Bias-Variance Optimized Preference Optimization for Large Reasoning Models

Mingkang Zhu[1] **Xi Chen**[2] **Bei Yu**[1] **Hengshuang Zhao**[2] **Jiaya Jia**[3]

[1]The Chinese University of Hong Kong [2]The University of Hong Kong
[3]The Hong Kong University of Science and Technology

Abstract

Large reasoning models (LRMs) generate reasoning traces before producing final answers, yielding strong performance on multi-step reasoning tasks. However, preference alignment for LRMs remains underexplored. The ideal answer-level preference objective marginalizes over reasoning traces, but this trace-marginalized objective is computationally intractable. As a result, practical methods rely on single-trace surrogates, which can induce high gradient variance due to stochastic trace sampling. We cast this challenge as a gradient-estimator design problem and propose Bias-Variance Optimized Preference Optimization (BVPO), which combines a standard trace-based gradient estimator with an empty-trace gradient estimator from a separately sampled empty-trace branch. Under our fixed-prompt, fixed-answer conditional view, the empty-trace branch is deterministic with respect to the trace randomness studied in our analysis. We show that the composite estimator contracts the conditional reasoning-trace-noise component, admits a conditional MSE-optimal mixture, and appears in the estimator-dependent term of a standard biased-SGD bound. Empirically, BVPO improves alignment over baselines on alignment benchmarks, including AlpacaEval 2 and Arena-Hard. Although trained on general conversational data, BVPO also generally preserves reasoning performance on six math reasoning benchmarks. Empirical diagnostics further show that BVPO yields lower total gradient variance during training.

1 Introduction

Large reasoning models (LRMs), such as DeepSeek R1 [6], Gemini 2.5 [4], and GPT-o1 [14], scale test-time compute by generating intermediate reasoning traces before producing a final answer [30]. This explicit deliberation has led to large gains on multi-step and mathematically intensive tasks, and reinforcement learning with verifiable rewards (RLVR) further improves this capability [1, 6, 29, 36]. However, preference alignment, important for deployment, remains underexplored for LRMs.

Existing public discussion of preference optimization for LRMs with explicit reasoning traces is largely limited to brief remarks in technical reports accompanying foundational models [6, 14]. Existing alignment methods, from RLHF [24, 28, 40] to direct preference optimization (DPO) [26] and its variants [7, 22, 25, 37], were developed for conventional LLMs that do not externalize lengthy reasoning traces. When applied naively to LRMs, these methods inherit an additional source of instability: *trace-induced gradient variance*. This creates a structural mismatch between conventional preference-optimization methods and LRMs, since these methods were not designed to explicitly account for gradient variance induced by trace sampling.

To make this precise, we study preference optimization for LRMs under the trace-answer factorization $\pi_\theta(\tau, y \mid x) = \pi_\theta(\tau \mid x)\,\pi_\theta(y \mid x, \tau)$, where the model first generates a reasoning trace τ and then produces the final answer y. The trace-marginalized preference objective compares marginal answer probabilities $\pi_\theta(y \mid x) = \sum_\tau \pi_\theta(\tau, y \mid x)$, so that all possible traces leading to the same

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

answer are accounted for. However, this sum spans an exponentially large set of traces, making it computationally intractable. In practice, it is typically replaced by a single sampled trace, yielding a trace-based gradient estimator g_t [6]. This surrogate is computationally convenient, but relative to the intractable trace-marginalized target, it inherits estimator randomness from trace sampling; this effect is formalized under a fixed-prompt and fixed-answer conditional view in Section 4 and probed empirically through observable diagnostics in Section 5.3.

We cast this as a gradient-estimation problem under the bias–variance trade-off [8] and propose **Bias-Variance Optimized Preference Optimization (BVPO)** to reduce trace-induced gradient variance. Specifically, BVPO combines the standard trace-based gradient estimator g_t with a prompt-paired empty-trace gradient estimator g_e, yielding $g_c(\alpha) = \alpha g_t + (1-\alpha) g_e$. Here "prompt-paired" means that the two branches share the same prompt, but use separately sampled preference pairs under trace-based and empty-trace generation. Under our fixed prompt-and-answer conditional view, g_e is deterministic with respect to trace randomness isolated in our analysis, therefore removes the corresponding conditional trace-noise component; by contrast, g_t remains stochastic through sampled reasoning traces. The mixing weight α can then be chosen to optimize the conditional mean squared error (MSE) with respect to the ideal marginal gradient, thereby balancing bias and variance. Our analysis shows that the composite estimator contracts conditional trace noise, admits a conditional MSE-optimal mixture, and appears in the estimator-dependent term of a standard biased-SGD bound. Since the conditional trace-noise quantity is not directly measurable in practice, we complement the theory with observable empirical diagnostics and controlled comparisons.

Extensive experiments on alignment benchmarks show that BVPO consistently outperforms strong baselines on the tested alignment benchmarks. Since preference alignment is typically the final stage before deployment, we further examine whether BVPO preserves the reasoning capability of LRMs. Despite being trained exclusively on general conversational data, BVPO generally preserves the reasoning performance of the base models on a diverse suite of math reasoning benchmarks.

Our key contributions are summarized as follows:

- We formulate preference optimization for LRMs with an intractable trace-marginalized objective as a gradient-estimator design problem, and propose BVPO, a prompt-paired composite estimator that combines trace-based and empty-trace gradient estimators.
- We show that the composite estimator contracts the conditional trace-noise component induced by trace sampling, derive a closed-form conditional MSE-optimal mixing coefficient, and connect the resulting bias–variance trade-off to the estimator-dependent term in a standard biased-SGD bound.
- Extensive experiments show that BVPO improves alignment over strong baselines, with gains of up to 8.9 points on Arena-Hard and 13.3 points on AlpacaEval 2. Moreover, despite being trained only on general conversational data, BVPO generally preserves reasoning performance on six math reasoning benchmarks.

2 Related work

Large reasoning models. Large reasoning models (LRMs), such as DeepSeek R1 [6], Gemini 2.5 [4], and GPT-o1 [14], represent a new paradigm in large language models by explicitly scaling test-time computation through reasoning traces [30]. Unlike conventional LLMs that directly generate final answers, LRMs produce explicit reasoning traces followed by answer production. This design has led to substantial gains on complex, multi-step, and mathematically intensive tasks [6, 29]. Recent advances further improve this capability using RLVR, particularly in math reasoning [1, 6, 29, 36]. Despite this progress, the preference alignment of LRMs remains underexplored. Existing discussions of alignment for LRMs are sparse and typically confined to brief remarks in technical reports accompanying foundational models [6]. Our work focuses on this gap by analyzing preference optimization for LRMs through the lens of gradient estimation, and by characterizing the instability caused by stochastic trace sampling.

Reinforcement learning from human feedback (RLHF). RLHF is a central paradigm for preference alignment [24, 28, 37–40]. Recent work has explored alternatives to training reward models. A prominent example is Direct Preference Optimization (DPO) [26], which derives a closed-form preference loss corresponding to a PPO-induced reward and enables stable training without a learned

reward model. DPO and its variants have demonstrated strong performance across a wide range of tasks [13, 23, 31]. Several extensions further refine the DPO framework: R-DPO [25] addresses sensitivity to sequence length, and SimPO [22] removes the reference model and improves consistency with the sampling distribution. However, these methods are developed for conventional LLMs that directly generate final answers. When applied to LRMs that externalize long and stochastic reasoning traces, the resulting gradient estimators inherit an additional source of randomness. In particular, conditioning preference optimization on a single sampled trace can introduce *trace-induced gradient-estimator variability* relative to the intractable trace-marginalized gradient. To the best of our knowledge, this trace-induced estimator instability has not been addressed by existing preference optimization methods. Our work studies this gap by formulating preference optimization for LRMs as a stochastic gradient-estimation problem, and analyzing how the bias-variance trade-off of the resulting gradient estimator affects training stability.

3 Preference optimization for LRMs

We study preference optimization for LRMs with an intractable trace-marginalized objective. We formalize the gradient estimation challenge induced by stochastic reasoning traces, and then introduce **Bias-Variance Optimized Preference Optimization** (BVPO), which uses an empty-trace branch to address trace-induced gradient variance.

3.1 Preliminaries

Direct preference optimization (DPO). DPO [26] aligns language models to preference data without explicit reward modeling. Given a preference dataset $\mathcal{D} = \{(x, y^+, y^-)\}$ and using the Bradley–Terry preference model [3], the probability that response y^+ is preferred to y^- is written as

$$\Pr(y^+ \succ y^- \mid x) = \sigma\left(r(x, y^+) - r(x, y^-)\right),$$

where $\sigma(\cdot)$ is the sigmoid function and $r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$, with reference policy π_{ref} and scaling constant $\beta > 0$. DPO then minimizes the loss:

$$\mathcal{L}_{\text{DPO}}(\theta) = \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}\ell_{\text{DPO}}(\theta; x, y^\pm), \quad \text{where}$$

$$\ell_{\text{DPO}}(\theta; x, y^\pm) = -\log\sigma\Big(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}\Big).$$

Large reasoning models (LRMs). An LRM is modeled as a stochastic policy π_θ parameterized by θ. Given a prompt x, the model first generates a reasoning trace τ and then produces a final answer y, inducing a distribution over trajectories (τ, y): $\pi_\theta(\tau, y \mid x) = \pi_\theta(\tau \mid x)\, \pi_\theta(y \mid x, \tau)$. The marginal probability of the final answer is obtained by summing over all possible reasoning traces:

$$\pi_\theta(y \mid x) = \sum_\tau \pi_\theta(\tau, y \mid x). \tag{1}$$

3.2 Problem formulation: the estimator gap

We study preference optimization for LRMs in the setting where preferences are determined by final-answer quality, following [6]. Applying DPO in this setting requires accounting for the combinatorially large space of reasoning traces. The resulting trace-marginalized objective is well defined but intractable to optimize directly, making gradient estimation computationally challenging. We therefore contrast the ideal trace-marginalized objective and gradient with tractable surrogates.

Ideal objective: marginal preference loss. The ideal objective applies DPO to the *marginal* likelihood of final answers. For a prompt x with preferred and dispreferred final answers (y^+, y^-), we define the **marginal preference loss**

$$\ell_m(\theta; x, y^\pm) = -\log\sigma\Big(\beta \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \beta \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}\Big). \tag{2}$$

Here, $\pi_\theta(y \mid x) = \sum_\tau \pi_\theta(\tau, y \mid x)$ is the marginal probability obtained by summing over all reasoning traces leading to the final answers; the same marginalization applies to $\pi_{\mathrm{ref}}(y \mid x)$. The corresponding population objective is then defined as the expectation over the preference dataset $\mathcal{D}$:

$$\mathcal{L}(\theta) := \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}\big[\ell_m(\theta; x, y^\pm)\big]. \tag{3}$$

While $\mathcal{L}(\theta)$ is ideal, evaluating it and its gradient is intractable because it sums over all traces.

Practical proxy: trace-based stochastic gradient estimator. To bypass this intractability, standard approaches (e.g., [6]) optimize a **trace-based surrogate**. Instead of marginalizing over all possible reasoning traces,

the method uses a single sampled reasoning trace τ and evaluates the preference loss on the resulting trajectory. This corresponds to the gradient estimator g_t in our setting:

$$g_t(\theta; x, (\tau, y)^\pm) = \nabla_\theta \ell_{\mathrm{DPO}}\left(\theta; x, (\tau, y)^\pm\right), \tag{4}$$

where $(\tau, y)^\pm \sim \pi_{\mathrm{ref}}(\cdot \mid x)$ denotes the preferred/dispreferred pair sampled for prompt x. Thus, g_t is a tractable single-trace gradient surrogate used in place of the intractable marginal gradient of Equation (2). Relative to the ideal trace-marginalized gradient, it remains stochastic because different sampled traces for the same prompt can induce different trajectory-level gradients. This trace-sampling variability is the source of estimator instability studied in our theory.

3.3 Bias-variance optimized preference optimization (BVPO)

The empty-trace gradient estimator. To reduce estimator randomness from stochastic trace sampling, we introduce a *prompt-paired* empty-trace gradient estimator g_e. This estimator is constructed from the same prompt as g_t by explicitly conditioning the model on an *empty reasoning trace* $\tau = \emptyset$, forcing the model to generate the final answer y directly via the shortcut pathway $x \to y$. Preference labels for these answers are assigned using the same method as in the trace-based branch. Here "prompt-paired" means that the two branches share the same prompt. The empty-trace branch closes the reasoning span using the model-specific chat template (Appendix B.3). The bias of g_e relative to the marginal gradient is explicitly included in the conditional MSE analysis in Section 4.2.

Under the fixed prompt-and-answer conditional view in Section 4, g_e is deterministic with respect to the trace randomness isolated in our analysis. Formally,

$$g_e(\theta; x, y_e^\pm) := \nabla_\theta \ell_{\mathrm{DPO}}(\theta; x, y_e^\pm).$$

The composite gradient estimator. We combine the trace-based estimator g_t with the empty-trace estimator g_e as

$$g_c(\theta; x, (\tau, y)^\pm, y_e^\pm) = \alpha g_t(\theta; x, (\tau, y)^\pm) + (1-\alpha) g_e(\theta; x, y_e^\pm),$$

where $\alpha \in [0, 1]$. In practice, these per-example gradients are averaged to form the parameter update.

4 Theoretical analysis

We analyze BVPO under trace-induced stochasticity. Our main variance result concerns the *conditional* reasoning-trace-noise component under a fixed prompt-and-answer conditional view, rather than the full training variance.

Analysis setup. Unless otherwise stated, we condition on a fixed prompt x and outcome set $\mathcal{O} = \{y^\pm, y_e^\pm\}$. Here, $\tau = (\tau^+, \tau^-)$ denotes the retained trace pair for the preferred and dispreferred sampled trajectories in the trace-based branch. Under this conditioning, g_e is deterministic with respect to trace randomness, while g_t remains stochastic through the distribution of τ given $(x, \mathcal{O})$, thereby isolating the trace-noise component studied below.

4.1 Contraction of trace-induced conditional variance

Let $\mathbb{E}_\tau[\cdot \mid x, \mathcal{O}]$ denote expectation with respect to the remaining trace randomness under the fixed-$(x, \mathcal{O})$ conditional formulation. Define the scalar conditional variance of an estimator g by

$$\mathrm{Var}_\tau(g \mid x, \mathcal{O}) := \mathbb{E}_\tau[\|g - \mathbb{E}_\tau[g \mid x, \mathcal{O}]\|^2 \mid x, \mathcal{O}].$$

Since g_e bypasses trace generation, it satisfies $\mathrm{Var}_\tau(g_e \mid x, \mathcal{O}) = 0$. We then obtain the following contraction result for the composite estimator.

Proposition 4.1 (Trace-induced variance contraction)**.** *For any mixing coefficient $\alpha \in [0, 1]$, the trace-induced variance of the BVPO estimator g_c satisfies*

$$\mathrm{Var}_\tau(g_c \mid x, \mathcal{O}) = \alpha^2 \, \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) \leq \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) \, .$$

This inequality is strict for $\alpha \in [0, 1)$ whenever $\mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) > 0$.

The proof is provided in Appendix A.1. Proposition 4.1 shows that, under the fixed-$(x, \mathcal{O})$ conditional view, the empty-trace branch contracts the reasoning-trace-noise component by a factor of α^2. This is a conditional, estimator-side statement and should be interpreted jointly with the bias introduced by the empty-trace branch. The next subsection makes this trade-off explicit via conditional MSE minimization.

4.2 Optimal combination via conditional MSE minimization

We now study the conditional mean-squared error (MSE) of the gradient estimator with respect to the target marginal gradient $\mu := \nabla_\theta \ell_m(\theta; x, y^\pm)$. As in Section 4.1, we condition on a fixed prompt x and outcome set $\mathcal{O} = \{y^\pm, y_e^\pm\}$, so that g_t remains stochastic only through the residual trace randomness in τ given $(x, \mathcal{O})$. Under this conditioning, g_e is deterministic with respect to the trace randomness, but remains biased relative to μ due to the separately sampled empty-trace pair $y_e^\pm$.

Definitions. For $i \in \{t, e\}$, define the conditional bias $b_i := \mathbb{E}_\tau[g_i \mid x, \mathcal{O}] - \mu$. Since g_e is deterministic under fixed $(x, \mathcal{O})$, we have $b_e = g_e - \mu$. For any gradient estimator g, define its conditional MSE as

$$\mathrm{MSE}(g) := \mathbb{E}_\tau \left[\|g - \mu\|^2 \mid x, \mathcal{O} \right] .$$

Objective formulation. Let $\Delta := g_t - g_e$. As shown in Appendix A.2, the error vector of the combined estimator $g_c(\alpha)$ can be written as

$$g_c(\alpha) - \mu = b_e + \alpha \Delta.$$

Taking the expected squared norm yields the conditional MSE as a quadratic function of the mixing weight α:

$$\mathrm{MSE}(\alpha) = \mathbb{E}_\tau \left[\|g_t - g_e\|^2 \mid x, \mathcal{O} \right] \alpha^2 + 2\langle b_e, \, b_t - b_e \rangle \alpha + \|b_e\|^2. \tag{5}$$

Minimizing this quadratic yields the following result.

Theorem 4.2 (Conditional MSE-optimal convex combination)**.** *Assume the estimators are distinct, i.e.,*

$$\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}] > 0.$$

For $\alpha \in [0, 1]$, the conditional MSE of the convex combination $g_c(\alpha) := \alpha g_t + (1 - \alpha) g_e$ is minimized by $\alpha^\star := \Pi_{[0,1]}(\alpha_{\mathrm{unc}})$, where α_{unc} is given by:

$$\alpha_{\mathrm{unc}} = \frac{\|b_e\|^2 - \langle b_t, b_e \rangle}{\mathbb{E}_\tau \left[\|g_t - g_e\|^2 \;\middle|\; x, \mathcal{O} \right]}. \tag{6}$$

Here, $\Pi_{[0,1]}$ denotes projection onto the interval $[0, 1]$.

Theorem 4.2 characterizes, under the fixed-$(x, \mathcal{O})$ conditional view, the conditionally optimal MSE trade-off between the stochastic trace-based branch and the biased empty-trace branch. When the unconstrained minimizer lies in the interior, the resulting mixture achieves lower conditional MSE than either endpoint estimator; see Appendix A.2.

4.3 Estimator-dependent SGD bound

We now connect the instance-wise analysis in Section 4.2 to a standard estimator-dependent SGD bound for the population objective (3). Our analysis follows the standard framework for non-convex optimization with biased gradient estimators [2, 9], and makes explicit how the bias and variance of the composite gradient estimator appear in the optimization guarantee.

Global randomness and variance decomposition. At iteration k, let $\mathbb{E}_k[\cdot]$ denote the full expectation conditioned on θ_k, over all randomness in the estimator. By the law of total variance,

$$\mathrm{Var}_k(g_c) = \underbrace{\mathrm{Var}_{x,\mathcal{O}}(\mathbb{E}_\tau[g_c \mid x, \mathcal{O}])}_{\text{(I) Outer Data / Outcome Randomness}} + \underbrace{\mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})]}_{\text{(II) Reasoning Trace Noise}}. \tag{7}$$

The proof of Equation (7) is deferred to Appendix A.3. Equation (7) decomposes the full variance into an outer data/outcome term and a conditional reasoning-trace-noise term.

Proposition 4.1 contracts the latter term after averaging over the outer randomness. Appendix A.4 gives a sufficient condition under which this contraction extends to the full variance.

Global bias and variance. We define the global bias and variance of g_c at iteration k by

$$\mathrm{Bias}_k(g_c) := \mathbb{E}_k[g_c(\theta_k)] - \nabla_\theta \mathcal{L}(\theta_k), \qquad \mathrm{Var}_k(g_c) := \mathbb{E}_k\Big[\|g_c(\theta_k) - \mathbb{E}_k[g_c(\theta_k)]\|^2\Big].$$

Since θ_k is itself random along the SGD trajectory, we use an outer expectation $\mathbb{E}[\cdot]$ in the convergence bound below to account for the resulting randomness of the iterates.

Theorem 4.3 (SGD convergence with biased estimators)**.** *Assume $\mathcal{L}(\theta)$ is L-smooth and bounded below by $\mathcal{L}^*$. Let the learning rate satisfy $\eta \le 1/L$. Then the average squared gradient norm over K iterations is bounded by:*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1} \|\nabla_\theta \mathcal{L}(\theta_k)\|^2\right] \le \frac{2\Delta\mathcal{L}}{K\eta} + \frac{1}{K}\sum_{k=0}^{K-1} \underbrace{\mathbb{E}\big[\|\mathrm{Bias}_k(g_c)\|^2 + \eta L\,\mathrm{Var}_k(g_c)\big]}_{:=\mathcal{E}_k}, \tag{8}$$

where $\Delta\mathcal{L} := \mathcal{L}(\theta_0) - \mathcal{L}^*$ is the initial optimality gap.

Appendix A.5 includes the complete proof. Theorem 4.3 is a standard biased-SGD bound. It shows that the choice of gradient estimator affects training through the estimator-dependent quantity

$$\mathcal{E}_k = \|\mathrm{Bias}_k(g_c)\|^2 + \eta L\,\mathrm{Var}_k(g_c),$$

which appears in the global bound through the outer expectation and the averaging over k. In particular, while Theorem 4.2 gives an instance-conditioned optimum under fixed $(x, \mathcal{O})$, Theorem 4.3 identifies the corresponding optimization-level quantity in the global bound. Appendix A.6 studies a global constant α through complementary fixed-reference-parameter analysis and existence results, while its training-time choice remains empirical.

5 Experiments

We empirically evaluate BVPO from three complementary perspectives: alignment quality on instruction following benchmarks, preservation of reasoning performance on math reasoning benchmarks, and the controlled empirical diagnostics for the proposed estimator interpretation.

5.1 Experiment settings

Models and training settings. We conduct experiments on four LRMs: DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-0528-Qwen3-8B [6], and MiMo-7B-RL [33]. The first three models are trained with chain-of-thought reasoning data using SFT from Qwen 2.5 [35] and Qwen 3 [34] base models. MiMo-7B-RL adopts a Multi-Token Prediction architecture and has been trained with RLVR for reasoning ability. All experimented LRMs studied here are reasoning models that have not been trained with RLHF for preference alignment.

Following [22], we use prompts from the UltraFeedback dataset [5] and let each model generate 5 responses with a temperature of 0.8. These responses are then ranked using ArmoRM [32]. Following [6], the score is computed using only the final answer part of each response. The highest and lowest-ranked responses are selected as the preferred and dispreferred samples, respectively. Training pairs are constructed offline. We instantiate BVPO with the composite gradient estimator g_c under the DPO objective, and compare it against the original base models and two state-of-the-art preference optimization methods: DPO [26] and SimPO [22].

Table 1: Alignment results on Arena-Hard [18] and AlpacaEval 2 [19] in *Thinking* and *NoThinking* modes. LC Win Rate denotes length-controlled win rate.

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
R1-Qwen-7B	16.3	15.7	18.4	16.7	15.2	17.0
SimPO	19.0	17.8	20.2	17.2	19.1	20.3
DPO	19.1	18.3	20.4	17.7	19.3	20.7
BVPO	**22.5**	**21.8**	**24.7**	**22.7**	**22.6**	**22.9**

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
R1-Qwen-1.5B	4.4	5.4	6.3	5.5	6.9	6.9
SimPO	5.5	6.2	8.4	4.5	4.6	3.8
DPO	5.1	6.4	8.0	7.2	7.8	7.1
BVPO	**6.5**	**6.9**	**8.7**	**7.6**	**10.2**	**10.7**

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
R1-0528-Qwen3-8B	65.4	48.7	39.6	65.2	37.5	31.8
SimPO	69.2	49.1	44.9	62.1	41.2	41.5
DPO	68.7	48.9	44.3	61.6	40.3	40.0
BVPO	**70.9**	**50.1**	**45.3**	**66.4**	**45.7**	**54.2**

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
MiMo-7B-RL	60.8	49.3	50.4	44.9	34.4	33.2
SimPO	62.0	46.4	47.2	41.3	35.9	35.2
DPO	62.9	48.8	50.1	42.1	35.7	34.4
BVPO	**65.7**	**52.6**	**52.0**	**53.8**	**49.2**	**43.8**

Evaluation benchmarks. Following [22], we evaluate alignment performance on two widely used open-ended instruction-following benchmarks, Arena-Hard [18] and AlpacaEval 2 [19], which measure response quality across diverse prompts. For Arena-Hard, we report the win rate against GPT-4-0314. For AlpacaEval 2, we report both the win rate and length-controlled win rate against GPT-4 Turbo. We assess LRMs in two modes: the standard reasoning mode, denoted *Thinking*; and suppressing reasoning trace generation by appending "`<think></think>`" to the input prompt, denoted *NoThinking*, reflecting scenarios in which users prefer instant responses without reasoning.

To examine whether reasoning capabilities are preserved after alignment, we further evaluate BVPO on six widely used math-reasoning benchmarks: AIME 2024, AIME 2025, AMC [17], Minerva [16], OlympiadBench [10], and MATH-500 [12]. We report avg@32 accuracy for AIME 2024, AIME 2025, and AMC due to their relatively small test sets, and pass@1 on the remaining benchmarks. All evaluations use a temperature of 0.6. Further experiment details are in Appendix B.

5.2 Main results

Alignment quality. Table 1 reports alignment results on AlpacaEval 2 [19] and Arena-Hard [18]. Across both benchmarks, BVPO consistently surpasses the best baselines. In *Thinking* mode, BVPO improves AlpacaEval 2 win rate by up to 3.5 points and the length-controlled win rate by up to 4.3 points, and increases the Arena-Hard win rate by up to 3.4 points. In *NoThinking* mode, BVPO yields gains of up to 8.9 points on Arena-Hard and up to 13.3 win rate and 12.7 length-controlled win rate points on AlpacaEval 2. These results demonstrate the effectiveness of the proposed composite gradient estimator in the evaluated settings. Moreover, these results confirm that BVPO effectively aligns models following both the SFT stage (DeepSeek-R1-Distill-Qwens) and the RLVR stage (MiMo-7B-RL), solidifying its utility across models with diverse training paradigms.

Table 2: Reasoning performance after alignment on six math benchmarks: AIME 2024, AIME 2025, AMC [17], MATH-500 [12], Minerva [16], and OlympiadBench [10].

Method	AIME 24	AIME 25	AMC	MATH-500	Minerva	OlympiadBench	Avg.
R1-Qwen-7B	56.3	40.3	79.7	89.2	40.1	57.5	60.5
SimPO	55.8	38.2	80.7	88.2	41.2	58.4	60.4
DPO	55.0	40.7	80.8	89.8	40.8	59.0	61.0
BVPO	56.0	39.4	80.9	90.2	43.0	58.7	**61.4**
Method	**AIME 24**	**AIME 25**	**AMC**	**MATH-500**	**Minerva**	**OlympiadBench**	**Avg.**
R1-Qwen-1.5B	28.6	21.7	62.2	81.8	29.0	44.9	44.7
SimPO	30.2	23.3	62.5	82.6	30.5	46.2	45.9
DPO	31.7	23.4	64.9	84.0	33.8	48.7	47.8
BVPO	33.4	24.4	65.6	85.2	33.8	48.6	**48.5**
Method	**AIME 24**	**AIME 25**	**AMC**	**MATH-500**	**Minerva**	**OlympiadBench**	**Avg.**
R1-0528-Qwen3-8B	73.1	66.0	91.8	96.4	47.1	73.5	74.7
SimPO	73.9	66.1	91.0	96.4	47.5	76.0	75.2
DPO	73.6	65.9	91.0	97.6	47.1	76.0	75.2
BVPO	74.9	67.2	91.7	96.0	48.2	75.6	**75.6**
Method	**AIME 24**	**AIME 25**	**AMC**	**MATH-500**	**Minerva**	**OlympiadBench**	**Avg.**
MiMo-7B-RL	64.3	56.9	87.1	95.0	46.7	72.7	70.5
SimPO	66.5	57.1	86.9	94.4	45.6	72.4	70.5
DPO	64.8	55.9	88.7	95.8	47.8	72.9	71.0
BVPO	66.5	55.6	88.5	95.2	48.9	72.9	**71.3**

Table 3: Comparison with data-mixing and Bernoulli controls.

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
R1-Qwen-1.5B	4.4	5.4	6.3	5.5	6.9	6.9
DPO	5.1	6.4	8.0	7.2	7.8	7.1
DPO w/ data mixing	5.0	5.6	7.1	7.2	9.3	9.6
Bernoulli	5.4	6.6	8.4	6.4	9.0	9.0
BVPO	**6.5**	**6.9**	**8.7**	**7.6**	**10.2**	**10.7**

Reasoning performance after alignment. Since preference alignment is typically the final tuning stage before deployment, it is crucial that preference optimization not erode LRMs' reasoning ability acquired from earlier training stages. Table 2 reports results on six math reasoning benchmarks. Since the training preference data consist of general conversational rather than math-specialized supervision, these benchmarks are used primarily to assess whether the alignment gains in Table 1 come at the cost of prior reasoning ability after preference optimization. Across the tested model families, preference optimization does not uniformly improve or degrade every benchmark, but BVPO remains competitive overall and does not exhibit a consistent drop across the six math benchmarks.

5.3 Analysis

Comparison with data-mixing and Bernoulli controls. A natural question is whether the empirical gain of BVPO is explained simply by adding empty-trace data, or instead reflects the estimator-level role of the empty-trace branch. To probe this, Table 3 compares BVPO against two controls on R1-Qwen-1.5B. The first is naive data mixing, which uses the same trace-based and empty-trace preference pairs as separate DPO training instances rather than combining them within a single per-prompt estimator. The second is a matched-expectation Bernoulli baseline, which replaces BVPO's deterministic within-step averaging with stochastic branch selection, whose expected per-example gradient matches BVPO's estimator; its empirical setup is described in Appendix C.1, and its formal construction and theoretical comparison are given in Appendix C.2.1. BVPO achieves higher scores than both controls on the alignment metrics. Together, these comparisons support the interpretation

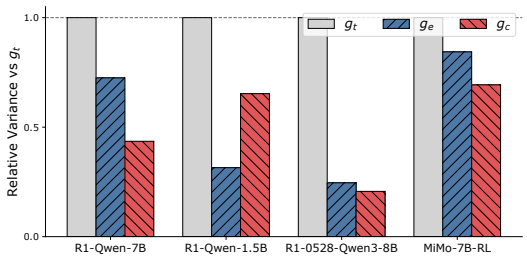

Figure 1: Observable empirical total gradient variance in the final-layer Q/K/V subspace, normalized by g_t.

Table 4: Training-time variance diagnostics on R1-Qwen-1.5B. Columns indicate training progress in epochs from 0 to 1.0. "BVPO vs. DPO" and "BVPO vs. Bern." report the relative reduction in observable empirical total gradient variance achieved by BVPO, while "Trace ratio" reports the trace-associated variance ratio of g_t.

Training epoch	**0**	**0.25**	**0.5**	**0.75**	**1.0**
BVPO vs. DPO	34.7%	33.9%	36.4%	40.6%	37.2%
BVPO vs. Bern.	24.7%	25.9%	26.7%	27.1%	26.7%
Trace ratio	81.5%	81.0%	74.6%	79.1%	77.8%

that BVPO's gain is consistent with arising from combining the trace-based and empty-trace branches within a single prompt-paired estimator, rather than solely from adding empty-trace supervision or from matching BVPO in expectation alone.

Gradient variance analysis. Proposition 4.1 concerns the *conditional* reasoning-trace-noise component under fixed $(x, \mathcal{O})$, which is not directly measurable in practical training. We therefore do not measure that quantity. Instead, we report *observable empirical total gradient variance* in the final-layer Q/K/V projection subspace; Appendix C.2.3 gives the definitions and measurement protocol. We use this as a tractable proxy for variance differences across estimators. Fig. 1 shows that the composite estimator g_c exhibits lower observable total gradient variance than g_t across the tested models, while Table 4 reports complementary training-time diagnostics on R1-Qwen-1.5B.

Table 4 supports three observations. First, across checkpoints spanning training, BVPO reduces observable empirical total gradient variance by 33.9%–40.6% relative to DPO over training. Second, BVPO reduces this variance by 24.7%–27.1% relative to the matched-expectation Bernoulli baseline. Third, the trace-associated variance ratio of g_t (measured via a token-partitioning heuristic as a practical proxy; see Appendix C.2.3) remains 74.6%–81.5% over training, indicating that trace-related variability is a substantial contributor to the observable empirical total variance of the trace-based gradient in this diagnostic subspace. These results provide observable evidence in the predicted direction and are consistent with lower variance under deterministic within-step averaging than under stochastic branch selection. Note that, our empirical variance results should be interpreted as diagnostic evidence consistent with the theory, not as a direct empirical verification of Proposition 4.1.

Additional comparisons and robustness. We provide several additional comparisons and robustness checks. On R1-Qwen-1.5B, BVPO outperforms both $K = 2$ and $K = 4$ multi-trace averaging DPO and DPO with token weighting on the reported alignment metrics. We further evaluate BVPO using a different reward model for preference labeling, Skywork-Reward-V2-Qwen3-8B [20]; under this labeling setup, BVPO again achieves higher scores than DPO. Sensitivity studies over the mixing coefficient α show only modest variation around the main setting, suggesting that the method is not overly sensitive to this hyperparameter. We also conduct a 14B-scale experiment on DeepSeek-R1-Distill-Qwen-14B. In this setting, BVPO achieves higher scores than both DPO and DPO with data mixing on the reported alignment metrics in both *Thinking* and *NoThinking* modes. For example, on Arena-Hard, BVPO scores 63.3 compared with 61.0/61.3 for DPO/DPO with data mixing in *Thinking* mode, and 52.2 compared with 45.3/49.3 in *NoThinking* mode. Full results and implementation details are provided in Appendices C.1.2–C.1.5.

6 Conclusion

We study preference optimization for large reasoning models with an intractable trace-marginalized objective and propose Bias-Variance Optimized Preference Optimization (BVPO), a prompt-paired composite estimator that combines trace-based and empty-trace gradients. Under a fixed prompt-and-answer conditional view, the empty-trace branch is deterministic with respect to the trace randomness isolated in our analysis. Our analysis shows that BVPO contracts the conditional reasoning-trace-noise component, admits a conditional MSE-optimal mixture, and appears in the estimator-dependent term of a standard biased-SGD bound. Empirically, BVPO improves alignment quality over the compared baselines while generally preserving reasoning performance on the reported benchmarks. Empirical diagnostics also show that BVPO yields lower total gradient variance during training.

Acknowledgements

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R, General Research Fund (No. 17213925) and National Natural Science Foundation of China (No. 62422606, 62441615).

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, August 2024.

[2] Ahmad Ajalloeian and Sebastian U. Stich. On the convergence of SGD with biased gradients, 2020. URL `https://arxiv.org/abs/2008.00051`.

[3] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324 – 345, 1952.

[4] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, and et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL `https://arxiv.org/abs/2507.06261`.

[5] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9722 – 9744, 2024.

[6] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

[7] 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*, volume 235, pages 12634–12651, 21–27 Jul 2024.

[8] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, January 1992.

[9] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[10] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

[11] Haoran He, Yuxiao Ye, Qingpeng Cai, Chen Hu, Binxing Jiao, Daxin Jiang, and Ling Pan. Random policy valuation is enough for LLM reasoning with verifiable rewards. In *The Fourteenth International Conference on Learning Representations*, 2026.

[12] 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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[13] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[14] OpenAI: Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick

Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. OpenAI o1 system card, 2024. URL `https://arxiv.org/abs/2412.16720`.

[15] Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Overthinking reduction with decoupled rewards and curriculum data scheduling. In *The Fourteenth International Conference on Learning Representations*, 2026.

[16] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[17] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in AI4Maths with 860k pairs of competition math problems and solutions, 2024. URL `https://github.com/project-numina/aimo-progress-prize`.

[18] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2025.

[19] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models, 2023. URL `https://github.com/tatsu-lab/alpaca_eval`.

[20] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and Yang Liu. Skywork-reward-v2: Scaling preference data curation via human-AI synergy. In *The Fourteenth International Conference on Learning Representations*, 2026.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[22] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[23] Yibo Miao, Bofei Gao, Shanghaoran Quan, Junyang Lin, Daoguang Zan, Jiaheng Liu, Jian Yang, Tianyu Liu, and Zhijie Deng. Aligning codeLLMs with direct preference optimization, 2024. URL `https://arxiv.org/abs/2410.18585`.

[24] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

[25] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL*, pages 4998–5017, 2024.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

[27] Paria Rashidinejad and Yuandong Tian. Sail into the headwind: Alignment via robust rewards and dynamic labels against reward hacking. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

[29] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL `https://arxiv.org/abs/2402.03300`.

[30] Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of EMNLP*, 2024.

[33] LLM-Core Xiaomi: Bingquan Xia, Bowen Shen, Cici, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, Liang Zhao, Peidian Li, Peng Wang, Shihua Yu, Shimao Chen, Weikun Wang, Wenhan Ma, Xiangwei Deng, Yi Huang, Yifan Song, Zihan Jiang, Bowen Ye, Can Cai, Chenhong He, Dong Zhang, Duo Zhang, Guoan Wang, Hao Tian, Haochen Zhao, Heng Qu, Hongshen Xu, Jun Shi, Kainan Bao, Kai Fang, Kang Zhou, Kangyang Zhou, Lei Li, Menghang Zhu, Nuo Chen, Qiantong Wang, Shaohui Liu, Shicheng Li, Shuhao Gu, Shuhuai Ren, Shuo Liu, Sirui Deng, Weiji Zhuang, Weiwei Lv, Wenyu Yang, Xin Zhang, Xing Yong, Xing Zhang, Xingchen Song, Xinzhe Xu, Xu Wang, Yihan Yan, Yu Tu, Yuanyuan Tian, Yudong Wang, Yue Yu, Zhenru Lin, Zhichao Song, and Zihao Yue. MiMo: unlocking the reasoning potential of language model – from pretraining to posttraining, 2025. URL `https://arxiv.org/abs/2505.07608`.

[34] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[35] Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

[36] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. SimpleRL-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025.

[37] Mingkang Zhu, Xi Chen, Zhongdao Wang, Bei Yu, Hengshuang Zhao, and Jiaya Jia. TGDPO: Harnessing token-level reward guidance for enhancing direct preference optimization. In *Forty-second International Conference on Machine Learning*, 2025.

[38] Mingkang Zhu, Xi Chen, Senqiao Yang, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Learning active perception for pixel-space reasoning via visual-intent stratified grpo. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari, editors, *Computer Vision – ECCV 2026*, pages 286–303, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-36969-7.

[39] Mingkang Zhu, Xi Chen, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Stratified GRPO: Handling structural heterogeneity in reinforcement learning of LLM search agents. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=ZX2lOgKOTm`.

[40] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL `https://arxiv.org/abs/1909.08593`.

A Proofs of theoretical results

A.1 Proof of Proposition 4.1

For completeness, we first restate the theorem with the relevant definitions and setup.

Proposition A.1 (Trace-induced variance contraction)**.** *Fix a prompt x and outcome set $\mathcal{O} = \{y^{\pm}, y_e^{\pm}\}$. Here, $\tau = (\tau^+, \tau^-)$ denotes the retained trace pair for the preferred and dispreferred sampled trajectories in the trace-based branch. Let g_t denote the trace-based gradient estimator, and let g_e denote the empty-trace estimator. All expectations $\mathbb{E}_\tau[\cdot \mid x, \mathcal{O}]$ are taken with respect to the remaining trace randomness under the fixed-$(x, \mathcal{O})$ conditional formulation. Since g_e bypasses the reasoning step, it is deterministic with respect to this trace randomness.*

Define the scalar trace-induced conditional variance of an estimator g as

$$\mathrm{Var}_\tau(g \mid x, \mathcal{O}) := \mathbb{E}_\tau\left[\|g - \mathbb{E}_\tau[g \mid x, \mathcal{O}]\|^2 \;\middle|\; x, \mathcal{O}\right].$$

For any mixing coefficient $\alpha \in [0, 1]$, the combined estimator $g_c := \alpha g_t + (1-\alpha) g_e$ satisfies the following contraction property:

$$\mathrm{Var}_\tau(g_c \mid x, \mathcal{O}) = \alpha^2 \, \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) \;\le\; \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}).$$

The inequality is strict whenever $\alpha \in [0, 1)$ and $\mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) > 0$.

Consequently, averaging this pointwise inequality over the outer randomness of $(x, \mathcal{O})$ yields

$$\mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})] \le \mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_t \mid x, \mathcal{O})].$$

Clarification. The quantity $\mathrm{Var}_\tau(\cdot \mid x, \mathcal{O})$ isolates the variance induced solely by the remaining reasoning-trace randomness under the fixed-$(x, \mathcal{O})$ conditional formulation. All other randomness is held fixed by the conditioning on $(x, \mathcal{O})$.

Proof. Fix $(x, \mathcal{O})$. We compute the scalar conditional variance with respect to the remaining trace randomness under this conditioning, denoted by $\mathrm{Var}_\tau(\cdot \mid x, \mathcal{O})$. Since

$$g_c = \alpha g_t + (1-\alpha) g_e,$$

and g_e is deterministic with respect to this trace randomness, we have

$$\mathbb{E}_\tau[g_e \mid x, \mathcal{O}] = g_e.$$

Hence

$$\mathbb{E}_\tau[g_c \mid x, \mathcal{O}] = \alpha \, \mathbb{E}_\tau[g_t \mid x, \mathcal{O}] + (1-\alpha) g_e,$$

so

$$g_c - \mathbb{E}_\tau[g_c \mid x, \mathcal{O}] = \alpha\big(g_t - \mathbb{E}_\tau[g_t \mid x, \mathcal{O}]\big).$$

Therefore,

$$\begin{aligned}\mathrm{Var}_\tau(g_c \mid x, \mathcal{O}) &= \mathbb{E}_\tau\Big[\|g_c - \mathbb{E}_\tau[g_c \mid x, \mathcal{O}]\|^2 \mid x, \mathcal{O}\Big] \\ &= \alpha^2 \,\mathrm{Var}_\tau(g_t \mid x, \mathcal{O}).\end{aligned}$$

Since $\alpha \in [0, 1]$, we have $\alpha^2 \leq 1$, and therefore

$$\mathrm{Var}_\tau(g_c \mid x, \mathcal{O}) \leq \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}).$$

The inequality is strict whenever $\alpha \in [0, 1)$ and $\mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) > 0$.

Finally, since this pointwise inequality holds for every fixed $(x, \mathcal{O})$, taking expectation over the outer randomness preserves it:

$$\mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})] \leq \mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_t \mid x, \mathcal{O})]\,. \qquad \square$$

A.2 Proofs of conditional MSE-optimality

In this appendix, we provide a detailed derivation of the conditional MSE-optimal combination weight. We restate the theorem for completeness.

Analysis setup. Consistent with Section 4.2, we condition on a fixed prompt x and outcome set $\mathcal{O} = \{y^\pm, y_e^\pm\}$. Here, $\tau = (\tau^+, \tau^-)$ denotes the retained trace pair for the preferred and dispreferred sampled trajectories in the trace-based branch. Consequently, g_e is deterministic under this conditioning. For notational simplicity, $\mathbb{E}_\tau[\cdot]$ denotes conditional expectation over the remaining trace randomness given fixed $(x, \mathcal{O})$.

Theorem A.2 (Restatement of Theorem 4.2)**.** *Fix a prompt x and outcome set $\mathcal{O} = \{y^\pm, y_e^\pm\}$. Let g_t and g_e be two estimators of the target marginal gradient*

$$\mu := \nabla_\theta \ell_m(\theta; x, y^\pm).$$

All expectations $\mathbb{E}_\tau[\cdot]$ are taken with respect to the remaining trace randomness under the fixed-$(x, \mathcal{O})$ conditional formulation.

Define the conditional bias vectors by

$$b_i := \mathbb{E}_\tau[g_i \mid x, \mathcal{O}] - \mu, \quad \textit{for } i \in \{t, e\}.$$

Since g_e is deterministic under this conditioning, we have

$$b_e = g_e - \mu, \qquad \mathrm{Var}_\tau(g_e \mid x, \mathcal{O}) = 0.$$

For $\alpha \in [0, 1]$, define the convex combination $g_c(\alpha) := \alpha g_t + (1 - \alpha) g_e$, and its conditional mean squared error:

$$\mathrm{MSE}(g_c(\alpha)) := \mathbb{E}_\tau\left[\|g_c(\alpha) - \mu\|^2 \mid x, \mathcal{O}\right].$$

Assuming the estimators are distinct (i.e., $\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}] > 0$), the unconstrained minimizer over $\alpha \in \mathbb{R}$ is given by:

$$\alpha_{\mathrm{unc}} = \frac{\|b_e\|^2 - \langle b_t, b_e \rangle}{\mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right]}. \tag{9}$$

Accordingly, the conditional MSE-optimal weight is

$$\alpha^\star := \Pi_{[0,1]}(\alpha_{\mathrm{unc}}).$$

Proof. Let $\Delta := g_t - g_e$ denote the difference vector between the trace-based and empty-trace estimators. We can rewrite the combined estimator as:

$$g_c(\alpha) = \alpha g_t + (1 - \alpha) g_e = g_e + \alpha(g_t - g_e) = g_e + \alpha\Delta.$$

The error vector with respect to the target μ is:

$$g_c(\alpha) - \mu = (g_e - \mu) + \alpha\Delta = b_e + \alpha\Delta.$$

Since g_e is deterministic under the fixed-$(x, \mathcal{O})$ conditional formulation, $b_e = g_e - \mu$ is also deterministic under this conditioning. The MSE is the expected squared norm of this error vector:

$$\begin{aligned}\mathrm{MSE}(\alpha) &= \mathbb{E}_\tau\left[\|b_e + \alpha\Delta\|^2 \;\middle|\; x, \mathcal{O}\right] \\ &= \mathbb{E}_\tau\left[\|b_e\|^2 + 2\alpha\langle b_e, \Delta\rangle + \alpha^2\|\Delta\|^2 \;\middle|\; x, \mathcal{O}\right] \\ &= \|b_e\|^2 + 2\alpha\langle b_e, \mathbb{E}_\tau[\Delta \mid x, \mathcal{O}]\rangle + \alpha^2\mathbb{E}_\tau[\|\Delta\|^2 \mid x, \mathcal{O}].\end{aligned}$$

To find the unconstrained minimum, we take the derivative with respect to α and set it to zero:

$$\frac{d}{d\alpha}\,\mathrm{MSE}(\alpha) = 2\langle b_e, \mathbb{E}_\tau[\Delta \mid x, \mathcal{O}]\rangle + 2\alpha\mathbb{E}_\tau[\|\Delta\|^2 \mid x, \mathcal{O}] = 0.$$

Solving for α, we obtain:

$$\alpha_{\mathrm{unc}} = -\frac{\langle b_e, \mathbb{E}_\tau[\Delta \mid x, \mathcal{O}]\rangle}{\mathbb{E}_\tau[\|\Delta\|^2 \mid x, \mathcal{O}]}.$$

Now we simplify the numerator term $\langle b_e, \mathbb{E}_\tau[\Delta \mid x, \mathcal{O}]\rangle$. Recalling that $\Delta = g_t - g_e$ and $\mathbb{E}_\tau[g_t \mid x, \mathcal{O}] = \mu + b_t$, we have:

$$\mathbb{E}_\tau[\Delta \mid x, \mathcal{O}] = \mathbb{E}_\tau[g_t \mid x, \mathcal{O}] - g_e = (\mu + b_t) - (\mu + b_e) = b_t - b_e.$$

Substituting this back into the inner product:

$$\begin{aligned}-\langle b_e, \mathbb{E}_\tau[\Delta \mid x, \mathcal{O}]\rangle &= -\langle b_e, b_t - b_e\rangle \\ &= -\left(\langle b_e, b_t\rangle - \langle b_e, b_e\rangle\right) \\ &= \|b_e\|^2 - \langle b_t, b_e\rangle.\end{aligned}$$

The denominator is simply $\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}]$. Combining these results yields the optimal unconstrained weight:

$$\alpha_{\mathrm{unc}} = \frac{\|b_e\|^2 - \langle b_t, b_e\rangle}{\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}]}.$$

Since the coefficient of α^2 is $\mathbb{E}_\tau[\|\Delta\|^2 \mid x, \mathcal{O}] > 0$, the objective is a strictly convex quadratic in α. Hence α_{unc} is the unique unconstrained minimizer, and the constrained minimizer over $[0, 1]$ is obtained by projection:

$$\alpha^\star = \Pi_{[0,1]}(\alpha_{\mathrm{unc}}).$$ □

Corollary A.3. *Under the assumptions of Theorem 4.2, the conditional MSE-optimal estimator $g_c(\alpha^\star)$ satisfies*

$$\mathrm{MSE}(g_c(\alpha^\star)) \le \min\{\mathrm{MSE}(g_t), \mathrm{MSE}(g_e)\}.$$

Moreover, if

$$\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}] > 0 \quad \textit{and} \quad \alpha_{\mathrm{unc}} \in (0, 1),$$

then the inequality is strict:

$$\mathrm{MSE}(g_c(\alpha^\star)) < \min\{\mathrm{MSE}(g_t),\ \mathrm{MSE}(g_e)\}.$$

Proof. Let $J(\alpha) := \mathrm{MSE}(g_c(\alpha))$ denote the conditional MSE objective. As established in Theorem 4.2, $J(\alpha)$ is a quadratic function with leading coefficient $A = \mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}]$. The feasible interval includes the endpoints of the interval $[0, 1]$:

- At $\alpha = 1$, $g_c(1) = g_t$, so $J(1) = \mathrm{MSE}(g_t)$.
- At $\alpha = 0$, $g_c(0) = g_e$, so $J(0) = \mathrm{MSE}(g_e)$.

Since $\alpha^\star$ is defined as the minimizer of $J(\alpha)$ over the domain $[0,1]$, it must satisfy:

$$\mathrm{MSE}(g_c(\alpha^\star)) = J(\alpha^\star) \le \min\{J(0), J(1)\} = \min\{\mathrm{MSE}(g_e), \mathrm{MSE}(g_t)\}.$$

To prove the strict inequality, we proceed with the following steps:

1. Strict convexity. The condition $\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}] > 0$ implies $A > 0$. Thus, $J(\alpha)$ is a strictly convex parabola. A fundamental property of strictly convex functions is that their global minimizer α_{unc} over $\mathbb{R}$ is unique, satisfying:

$$J(\alpha_{\text{unc}}) < J(\alpha) \quad \forall \alpha \neq \alpha_{\text{unc}}.$$

2. Interior solution. The condition $\alpha_{\text{unc}} \in (0,1)$ implies that the unconstrained minimizer lies strictly within the feasible set. Therefore, the constrained optimal weight is exactly the unconstrained minimizer:

$$\alpha^\star = \Pi_{[0,1]}(\alpha_{\text{unc}}) = \alpha_{\text{unc}}.$$

3. Strict dominance. Since $\alpha^\star \in (0,1)$, it follows that $\alpha^\star \neq 0$ and $\alpha^\star \neq 1$. Applying the uniqueness property from Step 1 to the points $\alpha = 0$ and $\alpha = 1$:

$$\alpha^\star \neq 0 \implies J(\alpha^\star) < J(0),$$
$$\alpha^\star \neq 1 \implies J(\alpha^\star) < J(1).$$

Consequently, $J(\alpha^\star)$ is strictly less than the minimum of the endpoint values:

$$\mathrm{MSE}(g_c(\alpha^\star)) < \min\{\mathrm{MSE}(g_e), \mathrm{MSE}(g_t)\}.$$

This concludes the proof. □

A.3 Detailed derivation of global variance decomposition (Equation (7))

In this appendix, we derive the variance decomposition used in Section 4.3. Throughout this proof, all expectations are conditioned on a fixed iterate θ_k, and we suppress the iteration index k for readability.

Notation. Let $Z := (x, \mathcal{O})$ denote the outer randomness, where x is the prompt and $\mathcal{O}$ is the outcome set. Let τ denote the remaining reasoning-trace randomness. Define

$$\bar{g}(Z) := \mathbb{E}_\tau[g_c \mid Z].$$

By construction, $\bar{g}(Z)$ is the conditional mean of g_c given Z.

Proof. By definition, the variance at the current iterate is

$$\mathrm{Var}(g_c) = \mathbb{E}\left[\|g_c - \mu'\|^2\right], \quad \text{where } \mu' = \mathbb{E}[g_c].$$

We employ the standard technique of conditioning on Z. We add and subtract the conditional mean $\bar{g}(Z)$:

$$g_c - \mu' = (g_c - \bar{g}(Z)) + (\bar{g}(Z) - \mu').$$

Expanding the squared norm inside the expectation:

$$\mathrm{Var}(g_c) = \mathbb{E}\Big[\|g_c - \bar{g}(Z)\|^2 + \|\bar{g}(Z) - \mu'\|^2 + 2\langle g_c - \bar{g}(Z),\, \bar{g}(Z) - \mu'\rangle\Big]. \tag{10}$$

We analyze these terms using the Law of Iterated Expectations, $\mathbb{E}[\cdot] = \mathbb{E}_Z[\mathbb{E}_\tau[\cdot \mid Z]]$.

1. The cross-term vanishes. The validity of this step relies only on the definition of conditional expectation; no independence assumption between the outer and inner randomness is required. Conditioned on Z, the term $\bar{g}(Z) - \mu'$ is deterministic. Thus:

$$\begin{aligned}\mathbb{E}\big[\langle g_c - \bar{g}(Z),\, \bar{g}(Z) - \mu'\rangle\big] &= \mathbb{E}_Z\Big[\mathbb{E}_\tau\big[\langle g_c - \bar{g}(Z),\, \bar{g}(Z) - \mu'\rangle \mid Z\big]\Big] \\ &= \mathbb{E}_Z\Big[\big\langle \underbrace{\mathbb{E}_\tau[g_c - \bar{g}(Z) \mid Z]}_{=0},\, \bar{g}(Z) - \mu'\big\rangle\Big] = 0.\end{aligned}$$

The inner expectation is zero because $\mathbb{E}_\tau[g_c \mid Z] = \bar{g}(Z)$ by definition.

2. Term I: Variance of the conditional expectation. The term $\|\bar{g}(Z) - \mu'\|^2$ depends only on Z. Its expectation is the variance of the conditional mean:

$$\mathbb{E}_Z\big[\|\bar{g}(Z) - \mu'\|^2\big] = \mathrm{Var}_Z(\bar{g}(Z)) = \mathrm{Var}_{x,\mathcal{O}}(\mathbb{E}_\tau[g_c \mid x, \mathcal{O}])\,.$$

3. Term II: Expectation of the conditional variance. The term $\|g_c - \bar{g}(Z)\|^2$ measures the deviation of g_c from its conditional mean $\bar{g}(Z)$. Taking the nested expectation:

$$\mathbb{E}_Z\Big[\mathbb{E}_\tau\big[\|g_c - \bar{g}(Z)\|^2 \mid Z\big]\Big] = \mathbb{E}_Z\Big[\mathrm{Var}_\tau(g_c \mid Z)\Big].$$

Substituting $Z = (x, \mathcal{O})$, we recover:

$$\text{Term (II)} = \mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})]\,.$$

Combining the terms, we verify the decomposition:

$$\mathrm{Var}(g_c) = \underbrace{\mathrm{Var}_{x,\mathcal{O}}(\mathbb{E}_\tau[g_c \mid x, \mathcal{O}])}_{\text{(I) Data \& Model Uncertainty}} + \underbrace{\mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})]}_{\text{(II) Reasoning Trace Noise}}. \qquad \square$$

A.4 A condition for total-variance reduction

In this appendix, we give a condition under which the prompt-paired composite estimator g_c also reduces the *full total variance*. This condition is additional to the conditional trace-noise contraction in Proposition 4.1 and does not follow from it automatically.

Proposition A.4 (A condition for total gradient variance reduction)**.** *Fix a model state θ, and take all expectations, variances, and covariances with respect to the joint randomness of the gradient estimator at this state, under which g_t and g_e are jointly defined random vectors. This randomness may include minibatch sampling, outcome generation, and trace/empty-trace sampling, depending on the estimator construction. For a random vector estimator g, define the scalar total gradient variance*

$$\mathrm{Var}(g) := \mathbb{E}\big[\|g - \mathbb{E}[g]\|^2\big] = \mathrm{tr}(\mathrm{Cov}(g)).$$

For random vectors X, Y, the covariance matrix is

$$\mathrm{Cov}(X, Y) := \mathbb{E}\big[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])^\top\big]\,.$$

Let

$$\Delta := g_t - g_e, \qquad g_c(\alpha) := g_e + \alpha\Delta, \qquad \alpha \in [0, 1].$$

Then

$$\mathrm{Var}(g_t) = \mathrm{Var}(g_e) + \mathrm{Var}(\Delta) + 2\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)), \tag{11}$$

$$\mathrm{Var}(g_c(\alpha)) = \mathrm{Var}(g_e) + \alpha^2\mathrm{Var}(\Delta) + 2\alpha\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)). \tag{12}$$

Consequently,

$$\mathrm{Var}(g_t) - \mathrm{Var}(g_c(\alpha)) = (1 - \alpha)\Big((1 + \alpha)\mathrm{Var}(\Delta) + 2\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta))\Big). \tag{13}$$

Hence, for any $\alpha \in [0, 1)$,

$$\mathrm{Var}(g_c(\alpha)) < \mathrm{Var}(g_t) \iff \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)) > -\frac{1 + \alpha}{2}\,\mathrm{Var}(\Delta). \tag{14}$$

In particular, in the nontrivial case $\mathrm{Var}(\Delta) > 0$, any nonnegative trace covariance $\mathrm{tr}(\mathrm{Cov}(g_e, \Delta)) \geq 0$ is sufficient for $\mathrm{Var}(g_c(\alpha)) < \mathrm{Var}(g_t)$. If instead $\mathrm{Var}(\Delta) = 0$, then $\mathrm{Var}(g_c(\alpha)) = \mathrm{Var}(g_t)$ for every $\alpha \in [0, 1]$.

Proof. Using

$$g_t = g_e + \Delta, \qquad g_c(\alpha) = g_e + \alpha\Delta,$$

together with the identity

$$\begin{aligned}\mathrm{Var}(X+Y) &= \mathrm{Var}(X) + \mathrm{Var}(Y) + \mathrm{tr}(\mathrm{Cov}(X,Y)) + \mathrm{tr}(\mathrm{Cov}(Y,X)) \\ &= \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\ \mathrm{tr}(\mathrm{Cov}(X,Y)),\end{aligned}$$

we obtain

$$\mathrm{Var}(g_t) = \mathrm{Var}(g_e + \Delta) = \mathrm{Var}(g_e) + \mathrm{Var}(\Delta) + 2\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)),$$

which proves Equation (11). Similarly,

$$\mathrm{Var}(g_c(\alpha)) = \mathrm{Var}(g_e + \alpha\Delta) = \mathrm{Var}(g_e) + \alpha^2\mathrm{Var}(\Delta) + 2\alpha\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)),$$

which proves Equation (12).

Subtracting Equation (12) from Equation (11) yields

$$\mathrm{Var}(g_t) - \mathrm{Var}(g_c(\alpha)) = \big(1-\alpha^2\big)\mathrm{Var}(\Delta) + 2(1-\alpha)\,\mathrm{tr}(\mathrm{Cov}(g_e, \Delta)).$$

Factoring out $(1-\alpha)$ gives Equation (13):

$$\mathrm{Var}(g_t) - \mathrm{Var}(g_c(\alpha)) = (1-\alpha)\Big((1+\alpha)\mathrm{Var}(\Delta) + 2\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta))\Big).$$

Now fix any $\alpha \in [0, 1)$. Since $1-\alpha > 0$, the strict inequality $\mathrm{Var}(g_c(\alpha)) < \mathrm{Var}(g_t)$ is equivalent to

$$(1+\alpha)\mathrm{Var}(\Delta) + 2\ \mathrm{tr}(\mathrm{Cov}(g_e, \Delta)) > 0.$$

Rearranging proves Equation (14). Finally, if $\mathrm{Var}(\Delta) > 0$ and $\mathrm{tr}(\mathrm{Cov}(g_e, \Delta)) \geq 0$, then the right-hand side of Equation (13) is strictly positive for every $\alpha \in [0, 1)$, which proves the last claim. If instead $\mathrm{Var}(\Delta) = 0$, then Equation (13) gives $\mathrm{Var}(g_t) - \mathrm{Var}(g_c(\alpha)) = 0$ for every $\alpha \in [0, 1]$. □

Remark A.5 (Interpretation)**.** *Proposition 4.1 guarantees contraction only of the conditional reasoning-trace-noise term in Equation* (7)*, namely* $\mathbb{E}_{x,\mathcal{O}}[\mathrm{Var}_\tau(g_c \mid x, \mathcal{O})]$*. Proposition A.4 shows that this alone does* not *imply reduction of the full total gradient variance, because the latter also depends on the covariance structure between the empty-trace branch* g_e *and the difference term* $\Delta = g_t - g_e$*, through* $\mathrm{tr}(\mathrm{Cov}(g_e, \Delta))$*. In our setting,* g_t *and* g_e *are constructed from the same prompt and the same preference-labeling pipeline, although the trace-based and empty-trace response pairs are sampled separately. This motivates checking the condition empirically through total gradient variance diagnostics rather than treating total gradient variance reduction as a direct consequence of Proposition 4.1.*

A.5 Proof of Theorem 4.3

In this appendix, we provide the proof of Theorem 4.3. Our analysis follows the standard framework for non-convex optimization with biased gradient estimators [2, 9], making explicit how the bias and variance of the composite estimator g_c appear in the optimization guarantee. The bounded-bias/variance assumption introduced below is used only for the final simplified consequence, not for the main bound in Theorem 4.3.

Remark A.6 (Hierarchy of randomness)**.** *Consistent with Section 4.3 and in contrast to the fixed-*$(x, \mathcal{O})$ *analysis in Appendices A.1–A.2, the expectation* $\mathbb{E}_k[\cdot]$ *in this proof integrates over the full joint randomness at iteration* k*. Unlike the instance-conditioned analysis in Appendix A.1, which fixes* $(x, \mathcal{O})$ *and isolates the remaining trace randomness, the stochasticity here includes:*

- *(i)* ***Outer randomness:*** *the randomness in* $(x, \mathcal{O})$*, including minibatch sampling and outcome generation;*
- *(ii)* ***Inner randomness:*** *the remaining reasoning-trace randomness* τ *under fixed* $(x, \mathcal{O})$*.*

Consequently, the variance term $\mathrm{Var}_k(g_c)$ *in this proof is the full variance in Equation* (7)*, incorporating both the outer* $(x, \mathcal{O})$ *variability and the conditional trace-noise component.*

Theorem A.7 (Restatement of Theorem 4.3 and a bounded-bias/variance consequence)**.** *Let the global objective be*

$$\mathcal{L}(\theta) := \mathbb{E}_{(x,y^\pm)\sim\mathcal{D}}[\ell_m(\theta; x, y^\pm)].$$

We adopt the following standard assumptions for stochastic non-convex optimization:

1. ***Lower bound:*** *The objective function is bounded below by a finite scalar $\mathcal{L}^* > -\infty$.*
2. ***L-smoothness:*** *The function $\mathcal{L}$ is L-smooth, i.e., $\|\nabla\mathcal{L}(\theta_1) - \nabla\mathcal{L}(\theta_2)\| \leq L\|\theta_1 - \theta_2\|$.*
3. ***Uniformly bounded bias and variance (only for the final inequality):*** *The global bias and variance of the estimator $g_c(\theta_k)$ are bounded for all k almost surely with respect to the outer training randomness:*

$$\|\mathrm{Bias}_k(g_c)\|^2 \leq \sigma_b^2 \quad \textit{and} \quad \mathrm{Var}_k(g_c) \leq \sigma_v^2.$$

Consider the SGD update rule $\theta_{k+1} = \theta_k - \eta\, g_c(\theta_k)$ with learning rate $\eta \leq \frac{1}{L}$. Let $\mathbb{E}_k[\cdot]$ denote the expectation conditioned on θ_k. We define the global bias and variance as

$$\mathrm{Bias}_k(g_c) := \mathbb{E}_k[g_c(\theta_k)] - \nabla_\theta\mathcal{L}(\theta_k),$$
$$\mathrm{Var}_k(g_c) := \mathbb{E}_k\left[\|g_c(\theta_k) - \mathbb{E}_k[g_c(\theta_k)]\|^2\right].$$

Then, for any number of iterations $K \geq 1$, the average squared gradient norm satisfies

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\mathcal{L}(\theta_k)\|^2\right] \leq \frac{2(\mathcal{L}(\theta_0) - \mathcal{L}^*)}{K\eta} + \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\Big[\|\mathrm{Bias}_k(g_c)\|^2 + \eta L\,\mathrm{Var}_k(g_c)\Big]$$
$$\leq \frac{2(\mathcal{L}(\theta_0) - \mathcal{L}^*)}{K\eta} + \left(\sigma_b^2 + \eta L\sigma_v^2\right).$$

Proof. Our analysis follows the standard framework for non-convex optimization with biased estimators [2, 9]. The key adaptation is the explicit handling of the composite error term arising from the reasoning model's joint stochasticity.

Since $\mathcal{L}(\theta)$ is L-smooth, the descent lemma guarantees:

$$\mathcal{L}(\theta_{k+1}) \leq \mathcal{L}(\theta_k) + \langle\nabla\mathcal{L}(\theta_k), \theta_{k+1} - \theta_k\rangle + \frac{L}{2}\|\theta_{k+1} - \theta_k\|^2. \tag{15}$$

Substituting the SGD update $\theta_{k+1} - \theta_k = -\eta g_c(\theta_k)$ and taking the conditional expectation $\mathbb{E}_k[\cdot]$:

$$\mathbb{E}_k[\mathcal{L}(\theta_{k+1})] \leq \mathcal{L}(\theta_k) \underbrace{-\eta\langle\nabla\mathcal{L}(\theta_k), \mathbb{E}_k[g_c(\theta_k)]\rangle}_{T_1} + \underbrace{\frac{L\eta^2}{2}\mathbb{E}_k[\|g_c(\theta_k)\|^2]}_{T_2}. \tag{16}$$

Step 1: Decomposition of moments. Let $b_k := \mathrm{Bias}_k(g_c) = \mathbb{E}_k[g_c(\theta_k)] - \nabla\mathcal{L}(\theta_k)$. We decompose the first and second moments of the estimator:

$$\mathbb{E}_k[g_c(\theta_k)] = \nabla\mathcal{L}(\theta_k) + b_k, \tag{17}$$
$$\mathbb{E}_k[\|g_c(\theta_k)\|^2] = \|\mathbb{E}_k[g_c(\theta_k)]\|^2 + \mathrm{Var}_k(g_c)$$
$$= \|\nabla\mathcal{L}(\theta_k) + b_k\|^2 + \mathrm{Var}_k(g_c). \tag{18}$$

Step 2: Relaxation and cancellation. Substituting Equation (17) and Equation (18) back into Equation (16):

$$\mathbb{E}_k[\mathcal{L}(\theta_{k+1})] \leq \mathcal{L}(\theta_k) - \eta\langle\nabla\mathcal{L}(\theta_k), \nabla\mathcal{L}(\theta_k) + b_k\rangle + \frac{L\eta^2}{2}\Big(\|\nabla\mathcal{L}(\theta_k) + b_k\|^2 + \mathrm{Var}_k(g_c)\Big).$$

Using the step-size assumption $\eta \leq \frac{1}{L}$, we have $\frac{L\eta^2}{2} \leq \frac{\eta}{2}$. To derive a tight bound, we apply this inequality selectively to the squared mean term to facilitate cancellation, while retaining $\frac{L\eta^2}{2}$ for the variance term. Expanding the squared mean term $\|\nabla\mathcal{L}(\theta_k) + b_k\|^2$:

$$\mathbb{E}_k[\mathcal{L}(\theta_{k+1})] \leq \mathcal{L}(\theta_k) \underbrace{-\eta\|\nabla\mathcal{L}(\theta_k)\|^2 - \eta\langle\nabla\mathcal{L}(\theta_k), b_k\rangle}_{\text{Expansion of linear term } T_1}$$
$$+ \underbrace{\frac{\eta}{2}\|\nabla\mathcal{L}(\theta_k)\|^2 + \frac{\eta}{2}\|b_k\|^2 + \frac{\eta}{2}\cdot 2\langle\nabla\mathcal{L}(\theta_k), b_k\rangle}_{\text{Relaxed squared mean term}} + \frac{L\eta^2}{2}\mathrm{Var}_k(g_c).$$

Crucially, the cross-terms involving the bias vector cancel out:

$$\underbrace{-\eta\langle\nabla\mathcal{L}(\theta_k), b_k\rangle}_{\text{from } T_1} + \underbrace{\eta\langle\nabla\mathcal{L}(\theta_k), b_k\rangle}_{\text{from squared mean}} = 0.$$

Combining the gradient norm terms ($-\eta + \frac{\eta}{2} = -\frac{\eta}{2}$), the inequality simplifies to:

$$\mathbb{E}_k[\mathcal{L}(\theta_{k+1})] \leq \mathcal{L}(\theta_k) - \frac{\eta}{2}\|\nabla\mathcal{L}(\theta_k)\|^2 + \frac{\eta}{2}\|b_k\|^2 + \frac{L\eta^2}{2}\mathrm{Var}_k(g_c). \tag{19}$$

Step 3: Telescoping sum. Rearranging Equation (19) to isolate the gradient norm:

$$\|\nabla\mathcal{L}(\theta_k)\|^2 \leq \frac{2}{\eta}\Big(\mathcal{L}(\theta_k) - \mathbb{E}_k[\mathcal{L}(\theta_{k+1})]\Big) + \|\mathrm{Bias}_k(g_c)\|^2 + \eta L\,\mathrm{Var}_k(g_c).$$

We take the total expectation $\mathbb{E}[\cdot]$ over the entire trajectory and average over $k = 0, \ldots, K-1$, using the tower property $\mathbb{E}[\mathbb{E}_k[\mathcal{L}(\theta_{k+1})]] = \mathbb{E}[\mathcal{L}(\theta_{k+1})]$. The first term telescopes:

$$\sum_{k=0}^{K-1}\mathbb{E}\Big[\mathcal{L}(\theta_k) - \mathcal{L}(\theta_{k+1})\Big] = \mathcal{L}(\theta_0) - \mathbb{E}[\mathcal{L}(\theta_K)] \leq \mathcal{L}(\theta_0) - \mathcal{L}^*.$$

Substituting this back yields the final bound:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\mathcal{L}(\theta_k)\|^2\right] &\leq \frac{2(\mathcal{L}(\theta_0) - \mathcal{L}^*)}{K\eta} + \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\Big[\|\mathrm{Bias}_k(g_c)\|^2 + \eta L\,\mathrm{Var}_k(g_c)\Big] \\ &\leq \frac{2(\mathcal{L}(\theta_0) - \mathcal{L}^*)}{K\eta} + \left(\sigma_b^2 + \eta L\sigma_v^2\right),\end{aligned}$$

where the second inequality follows from the uniform bounds

$$\|\mathrm{Bias}_k(g_c)\|^2 \leq \sigma_b^2, \qquad \mathrm{Var}_k(g_c) \leq \sigma_v^2$$

for all k. □

A.6 Global constant α: fixed-reference-parameter analysis and existence under actual training dynamics

For the global constant α used in training, our theory does not provide a universal optimality guarantee for the actual training dynamics. This appendix therefore provides only complementary fixed-reference-parameter analysis and existence results.

Concretely, we analyze the estimator-side effect of a single global constant mixture weight under a fixed parameter sequence. This isolates the dependence on the estimator choice itself and avoids comparing the self-induced optimization trajectories generated by different choices of α. Here, α is shared across all iterates k, rather than varying with k.

Corollary A.8 (Fixed-reference-parameter quadratic form and exact endpoint improvement)**.** *Fix any deterministic reference parameter sequence $\{\bar{\theta}_k\}_{k=0}^{K-1}$. For each k, let $\mathbb{E}_k^{\mathrm{ref}}[\cdot]$ denote the expectation over the full randomness at the fixed state $\bar{\theta}_k$ (minibatch sampling, outcome generation, and trace sampling), and define*

$$\mathcal{E}_k^{\mathrm{ref}}(g) := \mathbb{E}_k^{\mathrm{ref}}\left[\|g(\bar{\theta}_k) - \nabla\mathcal{L}(\bar{\theta}_k)\|^2\right].$$

Also define the trajectory-averaged reference estimator term

$$\bar{\mathcal{E}}_{\mathrm{ref}}(g) := \frac{1}{K}\sum_{k=0}^{K-1}\mathcal{E}_k^{\mathrm{ref}}(g).$$

For a global constant mixing weight $\alpha \in [0,1]$, let

$$g_c(\alpha) = g_e + \alpha(g_t - g_e).$$

Here, α is a single scalar shared across all iterates k, rather than an iteration-dependent weight α_k.

At each reference state $\bar{\theta}_k$, define the random vectors

$$d_k := g_t(\bar{\theta}_k) - g_e(\bar{\theta}_k), \qquad r_k := g_e(\bar{\theta}_k) - \nabla\mathcal{L}(\bar{\theta}_k).$$

Then the trajectory-averaged reference estimator term is a quadratic function of α:

$$\bar{J}_{\mathrm{ref}}(\alpha) := \bar{\mathcal{E}}_{\mathrm{ref}}(g_c(\alpha)) = \bar{A}\,\alpha^2 + \bar{B}\,\alpha + \bar{C}, \tag{20}$$

where

$$\bar{A} = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}_k^{\mathrm{ref}}\left[\|d_k\|^2\right], \tag{21}$$

$$\bar{B} = \frac{2}{K}\sum_{k=0}^{K-1} \mathbb{E}_k^{\mathrm{ref}}[\langle r_k, d_k\rangle], \tag{22}$$

$$\bar{C} = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}_k^{\mathrm{ref}}\left[\|r_k\|^2\right]. \tag{23}$$

Since $\bar{J}_{\mathrm{ref}}(\alpha)$ is continuous on the compact interval $[0,1]$, a minimizer over $[0,1]$ always exists. Assume the nontrivial case $\bar{A} > 0$, so that $\bar{J}_{\mathrm{ref}}(\alpha)$ is strictly convex and the minimizer is unique. Define

$$\alpha_{\mathrm{unc}}^{\mathrm{ref}} := -\frac{\bar{B}}{2\bar{A}}, \qquad \alpha_{\mathrm{ref}}^{\star} := \Pi_{[0,1]}\big(\alpha_{\mathrm{unc}}^{\mathrm{ref}}\big).$$

If $\alpha_{\mathrm{unc}}^{\mathrm{ref}} \in [0,1]$, then

$$\bar{\mathcal{E}}_{\mathrm{ref}}(g_t) - \bar{\mathcal{E}}_{\mathrm{ref}}(g_c(\alpha_{\mathrm{ref}}^{\star})) = (1-\alpha_{\mathrm{ref}}^{\star})^2\,\bar{A} = (1-\alpha_{\mathrm{ref}}^{\star})^2\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_k^{\mathrm{ref}}\left[\|g_t(\bar{\theta}_k) - g_e(\bar{\theta}_k)\|^2\right]. \tag{24}$$

If $\alpha_{\mathrm{unc}}^{\mathrm{ref}} \notin [0,1]$, then the projected minimizer $\alpha_{\mathrm{ref}}^{\star}$ still satisfies

$$\bar{\mathcal{E}}_{\mathrm{ref}}(g_c(\alpha_{\mathrm{ref}}^{\star})) \le \min\Big\{\bar{\mathcal{E}}_{\mathrm{ref}}(g_e),\, \bar{\mathcal{E}}_{\mathrm{ref}}(g_t)\Big\}. \tag{25}$$

Proof. For each fixed k, with the same global constant α shared across all iterates, we have

$$g_c(\alpha;\bar{\theta}_k) - \nabla\mathcal{L}(\bar{\theta}_k) = g_e(\bar{\theta}_k) + \alpha\big(g_t(\bar{\theta}_k) - g_e(\bar{\theta}_k)\big) - \nabla\mathcal{L}(\bar{\theta}_k) = r_k + \alpha d_k.$$

Therefore,

$$\|g_c(\alpha;\bar{\theta}_k) - \nabla\mathcal{L}(\bar{\theta}_k)\|^2 = \|r_k + \alpha d_k\|^2 = \|r_k\|^2 + 2\alpha\langle r_k, d_k\rangle + \alpha^2\|d_k\|^2.$$

Taking $\mathbb{E}_k^{\mathrm{ref}}[\cdot]$ and averaging over $k = 0, \ldots, K-1$ yields

$$\bar{J}_{\mathrm{ref}}(\alpha) = \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_k^{\mathrm{ref}}\left[\|r_k\|^2 + 2\alpha\langle r_k, d_k\rangle + \alpha^2\|d_k\|^2\right],$$

which is exactly the quadratic form (20) with coefficients (21)–(23).

Since $\bar{A} > 0$, the quadratic is strictly convex, and its unconstrained minimizer is

$$\alpha_{\mathrm{unc}}^{\mathrm{ref}} = -\frac{\bar{B}}{2\bar{A}}.$$

If $\alpha_{\mathrm{unc}}^{\mathrm{ref}} \in [0,1]$, then $\alpha_{\mathrm{ref}}^{\star} = \alpha_{\mathrm{unc}}^{\mathrm{ref}}$, and completing the square gives

$$\bar{J}_{\mathrm{ref}}(\alpha) = \bar{A}\big(\alpha - \alpha_{\mathrm{ref}}^{\star}\big)^2 + \bar{J}_{\mathrm{ref}}(\alpha_{\mathrm{ref}}^{\star}).$$

Evaluating at $\alpha = 1$ yields

$$\bar{J}_{\mathrm{ref}}(1) - \bar{J}_{\mathrm{ref}}(\alpha_{\mathrm{ref}}^{\star}) = (1-\alpha_{\mathrm{ref}}^{\star})^2\bar{A}.$$

Since $\bar{J}_{\mathrm{ref}}(1) = \bar{\mathcal{E}}_{\mathrm{ref}}(g_t)$, this proves Equation (24).

If instead $\alpha_{\mathrm{unc}}^{\mathrm{ref}} \notin [0,1]$, then projection onto $[0,1]$ gives the constrained minimizer

$$\alpha_{\mathrm{ref}}^{\star} = \Pi_{[0,1]}\big(\alpha_{\mathrm{unc}}^{\mathrm{ref}}\big).$$

Hence

$$\bar{J}_{\mathrm{ref}}(\alpha_{\mathrm{ref}}^{\star}) \le \bar{J}_{\mathrm{ref}}(0) = \bar{\mathcal{E}}_{\mathrm{ref}}(g_e), \qquad \bar{J}_{\mathrm{ref}}(\alpha_{\mathrm{ref}}^{\star}) \le \bar{J}_{\mathrm{ref}}(1) = \bar{\mathcal{E}}_{\mathrm{ref}}(g_t),$$

which proves Equation (25). □

We next record a complementary existence statement for a fixed global α, where a fixed global mixing coefficient α is chosen before the self-induced trajectory is generated. A fixed global α is also consistent with the training procedure used in our experiments, provides a stable estimator-level bias–variance trade-off, and avoids the additional complexity of state- or iteration-dependent weighting.

Proposition A.9 (Existence of a training-horizon-optimal fixed mixing coefficient)**.** *Fix an initialization θ_0, a learning rate η, a training horizon K, and all other training hyperparameters. For each $\alpha \in [0,1]$, let*

$$\{\theta_k(\alpha)\}_{k=0}^{K-1}$$

denote the (random) iterate sequence generated by SGD with the fixed global mixing coefficient α. Define the trajectory-averaged estimator term

$$\Phi_K(\alpha) := \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}[\mathcal{E}_k(\alpha)]\,,$$

where

$$\mathcal{E}_k(\alpha) := \|\mathrm{Bias}_k(g_c(\alpha))\|^2 + \eta L\,\mathrm{Var}_k(g_c(\alpha)).$$

Assume that the resulting function $\Phi_K : [0,1] \to \mathbb{R}$ is well-defined and continuous. Then there exists $\alpha_K^\star \in [0,1]$ such that

$$\Phi_K(\alpha_K^\star) = \min_{\alpha\in[0,1]} \Phi_K(\alpha).$$

Moreover, when $\eta = 1/L$,

$$\Phi_K(\alpha) = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\big[\|g_c(\alpha;\theta_k(\alpha)) - \nabla\mathcal{L}(\theta_k(\alpha))\|^2\big]\,.$$

Proof. For each fixed $\alpha \in [0,1]$, the corresponding SGD update rule with global mixing coefficient α generates a trajectory $\{\theta_k(\alpha)\}_{k=0}^{K-1}$ and hence a well-defined trajectory-averaged estimator term $\Phi_K(\alpha)$. By assumption, $\Phi_K(\alpha)$ is continuous on the compact interval $[0,1]$. Therefore, by the Weierstrass extreme-value theorem, there exists $\alpha_K^\star \in [0,1]$ such that

$$\Phi_K(\alpha_K^\star) = \min_{\alpha\in[0,1]} \Phi_K(\alpha).$$

For the final statement, when $\eta = 1/L$, we have

$$\mathcal{E}_k(\alpha) = \|\mathrm{Bias}_k(g_c(\alpha))\|^2 + \mathrm{Var}_k(g_c(\alpha)).$$

By the bias–variance decomposition at iteration k,

$$\mathbb{E}_k\big[\|g_c(\alpha) - \nabla\mathcal{L}(\theta_k(\alpha))\|^2\big] = \|\mathrm{Bias}_k(g_c(\alpha))\|^2 + \mathrm{Var}_k(g_c(\alpha)),$$

where $\mathbb{E}_k[\cdot]$ denotes expectation conditioned on the current iterate $\theta_k(\alpha)$. Hence,

$$\mathcal{E}_k(\alpha) = \mathbb{E}_k\big[\|g_c(\alpha) - \nabla\mathcal{L}(\theta_k(\alpha))\|^2\big]\,.$$

Taking outer expectation and averaging over $k = 0, \ldots, K-1$ yields

$$\Phi_K(\alpha) = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\big[\|g_c(\alpha;\theta_k(\alpha)) - \nabla\mathcal{L}(\theta_k(\alpha))\|^2\big]\,.$$

□

Remark A.10 (Scope of the two global-α results)**.** *Corollary A.8 and Proposition A.9 play complementary roles. The fixed-reference-parameter corollary yields a stronger quantitative result: a quadratic dependence on α, a closed-form projected optimum, and exact endpoint-improvement statements. By contrast, Proposition A.9 reflects the actual training setup, in which a fixed global α is chosen before the self-induced trajectory $\{\theta_k(\alpha)\}_{k=0}^{K-1}$ is generated, but it provides only an existence statement under a continuity assumption rather than a closed-form rule for choosing α in practice.*

Table 5: The hyperparameters of DPO for each training setting.

Setting	β	learning rate
DeepSeek-R1-Distill-Qwen-7B	0.01	7e-7
DeepSeek-R1-Distill-Qwen-1.5B	0.01	7e-7
DeepSeek-R1-0528-Qwen3-8B	0.01	7e-7
MiMo-7B-RL	0.01	7e-7

Table 6: The hyperparameters of SimPO for each training setting.

Setting	β	γ	learning rate
DeepSeek-R1-Distill-Qwen-7B	2.5	1.0	7e-7
DeepSeek-R1-Distill-Qwen-1.5B	2.5	1.0	7e-7
DeepSeek-R1-0528-Qwen3-8B	2.5	1.0	7e-7
MiMo-7B-RL	2.5	0.3	7e-7

Table 7: The hyperparameters of BVPO for each training setting.

Setting	β	α	learning rate
DeepSeek-R1-Distill-Qwen-7B	0.01	0.3	7e-7
DeepSeek-R1-Distill-Qwen-1.5B	0.01	0.8	7e-7
DeepSeek-R1-0528-Qwen3-8B	0.01	0.3	7e-7
MiMo-7B-RL	0.01	0.1	7e-7

B Experiment details

B.1 Training details

Unless otherwise stated, we use a consistent batch size of 128 and train all methods for 1 epoch in all settings. The AdamW optimizer [21] is used. The max sequence length is set to 4096, and a cosine learning rate schedule with 10% warm-up steps is used. The hyperparameters for each method are grid-searched following prior practices [22, 27] and are shown in Table 5 for DPO, Table 6 for SimPO, and Table 7 for our BVPO, respectively. All hyperparameters were selected on the same held-out validation split with the same search budget across methods.

For the DPO w/ data mixing baseline, we double the batch size so that the number of training steps matches that of the other methods like DPO and BVPO.

All training runs are conducted on 8 GPUs. Compared with standard DPO, BVPO introduces an additional empty-trace branch at each training step. On DeepSeek-R1-Distill-Qwen-7B, this increases the wall-clock training time from 206 minutes to 279 minutes, corresponding to a $1.35\times$ overhead. In comparison, DPO with data mixing requires 360 minutes on DeepSeek-R1-Distill-Qwen-7B, corresponding to a $1.75\times$ overhead. The overhead of BVPO remains moderate because the empty-trace branch operates on substantially shorter sequences than the trace-based branch. By contrast, DPO with data mixing incurs higher overhead because it treats empty-trace samples as ordinary training samples, resulting in a less structured batching scheme.

B.2 Evaluation details

For the alignment benchmarks (AlpacaEval 2 and Arena-Hard), we set the maximum generation length to 8192 tokens and evaluate only the final-answer portion of each response, following [6]. Unless otherwise stated, GPT-4o-2024-11-20 is used as the judge model for the reported alignment benchmark evaluations.

For math reasoning benchmarks, we increase the maximum generation length to 32768 tokens to accommodate problems requiring extended reasoning and to ensure a sufficiently large context window, following [11, 15].

B.3 Data generation

For the trace-based data, we use the experimented models' official chat template to sample responses, which allows free-form reasoning trace generation.

Template for sampling trace-based data

<|begin_of_sentence|><|User|>{QUESTION}<|Assistant|><think>

For the empty-trace data, we explicitly disable reasoning trace sampling by additionally appending </think> at the beginning of the assistant turn, since the special token </think> denotes the end of reasoning trace generation. For the sampled responses, we prepend the special token </think> so that they remain consistent with the official chat template that generates reasoning traces.

Template for sampling empty-trace data

<|begin_of_sentence|><|User|>{QUESTION}<|Assistant|><think></think>

C More results

C.1 Additional controlled comparisons and robustness checks

This appendix reports additional comparisons and robustness checks that complement the main results in Section 5. The purpose is to probe alternative explanations for the empirical gain of BVPO and to test whether the observed pattern remains consistent across larger model scales, pair-construction pipelines, and practical choices of the mixing coefficient.

Taken together, these additional results are consistent with the interpretation that BVPO's empirical gain is not explained solely by extra empty-trace supervision, simple trace-token downweighting, or hyperparameter tuning. They also provide further empirical support for the estimator-level view developed in the main text.

C.1.1 Matched-expectation Bernoulli baseline

To provide a more targeted probe of the role of deterministic within-step averaging, we compare BVPO with a matched-expectation Bernoulli baseline, which replaces BVPO's deterministic within-step averaging with stochastic branch selection and consequently matches BVPO's per-example gradient estimator in expectation over branch selection. The corresponding empirical comparison is reported in Table 3, where BVPO achieves higher scores than this baseline on all reported alignment metrics.

The formal construction and theoretical comparison are given in Appendix C.2.1. In particular, Proposition C.1 establishes the matched-expectation property for a fixed prompt-paired branch sample, Proposition C.2 shows the additional variance introduced by stochastic branch selection under the same fixed-sample comparison, and Proposition C.3 compares its conditional variance with BVPO under the same fixed-$(x, \mathcal{O})$ setup as Proposition 4.1. Taken together, these results clarify why the Bernoulli baseline serves as a more mechanism-specific comparison than direct data mixing: it preserves BVPO's per-example gradient estimator over branch-selection randomness.

We also report training-time observable empirical total gradient variance diagnostics in Table 4, following the definitions in Appendix C.2.3. Relative to the matched-expectation Bernoulli baseline, BVPO reduces observable total gradient variance by 24.7%, 25.9%, 26.7%, 27.1%, and 26.7% at epochs 0, 0.25, 0.5, 0.75, and 1.0, respectively. For reference, Table 4 also reports the corresponding reduction relative to DPO, as well as the trace-associated variance ratio of g_t over training. Together, these diagnostics are consistent with the view that BVPO's gain is not explained by matched expectation alone, and are also consistent with lower gradient-estimator variance from deterministic averaging.

Table 8: Additional baselines and comparisons on R1-Qwen-1.5B. $K = 2$ and $K = 4$ multi-trace averaging DPO use two and four independently sampled trace-based preference pairs per prompt, respectively, without the empty-trace branch; trace-only token weighting assigns weight 0.75 to reasoning-trace tokens and 1.0 to final-answer tokens within DPO. In this setting, BVPO achieves the highest scores among the compared methods.

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
Base	4.4	5.4	6.3	5.5	6.9	6.9
SimPO	5.5	6.2	8.4	4.5	4.6	3.8
DPO	5.1	6.4	8.0	7.2	7.8	7.1
$K = 2$ multi-trace averaging DPO	5.8	6.1	7.5	5.9	6.9	7.2
$K = 4$ multi-trace averaging DPO	6.3	5.8	7.2	7.0	7.4	7.5
DPO w/ trace weighting	5.0	6.5	8.3	7.0	6.9	6.9
BVPO	**6.5**	**6.9**	**8.7**	**7.6**	**10.2**	**10.7**

Table 9: Robustness to reward model used for preference labeling. Training preference pairs are re-annotated using Skywork-Reward-V2-Qwen3-8B [20] under the same pipeline. Under these re-annotated preference pairs, BVPO achieves higher scores than DPO on the reported alignment metrics.

Method	*Thinking*			*NoThinking*		
	Arena-Hard	AlpacaEval 2		Arena-Hard	AlpacaEval 2	
	Win Rate (%)	Win Rate (%)	LC Win Rate (%)	Win Rate (%)	Win Rate (%)	LC Win Rate (%)
DPO (Skywork RM)	7.0	6.1	7.0	5.3	7.3	7.0
BVPO (Skywork RM)	**7.3**	**7.6**	**8.6**	**8.0**	**9.1**	**8.2**

C.1.2 Additional baselines and comparisons on R1-Qwen-1.5B

In addition to the standard baselines in the main text, we compare BVPO against three additional baselines on R1-Qwen-1.5B. The first is $K = 2$ multi-trace averaging DPO, which averages two trace-based preference samples per prompt. We additionally evaluate $K = 4$ multi-trace averaging DPO, which averages four independently sampled trace-based preference pairs per prompt. These baselines test whether variance reduction from additional trace-based averaging can provide similar level of help. The third is trace-only token weighting, which assigns weight 0.75 to reasoning-trace tokens and weight 1.0 to final-answer tokens when forming the sequence-level log-probability used in DPO. This baseline tests whether simply reducing the contribution of reasoning-trace tokens is sufficient, without the estimator-level empty-trace branch used by BVPO. For reference, Table 8 also includes Base, SimPO, and DPO. In this setting, none of the additional baselines matches BVPO, which achieves the highest scores among the compared methods on the reported alignment metrics. Relative to standard DPO, training costs are $1.9\times$ for $K = 2$, $3.7\times$ for $K = 4$, and $1.4\times$ for BVPO.

C.1.3 Robustness to reward model used for preference labeling

We further examine whether the empirical comparison between BVPO and baselines is sensitive to the reward model used for preference labeling. We therefore re-annotate the R1-Qwen-1.5B training preference pairs using Skywork-Reward-V2-Qwen3-8B [20] under the same labeling pipeline and training setup. As shown in Table 9, BVPO continues to outperform DPO on the reported alignment metrics under these re-annotated pairs. This suggests that the observed advantage of BVPO is not tied to the original reward model used for labeling.

C.1.4 Sensitivity to the mixing coefficient

We test sensitivity to the fixed mixing coefficient α on both R1-Qwen-1.5B and R1-Qwen-7B. The results are shown in Table 10 and Table 11. Within reasonable ranges, performance varies modestly around the value used in the main experiment. These results show that BVPO is not overly sensitive to the mixing coefficient α value.

Table 10: α-sensitivity on R1-Qwen-1.5B. Across the tested range, performance varies modestly around the value used in the main experiments.

α	*Thinking*			*NoThinking*		
	Arena-Hard	**AlpacaEval 2**		**Arena-Hard**	**AlpacaEval 2**	
	Win Rate (%)	**Win Rate (%)**	**LC Win Rate (%)**	**Win Rate (%)**	**Win Rate (%)**	**LC Win Rate (%)**
0.7	6.1	6.4	8.1	7.4	9.3	9.8
0.8	**6.5**	**6.9**	**8.7**	**7.6**	**10.2**	**10.7**
0.9	6.3	6.7	8.4	7.6	8.6	8.9

Table 11: α-sensitivity on R1-Qwen-7B. Across the tested range, performance varies modestly around the value used in the main experiments.

α	*Thinking*			*NoThinking*		
	Arena-Hard	**AlpacaEval 2**		**Arena-Hard**	**AlpacaEval 2**	
	Win Rate (%)	**Win Rate (%)**	**LC Win Rate (%)**	**Win Rate (%)**	**Win Rate (%)**	**LC Win Rate (%)**
0.2	21.2	21.6	24.6	22.9	23.0	23.6
0.3	**22.5**	**21.8**	**24.7**	22.7	22.6	22.9
0.4	22.2	21.3	23.2	**25.3**	**23.8**	**24.6**

Table 12: Additional naive data-mixing results on R1-Qwen-7B and R1-Qwen-14B. Across both larger models, BVPO achieves higher scores than DPO w/ data mixing on the reported alignment metrics, consistent with the view that the empirical gain is not explained solely by adding empty-trace supervision.

Method	*Thinking*			*NoThinking*		
	Arena-Hard	**AlpacaEval 2**		**Arena-Hard**	**AlpacaEval 2**	
	Win Rate (%)	**Win Rate (%)**	**LC Win Rate (%)**	**Win Rate (%)**	**Win Rate (%)**	**LC Win Rate (%)**
R1-Qwen-7B						
Base	16.3	15.7	18.4	16.7	15.2	17.0
DPO	19.1	18.3	20.4	17.7	19.3	20.7
DPO w/ data mixing	20.4	20.4	22.4	21.5	21.6	22.7
BVPO	**22.5**	**21.8**	**24.7**	**22.7**	**22.6**	**22.9**
R1-Qwen-14B						
Base	44.9	33.7	42.1	44.2	30.8	34.3
DPO	61.0	51.0	48.2	45.3	42.1	40.1
DPO w/ data mixing	61.3	50.7	48.3	49.3	44.4	48.8
BVPO	**63.3**	**51.4**	**49.6**	**52.2**	**45.2**	**49.5**

C.1.5 Additional data-mixing results on larger models

The main text reports the 1.5B comparison against naive empty-trace data mixing. Here we extend the same comparison to R1-Qwen-7B and R1-Qwen-14B. The data-mixing baseline differs only in whether the two branches are combined within-step or treated as separate DPO training instances. In both settings, BVPO achieves higher scores than DPO w/ data mixing on the reported alignment metrics, consistent with the same empirical pattern observed at 1.5B. In particular, on R1-Qwen-14B, BVPO improves Arena-Hard from 61.3 to 63.3 under Thinking and from 49.3 to 52.2 under NoThinking; see Table 12.

C.2 Derivations for controlled comparisons

This appendix collects the derivations and precise diagnostic definitions used by the additional controlled comparisons in Appendix C.1. These materials specify the comparison constructions, variance comparisons, and practical diagnostics underlying the corresponding empirical results and their interpretation in the main text.

C.2.1 Matched-expectation Bernoulli baseline

To more directly compare deterministic within-step averaging with stochastic branch selection, we introduce a Bernoulli baseline that matches BVPO's per-example gradient estimator in expectation over branch-selection randomness, while replacing deterministic averaging with stochastic branch selection.

For a fixed prompt-paired per-example trace-based and empty-trace gradient sample (g_t, g_e), define

$$\Delta := g_t - g_e, \qquad g_c(\alpha) = g_e + \alpha\Delta = \alpha g_t + (1-\alpha)g_e, \qquad \alpha \in [0,1].$$

Let $\xi \sim \mathrm{Bernoulli}(\alpha)$ be independent of the branch sample (g_t, g_e). We define the Bernoulli baseline by

$$g_{\mathrm{Bern}}(\alpha;\xi) := \xi g_t + (1-\xi)g_e = g_e + \xi\Delta. \tag{26}$$

Conditioned on the fixed prompt-paired branch sample (g_t, g_e), the only remaining randomness in $g_{\mathrm{Bern}}(\alpha;\xi)$ comes from ξ.

Proposition C.1 (Matched-expectation property)**.** *For any fixed prompt-paired branch sample (g_t, g_e) and any $\alpha \in [0,1]$,*

$$\mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha;\xi) \mid g_t, g_e] = g_c(\alpha). \tag{27}$$

Equivalently, conditioned on the same prompt-paired branch sample, the Bernoulli baseline matches BVPO's per-example gradient estimator over the branch-selection randomness.

Proof. Conditioned on (g_t, g_e), the only remaining randomness in $g_{\mathrm{Bern}}(\alpha;\xi)$ comes from $\xi \sim \mathrm{Bernoulli}(\alpha)$. Therefore,

$$\begin{aligned}
\mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha;\xi) \mid g_t, g_e] &= \mathbb{E}_\xi[\xi g_t + (1-\xi)g_e \mid g_t, g_e] \\
&= \mathbb{E}_\xi[\xi]\, g_t + \big(1 - \mathbb{E}_\xi[\xi]\big) g_e \\
&= \alpha g_t + (1-\alpha)g_e \\
&= g_c(\alpha).
\end{aligned}$$

□

Proposition C.2 (Extra variance from stochastic branch selection)**.** *For any fixed prompt-paired branch sample (g_t, g_e) and any $\alpha \in [0,1]$,*

$$\mathrm{Var}_\xi(g_{\mathrm{Bern}}(\alpha;\xi) \mid g_t, g_e) = \alpha(1-\alpha)\|g_t - g_e\|^2. \tag{28}$$

Hence, whenever $\alpha \in (0,1)$ and $g_t \neq g_e$, the Bernoulli baseline introduces strictly positive within-step variance over the branch-selection randomness, whereas deterministic averaging has zero variance with respect to this randomness.

Proof. Using Equation (26) and Equation (27), and writing $\Delta := g_t - g_e$,

$$g_{\mathrm{Bern}}(\alpha;\xi) - \mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha;\xi) \mid g_t, g_e] = (\xi - \alpha)\Delta.$$

Therefore,

$$\mathrm{Var}_\xi(g_{\mathrm{Bern}}(\alpha;\xi) \mid g_t, g_e) = \mathbb{E}_\xi\big[\|(\xi-\alpha)\Delta\|^2 \mid g_t, g_e\big] = \mathbb{E}_\xi\big[(\xi-\alpha)^2\big]\,\|\Delta\|^2.$$

Since $\xi \sim \mathrm{Bernoulli}(\alpha)$,

$$\mathbb{E}_\xi\big[(\xi-\alpha)^2\big] = \mathrm{Var}_\xi(\xi) = \alpha(1-\alpha),$$

which proves Equation (28). □

C.2.2 Conditional-variance comparison with BVPO

We now compare the Bernoulli baseline with BVPO under the same fixed-$(x, \mathcal{O})$ conditional setup as Proposition 4.1. As in the main analysis, we condition on a fixed prompt x and outcome set $\mathcal{O}$, so that g_e is deterministic with respect to trace randomness while g_t remains stochastic through the reasoning trace τ. For the Bernoulli baseline, there is in addition an independent branch-selection random variable $\xi \sim \mathrm{Bernoulli}(\alpha)$. The comparison below therefore contrasts $\mathrm{Var}_{\tau,\xi}(g_{\mathrm{Bern}}(\alpha) \mid x, \mathcal{O})$ with $\mathrm{Var}_\tau(g_c(\alpha) \mid x, \mathcal{O})$ under the same conditioning.

Proposition C.3 (Conditional variance of the Bernoulli baseline)**.** *Let $\alpha \in [0,1]$. Under the same conditioning as Proposition 4.1, let $\xi \sim \mathrm{Bernoulli}(\alpha)$ be independent of the trace randomness τ. Then*

$$\mathrm{Var}_{\tau,\xi}(g_{\mathrm{Bern}}(\alpha) \mid x, \mathcal{O}) = \alpha^2 \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}) + \alpha(1-\alpha)\, \mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right]. \tag{29}$$

By contrast, BVPO satisfies

$$\mathrm{Var}_\tau(g_c(\alpha) \mid x, \mathcal{O}) = \alpha^2 \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}). \tag{30}$$

Hence, for any $\alpha \in (0,1)$,

$$\mathrm{Var}_{\tau,\xi}(g_{\mathrm{Bern}}(\alpha) \mid x, \mathcal{O}) - \mathrm{Var}_\tau(g_c(\alpha) \mid x, \mathcal{O}) = \alpha(1-\alpha)\, \mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right], \tag{31}$$

which is strictly positive whenever

$$\mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right] > 0,$$

equivalently, whenever $g_t \neq g_e$ with positive conditional probability under τ.

Proof. Fix $(x, \mathcal{O})$. Under this conditioning, g_e is deterministic with respect to trace randomness, while g_t remains random through the reasoning trace τ. The Bernoulli baseline is

$$g_{\mathrm{Bern}}(\alpha; \xi) = \xi g_t + (1-\xi) g_e,$$

where $\xi \sim \mathrm{Bernoulli}(\alpha)$ is independent of τ.

Apply the law of total variance by conditioning first on τ:

$$\begin{aligned}\mathrm{Var}_{\tau,\xi}(g_{\mathrm{Bern}}(\alpha) \mid x, \mathcal{O}) = \mathbb{E}_\tau[\mathrm{Var}_\xi(g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}) | x, \mathcal{O}] \\ + \mathrm{Var}_\tau(\mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}] | x, \mathcal{O}).\end{aligned} \tag{32}$$

By Proposition C.2,

$$\mathrm{Var}_\xi(g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}) = \alpha(1-\alpha)\|g_t - g_e\|^2.$$

Therefore,

$$\mathbb{E}_\tau[\mathrm{Var}_\xi(g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}) | x, \mathcal{O}] = \alpha(1-\alpha)\, \mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right]. \tag{33}$$

By Proposition C.1,

$$\mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}] = g_c(\alpha).$$

Since g_e is deterministic under the fixed-$(x, \mathcal{O})$ conditioning, Proposition 4.1 gives

$$\mathrm{Var}_\tau(\mathbb{E}_\xi[g_{\mathrm{Bern}}(\alpha) \mid \tau, x, \mathcal{O}] | x, \mathcal{O}) = \mathrm{Var}_\tau(g_c(\alpha) \mid x, \mathcal{O}) = \alpha^2 \mathrm{Var}_\tau(g_t \mid x, \mathcal{O}). \tag{34}$$

Substituting Equations (33) and (34) into Equation (32) yields Equation (29). Equation (30) is exactly the second identity above, and subtracting it from Equation (29) gives Equation (31). Finally, for $\alpha \in (0,1)$, this gap is strictly positive whenever

$$\mathbb{E}_\tau\left[\|g_t - g_e\|^2 \mid x, \mathcal{O}\right] > 0,$$

equivalently, whenever $g_t \neq g_e$ with positive conditional probability under τ. □

Remark C.4. *Proposition C.3 formalizes the intended role of the Bernoulli baseline: it matches BVPO's per-example gradient estimator over the branch-selection randomness, but introduces the additional conditional-variance term $\alpha(1-\alpha)\mathbb{E}_\tau[\|g_t - g_e\|^2 \mid x, \mathcal{O}]$ due to stochastic branch selection. Empirically, BVPO outperforming this baseline is consistent with the view that lower within-step gradient variance is one contributing factor to the gain.*

C.2.3 Variance-diagnostic definitions

We next record the precise practical diagnostics used in the total gradient variance analyses reported in the main text and in Appendix C.1. These diagnostics are empirical measurements of observable training variance, rather than exact estimators of the conditional reasoning-trace-noise quantity in Proposition 4.1.

Gradient subspace used in all variance diagnostics. Following the main text, all reported gradient variances are measured in the subspace of the final layer's Q/K/V projection weights. Let $\mathcal{S}$ denote this parameter subspace, and let $g|_{\mathcal{S}}$ denote the corresponding flattened gradient vector. For notational simplicity, we suppress the restriction notation below and write g for the resulting vector in the chosen subspace. We use this subspace as a consistent and computationally tractable comparison space across estimators. These diagnostics are intended to test whether estimator-level variance differences appear in a stable observable direction, rather than to exactly estimate Proposition 4.1.

Per-prompt observable empirical total gradient variance. Fix a prompt x and a chosen gradient construction. Draw M independent preference-pair samples using that same construction. Let $g^{(1)}, \dots, g^{(M)}$ be the corresponding gradient samples in the chosen subspace. We define the per-prompt observable empirical total variance as the sample variance

$$\widehat{V}_x(g) := \frac{1}{M-1} \sum_{m=1}^{M} \left\| g^{(m)} - \bar{g}_x \right\|_2^2, \qquad \bar{g}_x := \frac{1}{M} \sum_{m=1}^{M} g^{(m)}. \tag{35}$$

That is, $\widehat{V}_x(g)$ measures observable empirical total gradient variance across repeated preference-pair samples for the same prompt x.

Aggregate observable empirical total gradient variance across prompts. Given a prompt set $\mathcal{X}$, we report the average observable empirical total gradient variance

$$\widehat{V}(g) := \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \widehat{V}_x(g). \tag{36}$$

That is, $\widehat{V}(g)$ averages the per-prompt observable empirical total gradient variances over the prompt set $\mathcal{X}$. In variance plots and tables, we normalize this quantity by the corresponding value of a reference method or estimator (for example, g_t, DPO, or the Bernoulli baseline), so that the reported numbers indicate either relative variance or variance reduction with respect to that reference.

Training-time observable empirical total gradient variance. For the training-time diagnostics, we repeat the aggregate observable empirical total gradient variance measurement above at selected checkpoints (e.g., epochs 0, 0.25, 0.5, 0.75, and 1.0). That is, for each checkpoint s, we compute

$$\widehat{V}^{(s)}(g) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \widehat{V}_x^{(s)}(g),$$

where the superscript indicates that the gradients are evaluated using the model parameters at that checkpoint. In the reported tables, the percentages are obtained by normalizing $\widehat{V}^{(s)}(g)$ with respect to the corresponding value of the same reference method or estimator at that checkpoint.

Trace-associated variance ratio. To measure the relative size of the trace-token-only component in the practical variance of the trace-based gradient, we define a token-partition-based decomposition of each sampled trace-based gradient as

$$g_t = g_t^{\mathrm{trace}} + g_t^{\mathrm{ans}},$$

where the decomposition is defined by partitioning the model-side token log-probability terms in the sampled trace-based loss into reasoning-trace tokens and final-answer tokens. Here, g_t^{trace} is the gradient contribution obtained by retaining only the terms corresponding to reasoning-trace tokens, and g_t^{ans} is the remaining contribution from final-answer tokens.

For each prompt x, we compute the empirical variance of the trace-token-only component:

$$\widehat{V}_x\big(g_t^{\mathrm{trace}}\big) = \frac{1}{M-1} \sum_{m=1}^{M} \left\| \big(g_t^{\mathrm{trace}}\big)^{(m)} - \bar{g}_x^{\mathrm{trace}} \right\|_2^2, \qquad \bar{g}_x^{\mathrm{trace}} := \frac{1}{M} \sum_{m=1}^{M} \big(g_t^{\mathrm{trace}}\big)^{(m)}. \tag{37}$$

We then report the trace-associated variance ratio

$$\widehat{\rho}_{\mathrm{trace}} := \frac{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \widehat{V}_x(g_t^{\mathrm{trace}})}{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \widehat{V}_x(g_t)}. \tag{38}$$

This is a practical diagnostic rather than an exact estimator of the conditional reasoning-trace-noise component in Proposition 4.1. It is not intended as an exact variance decomposition of g_t, but as a diagnostic of whether the trace-token-only component remains a non-negligible part of the observable total variance of the practical trace-based gradient throughout training.

Measurement protocol. The variance diagnostics follow the protocol as follows: we generate $M = 64$ preference pairs per prompt on the same prompt set $\mathcal{X}$ across compared methods, evaluate gradients in the final-layer Q/K/V projection subspace, and aggregate the resulting empirical variance measurements across prompts according to Equation (36) and Equation (38).

D Limitations

BVPO uses an empty-trace branch as the anchor estimator in the proposed composite-gradient framework. While this choice is simple and effective, it is only one possible instantiation. Exploring alternative anchors may yield further improvements. BVPO also incurs modest training overhead by evaluating both trace-based and empty-trace branches. Since empty-trace sequences are much shorter than full reasoning traces, the overhead is moderate in our experiments.

E Broader impact

This work aims to improve preference optimization for large reasoning models while preserving their reasoning capabilities. Such methods may support more useful assistants for education, coding, and mathematical problem solving, where both instruction following and multi-step reasoning are important. Its risks are similar to those of other post-training methods. It should be applied with appropriate dataset governance, evaluation, and deployment safeguards for the intended application domain.

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 claims have been validated.

   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: We discuss limitations in this paper.

   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: We provide assumptions and proofs for our theoretical results.

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: We provide experiment settings and details to reproduce our results.

   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 data and the code will be publicly available.

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: We provide experiment settings and details.

   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: The post-training experiment of LLMs is computationally expensive. Multiple runs would significantly increase costs.

   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: The compute resources are reported.

   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 follows the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: The paper discuss broader impacts.

    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: Our work does not involve releasing data or models that have a high risk for misuse.

    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: The datasets and models used are all properly cited.

    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: [N/A]

    Justification: There is no new assets in this paper.

    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: The paper does not involve crowdsourcing or research with human subjects.

    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 paper does not include study participants.

    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: N/A

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.