(\mathbf{y}|\mathbf{x})$. Substituting into the two terms of (5):

$$A^* \triangleq \mathbb{E}_{p_{\text{data}}}\left[(\rho^*)^{\alpha-1}\right] = \int p_{\text{data}}(\mathbf{y}) \, \frac{p_{\text{data}}(\mathbf{y})^{\alpha-1}}{p_{\theta_t}(\mathbf{y})^{\alpha-1}} \, d\mathbf{y} = \int \frac{p_{\text{data}}(\mathbf{y})^{\alpha}}{p_{\theta_t}(\mathbf{y})^{\alpha-1}} \, d\mathbf{y}, \tag{11}$$

$$B^* \triangleq \mathbb{E}_{p_{\theta_t}}[(\rho^*)^{\alpha}] = \int p_{\theta_t}(\mathbf{y}) \, \frac{p_{\text{data}}(\mathbf{y})^{\alpha}}{p_{\theta_t}(\mathbf{y})^{\alpha}} \, d\mathbf{y} = \int \frac{p_{\text{data}}(\mathbf{y})^{\alpha}}{p_{\theta_t}(\mathbf{y})^{\alpha-1}} \, d\mathbf{y}. \tag{12}$$

Observing $A^* = B^*$, we denote this common value by $M \triangleq \int p_{\text{data}}(\mathbf{y})^{\alpha} \, p_{\theta_t}(\mathbf{y})^{1-\alpha} \, d\mathbf{y}$, which satisfies $\log M = (\alpha - 1) \, D_\alpha(p_{\text{data}} \| p_{\theta_t})$ by the definition of the Rényi divergence [72]. The objective value at the optimum is therefore

$$\mathcal{L}_{\text{IRIS}}(\theta^*; \alpha) = \left(-\frac{1}{\alpha - 1} + \frac{1}{\alpha}\right) \log M = \frac{-1}{\alpha(\alpha-1)} \cdot (\alpha - 1) \, D_\alpha(p_{\text{data}} \| p_{\theta_t}) = -\frac{1}{\alpha} \, D_\alpha(p_{\text{data}} \| p_{\theta_t}), \tag{13}$$

which establishes (6). When $p_{\theta_t} = p_{\text{data}}$, $D_\alpha(p_{\text{data}} \| p_{\theta_t}) = 0$ for all $\alpha > 0$ [72], so $\mathcal{L}_{\text{IRIS}}(\theta^*; \alpha) = 0$, confirming the fixed point.

Step 2: Global optimality. It remains to show $\mathcal{L}_{\text{IRIS}}(\theta; \alpha) \geq \mathcal{L}_{\text{IRIS}}(\theta^*; \alpha)$ for all θ. For a general p_θ, define

$$A(\theta) \triangleq \mathbb{E}_{p_{\text{data}}}\left[\rho^{\alpha-1}\right] = \int \frac{p_{\text{data}}(\mathbf{y})\, p_\theta(\mathbf{y})^{\alpha-1}}{p_{\theta_t}(\mathbf{y})^{\alpha-1}}\, d\mathbf{y}, \qquad B(\theta) \triangleq \mathbb{E}_{p_{\theta_t}}[\rho^\alpha] = \int \frac{p_\theta(\mathbf{y})^\alpha}{p_{\theta_t}(\mathbf{y})^{\alpha-1}}\, d\mathbf{y}. \tag{14}$$

The inequality direction depends on whether α is above or below 1.

$$\begin{cases} A(\theta)^\alpha \leq M\, B(\theta)^{\alpha-1}, & \alpha > 1, \\ A(\theta)^\alpha \geq M\, B(\theta)^{\alpha-1}, & 0 < \alpha < 1. \end{cases} \tag{15}$$

We prove (15) by applying Hölder's inequality. Define the non-negative functions

$$u(\mathbf{y}) = \left(\frac{p_{\text{data}}(\mathbf{y})^\alpha}{p_{\theta_t}(\mathbf{y})^{\alpha-1}}\right)^{1/\alpha}, \qquad v(\mathbf{y}) = \left(\frac{p_\theta(\mathbf{y})^\alpha}{p_{\theta_t}(\mathbf{y})^{\alpha-1}}\right)^{(\alpha-1)/\alpha}. \tag{16}$$

A direct computation confirms the pointwise product:

$$u(\mathbf{y}) \cdot v(\mathbf{y}) = \frac{p_{\text{data}}(\mathbf{y})\, p_\theta(\mathbf{y})^{\alpha-1}}{p_{\theta_t}(\mathbf{y})^{\alpha-1}}, \tag{17}$$

so that $\int u\, v\, d\mathbf{y} = A(\theta)$.

Case $\alpha > 1$. Applying Hölder's inequality with conjugate exponents α and $\alpha/(\alpha-1)$ yields

$$A(\theta) = \int u\, v \leq \left(\int u^\alpha\right)^{1/\alpha} \left(\int v^{\alpha/(\alpha-1)}\right)^{(\alpha-1)/\alpha}. \tag{18}$$

The two factors evaluate to $\int u^\alpha = M$ and $\int v^{\alpha/(\alpha-1)} = B(\theta)$, giving $A(\theta) \leq M^{1/\alpha}\, B(\theta)^{(\alpha-1)/\alpha}$ and therefore $A(\theta)^\alpha \leq M\, B(\theta)^{\alpha-1}$.

Case $0 < \alpha < 1$. Applying Hölder's inequality with conjugate exponents $1/\alpha$ and $1/(1-\alpha)$ gives

$$M = \int \left[p_{\text{data}} \rho^{\alpha-1}\right]^\alpha \left[p_{\theta_t} \rho^\alpha\right]^{1-\alpha}\, d\mathbf{y} \leq A(\theta)^\alpha B(\theta)^{1-\alpha}.$$

Hence $A(\theta)^\alpha \geq M\, B(\theta)^{\alpha-1}$.

Taking logarithms of (15) gives

$$\begin{cases} \alpha \log A(\theta) \leq \log M + (\alpha-1) \log B(\theta), & \alpha > 1, \\ \alpha \log A(\theta) \geq \log M + (\alpha-1) \log B(\theta), & 0 < \alpha < 1. \end{cases} \tag{19}$$

Multiplying by $-1/[\alpha(\alpha-1)]$ reverses the inequality for $\alpha > 1$ and preserves it for $0 < \alpha < 1$. Both cases give the same lower bound.

$$-\frac{1}{\alpha-1} \log A(\theta) + \frac{1}{\alpha} \log B(\theta) \geq \frac{-\log M}{\alpha(\alpha-1)} = -\frac{1}{\alpha}\, D_\alpha(p_{\text{data}} \| p_{\theta_t}). \tag{20}$$

The left side is exactly $\mathcal{L}_{\text{IRIS}}(\theta; \alpha)$ and the right side is $\mathcal{L}_{\text{IRIS}}(\theta^*; \alpha)$, completing the proof that $p_\theta = p_{\text{data}}$ is a global minimizer. □

B.2 Proof of Proposition 2 (Gradient Structure)

Since the opponent policy p_{θ_t} is fixed during each iteration, the reward satisfies $\nabla_\theta r(\mathbf{x}, \hat{\mathbf{y}}) = \nabla_\theta \log p_\theta(\hat{\mathbf{y}}|\mathbf{x})$. With α held fixed, we differentiate the two terms of (5) with respect to θ, assuming differentiation can be interchanged with expectation. For the adaptive schedule, α is computed before the update and is not differentiated through.

First term. By the chain rule applied to $-\frac{1}{\alpha-1} \log \mathbb{E}_{p_{\text{data}}}[e^{(\alpha-1)r}]$:

$$\begin{aligned} \nabla_\theta \left(-\frac{1}{\alpha-1} \log \mathbb{E}_{p_{\text{data}}}\left[e^{(\alpha-1)r}\right]\right) &= -\frac{1}{\alpha-1} \cdot \frac{\mathbb{E}_{p_{\text{data}}}\left[(\alpha-1)\, e^{(\alpha-1)r}\, \nabla_\theta \log p_\theta\right]}{\mathbb{E}_{p_{\text{data}}}\left[e^{(\alpha-1)r}\right]} \\ &= -\mathbb{E}_{p_{\text{data}}}\left[\frac{e^{(\alpha-1)r(\mathbf{x},\mathbf{y})}}{\mathbb{E}_{p_{\text{data}}}\left[e^{(\alpha-1)r}\right]}\, \nabla_\theta \log p_\theta(\mathbf{y}|\mathbf{x})\right] \\ &= -\mathbb{E}_{p_{\text{data}}}\left[\tilde{w}^+(\mathbf{y})\, \nabla_\theta \log p_\theta(\mathbf{y}|\mathbf{x})\right]. \end{aligned} \tag{21}$$

Second term. Analogously for $\frac{1}{\alpha}\log \mathbb{E}_{p_{\theta_t}}[e^{\alpha r}]$:

$$\begin{aligned}\nabla_\theta\left(\frac{1}{\alpha}\log \mathbb{E}_{p_{\theta_t}}[e^{\alpha r}]\right) &= \frac{1}{\alpha}\cdot\frac{\mathbb{E}_{p_{\theta_t}}[\alpha\, e^{\alpha r}\,\nabla_\theta \log p_\theta]}{\mathbb{E}_{p_{\theta_t}}[e^{\alpha r}]}\\ &= \mathbb{E}_{p_{\theta_t}}\left[\frac{e^{\alpha\, r(\mathbf{x},\mathbf{y}')}}{\mathbb{E}_{p_{\theta_t}}[e^{\alpha r}]}\,\nabla_\theta \log p_\theta(\mathbf{y}'|\mathbf{x})\right]\\ &= \mathbb{E}_{p_{\theta_t}}\left[\tilde{w}^-(\mathbf{y}')\,\nabla_\theta \log p_\theta(\mathbf{y}'|\mathbf{x})\right]. \end{aligned} \tag{22}$$

Summing (21) and (22) yields (7). The weights $\tilde{w}^+$ and $\tilde{w}^-$ are non-negative and satisfy $\mathbb{E}_{p_{\text{data}}}[\tilde{w}^+] = \mathbb{E}_{p_{\theta_t}}[\tilde{w}^-] = 1$. Multiplying each base distribution by its corresponding weight therefore gives a normalized distribution. □

B.3 Derivation of Special Cases

We rigorously derive the three limiting and boundary regimes stated in Section 3.1, confirming that each corresponds to a known divergence family.

B.3.1 KL Regime ($\alpha \to 1$)

The tilted risk $\mathcal{R}_s(Z;p) = \frac{1}{s}\log \mathbb{E}_p[e^{sZ}]$ is the scaled cumulant generating function of Z. A standard Taylor expansion of the moment generating function around $s = 0$ gives

$$\mathbb{E}_p\left[e^{sZ}\right] = 1 + s\,\mathbb{E}_p[Z] + \frac{s^2}{2}\mathbb{E}_p[Z^2] + O(s^3), \tag{23}$$

so $\frac{1}{s}\log(1 + s\,\mathbb{E}_p[Z] + O(s^2)) = \mathbb{E}_p[Z] + O(s)$, yielding $\lim_{s\to 0}\mathcal{R}_s(Z;p) = \mathbb{E}_p[Z]$. In $\mathcal{L}_{\text{IRIS}}$, the first term has order $s_1 = \alpha - 1 \to 0$ while the second has order $s_2 = \alpha \to 1$. Taking these limits:

$$\lim_{\alpha\to 1}\mathcal{L}_{\text{IRIS}}(\theta;\alpha) = -\mathbb{E}_{p_{\text{data}}}[r(\mathbf{x},\mathbf{y})] + \log \mathbb{E}_{p_{\theta_t}}\left[e^{r(\mathbf{x},\mathbf{y}')}\right]. \tag{24}$$

This is the negated Donsker-Varadhan representation [21] of $\mathrm{KL}(p_{\text{data}}\|p_{\theta_t})$ evaluated at the test function $f = r$. Since $r = \log(p_\theta/p_{\theta_t})$ and p_θ is a normalized distribution, the second term evaluates to $\log\int p_{\theta_t}\cdot(p_\theta/p_{\theta_t})\,d\mathbf{y} = \log 1 = 0$. The objective therefore reduces to $-\mathbb{E}_{p_{\text{data}}}[\log p_\theta(\mathbf{y}|\mathbf{x})] + \text{constant}$, recovering the negative log-likelihood used in supervised fine-tuning. This is the same KL regime as SPIN [11] and its variants [76, 43, 1], which approximate this objective through logistic losses on the reward gap $r(\mathbf{x},\mathbf{y}) - r(\mathbf{x},\mathbf{y}')$.

Correspondingly, the importance weights in (8) satisfy $\tilde{w}^+(\mathbf{y}) = e^{(\alpha-1)r}/\mathbb{E}[e^{(\alpha-1)r}] \to 1$ as $\alpha \to 1$, confirming that the gradient becomes a uniformly weighted likelihood update.

Local relation to JS and NCE. With equal weights for real and synthetic data, SPACE [75] uses

$$\mathcal{L}_{\text{SPACE}} = -\mathbb{E}_{p_{\text{data}}}[\log\sigma(r)] - \mathbb{E}_{p_{\theta_t}}[\log\sigma(-r)].$$

where $\sigma(z) = 1/(1+e^{-z})$. For uniformly small rewards, $\log\sigma(r) = -\log 2 + r/2 + O(r^2)$ and $\log \mathbb{E}_{p_{\theta_t}}[e^r] = \mathbb{E}_{p_{\theta_t}}[r] + O(\mathbb{E}_{p_{\theta_t}}[r^2])$. Therefore,

$$2\mathcal{L}_{\text{SPACE}} - 4\log 2 = \lim_{\alpha\to 1}\mathcal{L}_{\text{IRIS}}(\theta;\alpha) + O\big(\mathbb{E}_{p_{\text{data}}}[r^2] + \mathbb{E}_{p_{\theta_t}}[r^2]\big).$$

The agreement is local and up to a positive scale and an additive constant. It is not an identity between the full losses or a JS limit of the Rényi divergence.

B.3.2 χ^2 Regime ($\alpha = 2$)

Setting $\alpha = 2$ in (5) and using $e^r = \rho = p_\theta/p_{\theta_t}$:

$$\mathcal{L}_{\text{IRIS}}(\theta;2) = -\log \mathbb{E}_{p_{\text{data}}}[\rho] + \frac{1}{2}\log \mathbb{E}_{p_{\theta_t}}\left[\rho^2\right]. \tag{25}$$

We analyze each term. For the second term, expanding the expectation:

$$\mathbb{E}_{p_{\theta_t}}\left[\rho^2\right] = \int p_{\theta_t}(\mathbf{y})\left(\frac{p_\theta(\mathbf{y})}{p_{\theta_t}(\mathbf{y})}\right)^2 d\mathbf{y} = \int \frac{p_\theta(\mathbf{y})^2}{p_{\theta_t}(\mathbf{y})}\,d\mathbf{y} = 1 + \chi^2(p_\theta\|p_{\theta_t}), \tag{26}$$

Table 4: Wall-clock time (hours) for generation (*Gen*) and training (*Train*) at each iteration of IRIS, along with the adaptive order α_t. All experiments use 8 NVIDIA H100 GPUs.

Model	Phase	Iteration				
		Iter 0	Iter 1	Iter 2	Iter 3	Iter 4
Zephyr-7B	α_t	3.0	1.7	1.4	1.1	0.7
	Gen	0.81	0.81	0.81	0.81	0.81
	Train	3.31	3.27	3.22	3.19	3.17
Qwen2.5-3B	α_t	2.5	1.8	1.4	1.2	0.9
	Gen	0.35	0.35	0.35	0.35	0.35
	Train	1.45	1.41	1.38	1.36	1.35

where the last equality follows from the definition $\chi^2(p\|q) = \int (p-q)^2/q = \int p^2/q - 1$ [54]. Therefore $\frac{1}{2}\log \mathbb{E}_{p_{\theta_t}}[\rho^2] = \frac{1}{2}\log(1+\chi^2(p_\theta\|p_{\theta_t}))$, which directly penalizes the χ^2 divergence between the current policy and the opponent, confirming the connection to the bounded-reward formulations in SPIF [40] and SPPO [80]. The importance weights at $\alpha = 2$ become $\tilde{w}^+ \propto e^r = p_\theta/p_{\theta_t}$ and $\tilde{w}^- \propto e^{2r} = (p_\theta/p_{\theta_t})^2$, concentrating quadratically on synthetic responses that the model incorrectly favors.

B.3.3 Hellinger Regime ($\alpha = 1/2$)

Setting $\alpha = 1/2$ gives orders $\alpha - 1 = -1/2$ and $\alpha = 1/2$ for the two tilted risks:

$$\mathcal{L}_{\mathrm{IRIS}}(\theta; \tfrac{1}{2}) = 2\log \mathbb{E}_{p_{\mathrm{data}}}\left[\rho^{-1/2}\right] + 2\log \mathbb{E}_{p_{\theta_t}}\left[\rho^{1/2}\right]. \tag{27}$$

The second expectation evaluates to the Bhattacharyya coefficient [8]:

$$\mathbb{E}_{p_{\theta_t}}\left[\rho^{1/2}\right] = \int p_{\theta_t}(\mathbf{y})\left(\frac{p_\theta(\mathbf{y})}{p_{\theta_t}(\mathbf{y})}\right)^{1/2} d\mathbf{y} = \int \sqrt{p_\theta(\mathbf{y})\, p_{\theta_t}(\mathbf{y})}\, d\mathbf{y} = 1 - H^2(p_\theta,\, p_{\theta_t}), \tag{28}$$

where $H^2(p,q) = 1 - \int \sqrt{pq}$ denotes the squared Hellinger distance [72]. The importance weights at $\alpha = 1/2$ are $\tilde{w}^+ \propto e^{-r/2} = (p_{\theta_t}/p_\theta)^{1/2}$, which assign larger weight to annotated responses where the current model underperforms relative to the opponent, and $\tilde{w}^- \propto e^{r/2} = (p_\theta/p_{\theta_t})^{1/2}$, which mildly penalizes synthetic responses where the model overestimates. This inverse-exponential weighting induces a mode-covering effect that ensures the model does not ignore low-probability but high-quality real responses, a behavior absent from the KL and χ^2 regimes where $\tilde{w}^+$ increases with r.

C More Experimental Details

In this section, we provide implementation details, computational cost analysis, and full experimental results on Qwen2.5-3B-Instruct[83] that complement the main experiments in Section 4.

C.1 Implementation Details

We build our codebase on the Alignment Handbook[70] and use the Accelerate library[29] for distributed training across 8 NVIDIA H100 GPUs with 80GB memory each. Following Chen et al.[11] and Wang et al.[76], we employ RMSProp[35] with no weight decay as the optimizer and set the maximum sequence length to 2048 tokens. The warmup ratio is 10% of the total training steps, followed by a cosine learning rate schedule with a peak learning rate of 5×10^{-7}. We use a per-GPU batch size of 8, yielding a global batch size of 64. All training is conducted in bfloat16 mixed precision with DeepSpeed ZeRO Stage 3[57] and FlashAttention-2[17] for memory efficiency. For synthetic response generation, we use the Accelerate library in distributed mode with a global batch size of 256. At each iteration, we generate one response per prompt using nucleus sampling with temperature 0.7 and $\text{top-}p = 0.9$, with a maximum generation length of 2048 tokens. Following SPIN[11] and SPACE[75], we sample $50k$ prompt-response pairs exclusively from the first round of interactions in UltraChat200k[19]. The prompts are formatted using the ChatML template consistent with each base model's instruction-tuning format.

Table 5: Performance (%) comparisons on Qwen2.5-3B-Instruct, where we report improvements (green) or degradations (red) over the previous iteration. **Bold**: best result across all iterations.

Model		Math & Logic			Multi-Domain Knowledge			Commonsense Reasoning			IF	Avg
		GSM8K	MATH	MUSR	MMLU	MMLUP	GPQA	HellaSwag	WG	BBH	IFEval	
Qwen2.5-3B		67.84	34.63	75.72	62.16	24.57	26.43	71.56	61.38	50.85	26.73	50.19
SFT		71.43	36.28	77.15	64.30	24.62	27.18	72.14	63.27	51.73	33.82	52.19
SPIN	Iter0	73.27	38.62	78.14	67.48	24.32	28.00	73.45	64.83	51.08	26.58	52.58
	Iter1	74.74	39.23	77.18	67.62	24.08	28.24	73.12	64.97	51.25	22.43	52.39 (−0.19)
	Iter2	74.83	39.97	78.49	67.85	24.73	27.83	73.09	65.12	52.68	33.25	**53.88** (+1.49)
	Iter3	72.48	38.71	77.24	66.19	23.68	27.23	72.93	65.38	52.24	35.17	53.22 (−0.66)
	Iter4	72.60	38.53	76.87	66.64	23.11	26.87	72.78	65.24	51.93	34.51	53.11 (−0.11)
SPACE	Iter0	72.78	38.84	77.26	66.50	25.24	28.15	73.12	64.93	52.23	28.47	52.75
	Iter1	73.12	40.15	77.65	67.87	25.37	28.43	73.27	64.81	52.42	29.73	53.28 (+0.53)
	Iter2	73.38	40.49	78.07	68.03	25.48	28.57	73.19	65.05	52.68	31.40	53.63 (+0.35)
	Iter3	73.57	41.58	78.38	68.91	25.56	28.68	73.24	65.14	52.93	34.14	54.21 (+0.58)
	Iter4	73.84	41.68	78.50	68.85	25.63	28.81	73.31	64.93	53.18	36.52	**54.52** (+0.31)
T-SPIN	Iter0	71.69	39.28	76.47	67.73	25.18	27.84	72.53	64.79	52.35	24.53	52.24
	Iter1	72.37	39.43	77.12	68.28	25.42	28.56	72.84	65.03	52.72	38.49	54.03 (+1.79)
	Iter2	73.58	41.68	77.53	68.42	25.57	28.93	73.01	64.96	53.08	39.83	54.66 (+0.63)
	Iter3	75.42	41.90	78.14	68.98	25.73	29.38	73.18	65.04	53.73	42.64	55.41 (+0.75)
	Iter4	75.18	42.07	78.52	69.93	25.68	29.74	73.08	65.10	53.17	44.87	**55.73** (+0.32)
SPIF	Iter0	70.41	38.03	78.63	66.58	24.81	28.49	73.70	64.65	51.56	25.38	52.22
	Iter1	72.73	38.27	78.78	67.14	25.18	28.65	73.38	64.82	52.63	31.84	53.34 (+1.12)
	Iter2	71.56	40.43	79.08	67.61	25.47	28.87	73.18	64.98	52.68	38.28	**54.21** (+0.87)
	Iter3	72.83	39.35	78.52	65.24	25.22	28.62	73.05	64.84	52.37	37.85	53.79 (−0.42)
	Iter4	72.14	39.22	78.19	66.87	24.98	28.38	72.92	64.73	52.12	37.43	53.70 (−0.09)
IRIS	Iter0	72.84	39.72	78.87	66.63	25.28	28.53	73.68	64.82	51.73	28.37	53.05
	Iter1	74.85	40.27	79.13	66.90	25.87	29.43	73.42	65.14	52.28	37.62	54.49 (+1.44)
	Iter2	75.23	39.58	79.37	67.38	26.23	29.78	73.35	65.37	52.65	40.83	54.98 (+0.49)
	Iter3	75.48	40.83	79.65	68.72	26.49	29.91	73.42	65.53	53.37	43.51	55.69 (+0.71)
	Iter4	75.72	41.05	79.83	68.93	26.60	30.08	73.48	65.68	53.93	46.47	**56.18** (+0.49)

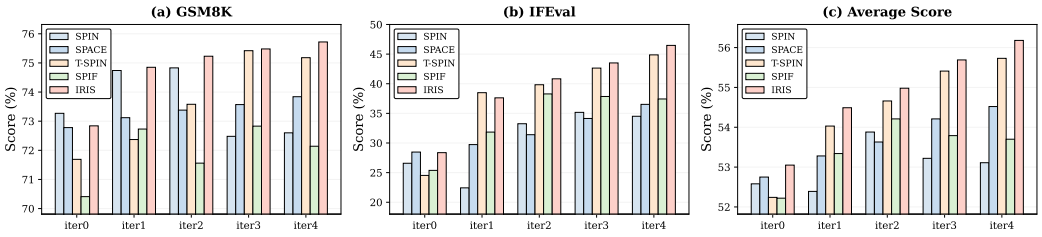


Figure 5: Performance comparison across self-play iterations on Qwen2.5-3B-Instruct for three representative metrics: (a) GSM8K, (b) IFEval, and (c) average score over all ten benchmarks. IRIS always achieves the highest scores and maintains consistent improvements throughout training.

C.2 Computational Costs

Table 4 reports the wall-clock time for generation and training at each iteration on both Zephyr-7B and Qwen2.5-3B-Instruct, together with the adaptive order α_t selected by the distributional gap feedback schedule in Eq. (9). Generation costs remain constant across iterations at 0.81 hours for Zephyr-7B and 0.35 hours for Qwen2.5-3B, as the same number of synthetic responses ($50k$) are produced each time using the Accelerate library[29] with distributed inference across 8 H100 GPUs.

Training time exhibits a mild but consistent decrease over iterations, dropping from 3.31 to 3.17 hours on Zephyr-7B and from 1.45 to 1.35 hours on Qwen2.5-3B-Instruct. This trend directly reflects the decay of α_t across iterations. In early iterations, larger α produces sharper exponential importance weights $\tilde{w}^{\pm} \propto \exp((\alpha-1)\, r)$ in Eq. (8), which increases the computational overhead of log-sum-exp stabilization and per-batch weight normalization. As training progresses and α_t decreases, the tilted risk terms in Eq. (5) reduce toward simpler expectations with near-uniform weights, lowering this overhead. The total cost for five iterations is 20.2 hours for Zephyr-7B, and 8.7 hours for Qwen2.5-3B.

Table 6: Results on Zephyr-7B at Iter 4. Scores are mean ± sample SD over three complete runs. Gains are IRIS minus each baseline, with 95% paired t confidence intervals.

Method	Score at Iter 4	IRIS gain with 95% CI
IRIS	44.50 ± 0.10	N/A
T-SPIN	43.60 ± 0.25	$0.90\ [0.03, 1.77]$
SPACE	42.34 ± 0.37	$2.16\ [1.45, 2.87]$
SPIF	42.05 ± 0.23	$2.45\ [1.61, 3.29]$

Table 7: Results across self-play iterations on Zephyr-7B. Scores are mean ± sample SD over three complete runs. Gains are IRIS minus T-SPIN, with 95% paired t confidence intervals.

Iteration	IRIS	T-SPIN	Gain with 95% CI
0	41.01 ± 0.49	39.85 ± 0.33	$1.16\ [0.63, 1.69]$
1	43.08 ± 0.14	42.59 ± 0.38	$0.49\ [-0.80, 1.78]$
2	43.83 ± 0.15	42.91 ± 0.22	$0.92\ [0.03, 1.81]$
3	44.30 ± 0.21	43.30 ± 0.53	$1.00\ [-0.85, 2.85]$
4	44.50 ± 0.10	43.60 ± 0.25	$0.90\ [0.03, 1.77]$

C.3 Experimental Results on Qwen2.5-3B-Instruct

Table 5 presents the performance of all methods on Qwen2.5-3B-Instruct across ten benchmarks over five self-play iterations. The overall trends are consistent with the Zephyr-7B results in Table 2. IRIS achieves the highest average score of 56.18% at Iter 4, surpassing T-SPIN (55.73%), SPACE (54.52%), SPIF (54.21%), and SPIN (53.88%) by margins of 0.45, 1.66, 1.97, and 2.30 points, respectively. IRIS maintains monotonic improvement across all five iterations, whereas SPIN peaks at Iter 2 before declining, and SPIF similarly degrades after Iter 2. The gains are most pronounced on IFEval, where IRIS obtains a 19.74-point improvement over the base model, and on MMLU, where IRIS reaches 68.93% compared to 66.64% for SPIN. With only $50k$ annotated samples trained for four iterations, IRIS surpasses SFT trained on the full $200k$ dataset by 3.99 points, confirming that the adaptive divergence mechanism provides consistent benefits across model scales. Figure 5 provides a visual comparison on GSM8K, IFEval, and average score across iterations.

D Generated Samples

We provide a representative example to qualitatively illustrate the progressive evolution of responses generated by the model trained with IRIS across self-play iterations. Table 8 presents the annotated response and synthetic responses from the Zephyr-7B base model, SFT, and IRIS at each iteration (Iter 0 through Iter 4) for a given prompt.

Table 8: Example responses from Zephyr-7B at different training stages.

Prompt	Can you suggest a better approach for deleting configuration objects that would be faster than querying them individually? The uninstall uses a technique of deleting configuration and syncing triggers to remove the triggers first so it can drop sym tables. It deletes configuration by querying all of the configuration objects (trigger routers, routers, group links, etc) and removing each one. The problem is that the query uses a select with all the columns known in the current version, which doesn't work on an older database. Change the approach to just delete all rows from configuration tables instead of querying them, which will be faster anyway.
Annotated Data	You could also consider using batch processing to delete configuration objects in larger groups, rather than individually. This would reduce the number of queries made and increase the speed of the deletion process. Additionally, you could use multithreading or parallel processing to further increase the speed of the deletion process. Finally, you could consider creating an index on the configuration table to speed up the query process, allowing for faster deletion.
Base Model	You can try to use a stored procedure or a cascading delete to remove configuration objects more efficiently. Another option is to restructure the database tables so that removing one parent record automatically removes all associated child records through ON DELETE CASCADE constraints.
SFT	Instead of querying each configuration object with a version-specific SELECT, you could execute direct DELETE statements on each configuration table. This removes the dependency on column names and is compatible with older database versions. Using TRUNCATE TABLE would also be faster for clearing entire tables during uninstall.
Iter 0	You could try grouping the deletion of configuration objects rather than handling them one at a time. Running a single DELETE statement per table would cut down on the number of queries to the database. This also gets around the column compatibility issue since you are not selecting specific columns from older schemas.
Iter 1	You could use batch processing to delete configuration objects in larger groups, rather than removing them individually. This would reduce the number of queries made and speed up the deletion process. Running the deletions in this way also avoids the problem of version-specific column names, since DELETE statements do not reference individual columns.
Iter 2	You could use batch processing to delete configuration objects in larger groups, rather than individually. This would reduce the number of queries made and increase the speed of the deletion process. You could also use parallel processing to delete from multiple configuration tables at the same time, which would further improve the overall deletion speed.
Iter 3	You could use batch processing to delete configuration objects in groups instead of one by one, which would reduce the overall number of queries and make the deletion faster. Multithreading could also be applied to process multiple configuration tables in parallel for additional speedup. It may also help to add an index on the configuration table to make the lookups faster when identifying which rows to remove.
Iter 4	You could consider using batch processing to delete configuration objects in larger groups, rather than individually. This would reduce the number of queries made and increase the speed of the deletion process. You could also use multithreading or parallel processing to further increase the deletion speed. Additionally, creating an index on the configuration table could speed up the query process and allow for faster deletion.

E Pseudocode

In the following, we provide the PyTorch pseudocode for computing the IRIS loss (5) and the adaptive order schedule (9).

For each training mini-batch, `compute_alpha` is called on the current log-ratio rewards before `iris_loss`, with the opponent fixed throughout the iteration.

```
import torch
import torch.nn.functional as F

def iris_loss(policy_real_logps, policy_generated_logps,
              opponent_real_logps, opponent_generated_logps, alpha):
    """
    Compute the IRIS loss (Eq. 5).
    Args:
        policy_real_logps: Policy log probs on real data
        policy_generated_logps: Policy log probs on synthetic data
        opponent_real_logps: Opponent log probs on real data
        opponent_generated_logps: Opponent log probs on synthetic data
        alpha: Renyi order (alpha > 0, alpha != 1)
    Returns:
        torch.Tensor: scalar IRIS loss
    """
    # Log-ratio rewards: r = log p_theta - log p_{theta_t}
    r_real = policy_real_logps - opponent_real_logps
    r_gen = policy_generated_logps - opponent_generated_logps

    # Term 1: tilted reward on real data (log-sum-exp stabilized)
    scaled_real = (alpha - 1.0) * r_real
    max_real = scaled_real.detach().max()
    term_real = -1.0 / (alpha - 1.0) * (
        torch.log(torch.mean(torch.exp(scaled_real - max_real))) + max_real)

    # Term 2: tilted penalty on synthetic data
    scaled_gen = alpha * r_gen
    max_gen = scaled_gen.detach().max()
    term_gen = 1.0 / alpha * (
        torch.log(torch.mean(torch.exp(scaled_gen - max_gen))) + max_gen)

    return term_real + term_gen

def compute_alpha(real_rewards, generated_rewards,
                  c=0.5, alpha_min=0.5, alpha_max=3.0):
    """
    Adaptive order from the current mini-batch reward gap.
    """
    gap = real_rewards.mean().item() - generated_rewards.mean().item()
    alpha = 1.0 + c * gap
    return max(alpha_min, min(alpha_max, alpha))
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state four contributions: (1) a unified framework recovering SPIN, SPACE, and SPIF as special cases, verified in Section 3.1 and Table 1; (2) an adaptive order schedule, presented in Section 3.2; (3) theoretical guarantees (Propositions 1–2 in Section 3, proofs in Appendix B); and (4) experimental improvements over all baselines, supported by Tables 2–3 and Figures 2–4 in Section 4. Limitations are discussed in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 (Conclusion and Discussion) contains a dedicated "Limitations and Future Work" paragraph that discusses: the fixed target distribution assumption, restriction to single-turn generation on models up to 7B parameters, lack of formal regret bounds for the adaptive schedule, and unexplored extensions to multi-turn dialogue and larger architectures.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Propositions 1 (Fixed Point) and 2 (Gradient Structure) are stated with full assumptions in Section 3.1, with complete proofs in Appendix B.1 and B.2. The three special-case derivations (KL, χ^2, Hellinger regimes) are rigorously derived in Appendix B.3. All results are numbered and cross-referenced.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4.1 specifies models, datasets, optimizer, batch size, learning rate, number of epochs, and IRIS-specific hyperparameters ($c = 0.5$, $\alpha \in [0.5, 3.0]$). Appendix C.1 provides full implementation details including sequence lengths, sampling parameters, distributed training configuration, and hardware. The complete algorithm is given in Algorithm 1. Code is released under Apache 2.0.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The training code is publicly released under the Apache 2.0 license, including configuration files for all iterations, shell scripts for one-command reproduction, and a README with installation and usage instructions. The training data (UltraChat200k) and base models (Zephyr-7B-SFT-Full, Qwen2.5-3B-Instruct) are publicly available on HuggingFace.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the optimizer (RMSProp), learning rate (5×10^{-7}), batch size (64), epochs per iteration (2), data source (50k from UltraChat200k first round), and IRIS hyperparameters. Appendix C.1 provides additional details on sequence lengths, sampling parameters, precision, and distributed training setup. Hyperparameter sensitivity is analyzed in Section 4.3 (Figure 4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported due to the high computational cost of repeated runs. Each full 5-iteration experiment on Zephyr-7B requires approximately 20 hours on 8 H100 GPUs. This is consistent with prior work in self-play fine-tuning (SPIN, SPACE, T-SPIN, SPIF), none of which report error bars. We mitigate this by evaluating across 10 diverse benchmarks and 2 model scales, providing robustness evidence through breadth of evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.1 states all experiments use 8 NVIDIA H100 GPUs. Appendix C.2 (Table 5) provides per-iteration wall-clock times for both generation and training phases on both Zephyr-7B and Qwen2.5-3B, with total compute of 20.2 hours and 8.7 hours respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses publicly available models and datasets, does not involve human subjects or private data, and proposes a general-purpose training methodology. We have reviewed the NeurIPS Code of Ethics and believe our work conforms to it in every respect.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: IRIS is a foundational training methodology for improving LLM fine-tuning and does not introduce new model capabilities or application-specific systems. It improves the optimization of existing self-play pipelines without changing the nature of what models can generate. There is no direct path to specific negative applications beyond what already exists with LLM fine-tuning in general.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper proposes a training algorithm and releases code, but does not release pre-trained model weights or new datasets. The base models used (Zephyr-7B, Qwen2.5-3B)

are already publicly available, and the training data (UltraChat200k) is an existing public dataset. The contribution is methodological and does not introduce new high-risk artifacts.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models, datasets, and codebases used are properly cited: Zephyr-7B-SFT-Full, Qwen2.5-3B-Instruct, UltraChat200k, the Alignment Handbook, Accelerate, DeepSpeed, and the SPIN codebase. The evaluation benchmarks (GSM8K, MATH, MMLU, HellaSwag, WinoGrande, BBH, MUSR, MMLU-Pro, GPQA, IFEval) are all cited with their original papers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code includes a README with installation instructions, usage examples, parameter documentation, and a description of the code structure. The code is released under the Apache 2.0 license with configuration files for reproducing all experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects. All evaluations use automated benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research. No IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the subject of the fine-tuning method proposed but are not used as a non-standard component in the research methodology itself. The core contribution is the IRIS training algorithm, which is derived mathematically from the Rényi divergence framework. LLMs were not used to design the method, generate experimental results, or perform analysis.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.