Distributionally Robust Token Optimization in RLHF

Yeping Jin
Division of Systems Engineering
Boston University, Boston, MA 02215, USA
yepjin@bu.edu

Jiaming Hu
Department of Math & Statistics
Boston University, Boston, MA 02215, USA
jh7453@bu.edu

Ioannis Ch. Paschalidis
Department of Electrical and Computer Engineering
Boston University, Boston, MA 02215, USA
yannisp@bu.edu

Abstract

Large Language Models (LLMs) tend to respond correctly to prompts that align well with the data they were trained and fine-tuned on. Yet, small shifts in wording, format, or language can trigger surprisingly large failures, especially on multi-step reasoning problems. To address this problem, we propose a **Distributionally Robust Token Optimization (DRTO)** approach, which combines token-level Reinforcement Learning from Human Feedback (RLHF) with *Distributionally Robust Optimization (DRO)*. DRTO constructs f-divergence ambiguity sets over span-level actor losses, providing a principled way to emphasize difficult response segments during policy optimization. Empirically, DRTO enhances consistency under natural distribution shifts in multiple reasoning benchmarks among different tasks, achieving accuracy of $+4.4$ percentage points on `MATH-500` and $+2.7$ percentage points on `LiveCodeBench` over standard RTO. Our code is available at `https://github.com/noc-lab/DRTO`.

1 Introduction

Large language models (LLMs) have achieved strong performance on a wide range of language and reasoning tasks, and reinforcement learning from human feedback (RLHF) has become a practical and widely used approach to align these models with human preferences. In modern post-training pipelines, two mainstream choices are Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2023). Despite their success, both methods can be vulnerable to distributional shifts between the preference data used for alignment and the prompts or evaluation conditions encountered at deployment. In practice, such vulnerability may appear as training instability and reduced robustness. Small changes in phrasing, notation, or other superficial input variations can lead to substantial drops in accuracy.

Due to this instability issue, recent work has begun to modify PPO and DPO models to enhance robustness (Wu et al., 2025; Huang et al., 2025). However, existing robust variants are often limited by increased conservatism or additional computational overhead, and they usually do not fully exploit the token-level credit assignment. As a result, they may struggle to improve out-of-distribution robustness without sacrificing strong in-distribution performance, especially on reasoning-intensive tasks.

To address these limitations, we develop a distributionally robust method, building on a recent token-level alignment framework, *Reinforced Token Optimization* (RTO) (Zhong et al., 2025), which already demonstrates consistent advantages over PPO and DPO by combining PPO-style policy optimization

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

with a token-wise reward signal learned from preferences. We propose *Distributionally Robust Token Optimization* (DRTO), which adapts *Distributionally Robust Optimization* (DRO) to the RTO actor update through span-level divergence-based ambiguity sets, based on KL and χ^2 divergence metrics. An overview of the resulting pipeline is shown in Figure 2. Such a formulation improves consistency on out-of-distribution evaluations, while retaining RTO's performance advantages over DPO and PPO in our setting.

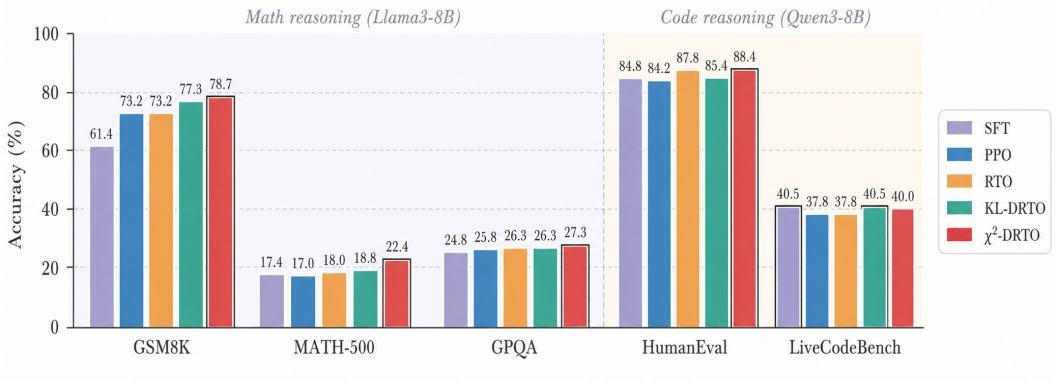


Figure 1: Performance of five RLHF fine-tuning methods on math and coding benchmarks under distribution shift. DRTO improves over standard RTO on most benchmarks.

1.1 Contributions

This work enhances the robustness of RLHF via a span-level DRO approach. We summarize our main contributions as follows.

- **Distributionally robust span-level objectives.** We introduce ambiguity sets over response spans from current RTO rollouts and derive robust objectives for the actor loss. Each response is partitioned into contiguous spans, where each span is assigned a mean PPO surrogate loss, and DRO is applied directly over the resulting span losses. To the best of our knowledge, we are not aware of prior work that applies DRO over response segmentations within an RLHF framework.

- **Realization of span DRO.** We use two distinct f-divergences to define the ambiguity set, leading to two different practical realizations. For KL-DRTO, we obtain an entropic objective that smoothly emphasizes high-loss spans through exponential tilting; for χ^2-DRTO, we obtain a tractable surrogate based on the response-balanced span mean and standard deviation.

- **Empirical improvements under distribution shifts.** Our practical implementations of KL-DRTO and χ^2-DRTO use the same training pipeline as standard RTO, with modest additional training time. Empirically, both methods yield more consistent performance under linguistic and symbolic shifts on math reasoning tasks, while preserving strong in-domain performance. As summarized in Figure 1, across the five math benchmarks DRTO attains up to $+5.5$ percentage points absolute accuracy improvement over standard RTO, including $+4.4$ on `MATH-500`, and both variants match or exceed RTO on all five math benchmarks in the reported runs; on coding, χ^2-DRTO matches or exceeds RTO on every benchmark (including $+2.12$ percentage points on `LiveCodeBench`), while KL-DRTO obtains the larger DRTO gain on `LiveCodeBench`, at $+2.70$ percentage points. Subsequent ablation studies further validate DRTO's efficiency.

In summary, we develop a theoretically grounded and practically effective approach for robust token-level RLHF, which also inherits the reasoning advantages of token-level methods over standard PPO and DPO.

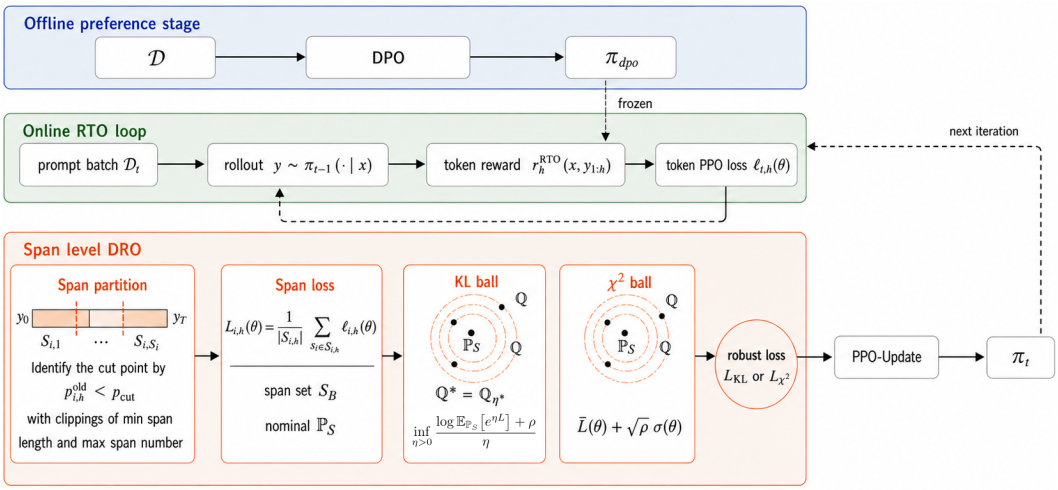


Figure 2: Overview of the DRTO pipeline. Each rollout is partitioned into spans at low-confidence cut points (where $p_{i,h}^{\mathrm{old}} < p_{\mathrm{cut}}$); per-token PPO losses driven by the RTO token reward are aggregated into span losses, which are then combined through KL- or χ^2-divergence DRO to form the final actor loss.

1.2 Related work

Token-level RLHF. RLHF (Ouyang et al., 2022; Bai et al., 2022) is widely used to align LLMs with human preferences. Among all popular frameworks, PPO (Schulman et al., 2017) remains a common choice for reinforcement-based fine-tuning, typically with a KL control term to limit deviation from a reference policy. As an alternative mainstream approach, DPO (Rafailov et al., 2023) reformulates preference optimization as a supervised objective on pairwise comparisons, avoiding explicit on-policy rollouts. To enhance the two popular methods with sequence-level objectives, token-level alignment methods aim to improve credit assignment and make training less brittle. For instance, Token-level DPO (TDPO) (Zeng et al., 2024) introduces a token-wise formulation of the preference objective. In this paper, we build on Reinforced Token Optimization (RTO) (Zhong et al., 2025), which augments PPO with a token-wise reward derived from a DPO-trained policy, enabling more localized learning signals. Even so, token-level training can be sensitive to reward noise, difficult response segments, and minibatch composition, which motivates robust objectives that enhance stability under such uncertainty.

Distributionally robust optimization for preference alignment. DRO provides a principled framework for learning under distributional uncertainty (Chen and Paschalidis, 2020; Rahimian and Mehrotra, 2022; Duchi et al., 2021), and divergence-based ambiguity sets yield tractable dual forms linked to variance-sensitive and entropic-risk objectives (Namkoong and Duchi, 2017). Recent alignment work has applied DRO to offline preference optimization: DrDPO (Wu et al., 2025) robustifies DPO against data noise via distributionally robust reweighting, and χ^2-Preference Optimization (Huang et al., 2025) addresses offline over-optimization via χ^2-based regularization. Group-robust preference optimization (GRPO) (Ramesh et al., 2024) maximizes worst-group performance when group labels are available. In contrast, DRTO enhances the robustness of the on-policy RTO actor loss by applying KL or χ^2 ambiguity sets over span-level PPO losses, while also inheriting RTO's advantages over DPO and PPO under in-distribution settings.

Segment-level RL for LLMs. A growing line of work segments responses for finer-grained learning signals: Yin et al. (2025) and Li et al. (2024) learn segment-level rewards via dynamic or semantic boundaries. Our span construction shares the segment-level granularity but is used for divergence-based DRO reweighting rather than reward shaping or advantage smoothing.

Robustness and safety beyond DRO. A complementary line of work targets adversarial robustness and safety. White-box methods such as GradSafe (Xie et al., 2024), Gradient Cuff (Hu et al., 2024), and Robust Prompt Optimization (Zhou et al., 2024) exploit internal model signals or gradient-based suffix optimization, while black-box defenses such as SmoothLLM (Robey et al., 2025) and

CCFC (Hu et al., 2025) aggregate over input perturbations or dual-track prompts. These approaches are orthogonal to DRTO: they target safety against adversarial prompts, whereas DRTO targets consistent performance under natural distribution shift.

2 Preliminaries

In this section, we introduce the standard token-level optimization framework for RLHF and establish the notation used throughout the paper.

2.1 Proximal Policy Optimization

The RLHF objective maximizes a learned reward $r(x, y)$ while keeping the policy π_θ close to a reference π_{ref} via a KL penalty (Ouyang et al., 2022). PPO optimizes this objective with clipped on-policy updates. Concretely, we sample a minibatch of n rollouts $\mathcal{B} = \{\tau_i = (x_i, y_i)\}_{i=1}^n$, where each x_i is an input prompt and $y_i = (y_{i,1}, \ldots, y_{i,H_i})$ is the response of length H_i generated by the rollout policy π_{old} (the snapshot of π_θ used to collect the current minibatch, refreshed at each outer iteration), with each token $y_{i,h}$ drawn from the model's vocabulary. At answer position $h \in [H_i]$, let $s_{i,h} := (x_i, y_{i,1:h-1})$ denote the partial context (prompt and tokens generated so far) and $a_{i,h} := y_{i,h}$ the next-token action; the PPO importance-sampling ratio is then $\rho_{i,h}(\theta) = \pi_\theta(a_{i,h} \mid s_{i,h})/\pi_{\text{old}}(a_{i,h} \mid s_{i,h})$, where π_{old} is the rollout policy and is distinct from the frozen reference π_{ref} used only in the KL regularization term. The per-token advantage $\widehat{A}_{i,h} \in \mathbb{R}$ estimates how much better action $a_{i,h}$ is than the policy's average behavior in state $s_{i,h}$, computed from the per-token rewards using generalized advantage estimation (GAE) (Schulman et al., 2016); the clip function $\text{clip}(u, 1-\epsilon, 1+\epsilon)$ truncates u to the interval $[1-\epsilon, 1+\epsilon]$. The PPO clipped surrogate then defines the token-level actor loss

$$\ell_{i,h}(\theta) := -\min\Big(\rho_{i,h}(\theta)\widehat{A}_{i,h}, \text{clip}\big(\rho_{i,h}(\theta), 1-\epsilon, 1+\epsilon\big)\widehat{A}_{i,h}\Big), \tag{1}$$

where $\epsilon > 0$ is the clipping parameter. The response-level and minibatch PPO losses are

$$L_i^{\text{resp}}(\theta) := \tfrac{1}{H_i} \sum_{h \in [H_i]} \ell_{i,h}(\theta), \qquad L_{\text{PPO}}(\theta) = \tfrac{1}{n} \sum_{i=1}^n L_i^{\text{resp}}(\theta). \tag{2}$$

We use `PPO-Update` to denote one minimization step on this loss.

2.2 Reinforced Token Optimization

RTO (Zhong et al., 2025) keeps the PPO actor update but replaces the sparse terminal reward with a dense token-wise shaping reward derived from a preference policy π_{dpo} trained from offline preference data. Starting from π_{ref}, at answer-token position $h \in [H_i]$, the shaping reward is

$$\begin{aligned} r_h^{\text{RTO}}(x, y_{1:h}) = \beta_1 \log \frac{\pi_{\text{dpo}}(y_h \mid x, y_{1:h-1})}{\pi_{\text{ref}}(y_h \mid x, y_{1:h-1})} \\ - \beta_2 \log \frac{\pi_\theta(y_h \mid x, y_{1:h-1})}{\pi_{\text{ref}}(y_h \mid x, y_{1:h-1})} + \beta_3 \mathbf{1}_{\{h=H_i\}}\, r(x, y_{1:H_i}), \end{aligned} \tag{3}$$

where $\beta_1, \beta_2, \beta_3 > 0$ are tuning parameters and the last term is a terminal reward applied only at $h = H_i$. Algorithm 1 formulates the procedure: at each iteration, the DPO model is used to compute token-wise rewards via (3), and a PPO update follows.

3 Span-level Distributionally Robust Token Optimization

We now formulate a span-level DRO objective that models distribution shift via adversarial reweighting over response segments. Span-level reweighting can be viewed as a fine-grained generalization of trajectory-level reweighting: it can up-weight difficult segments inside each response while still aggregating to a valid actor loss. Throughout this section we work with the same on-policy minibatch $\mathcal{B} = \{\tau_i = (x_i, y_i)\}_{i=1}^n$ of n prompt-response rollouts as in Section 2, sampled at each PPO-style iteration; the adversarial reweighting is applied within $\mathcal{B}$ at every update.

Algorithm 1 Standard RTO

Input: offline dataset $\mathcal{D}$; $\beta_1, \beta_2, \beta_3 > 0$; DPO algorithm `DPO`; PPO trainer `PPO-Update`.
Train $\pi_{\mathrm{dpo}} \leftarrow \texttt{DPO}(\mathcal{D})$; initialize $\pi_0 = \pi_{\mathrm{ref}}$.
for $t = 1, \ldots, T$ **do**
  Sample a batch of prompts D_t.
  For each $x \in D_t$, generate response $y \sim \pi_{t-1}(\cdot \mid x)$.
  Compute token-wise rewards $\{r_h^{\mathrm{RTO}}(x, y_{1:h})\}_{h \in [H]}$ by (3).
  Update $\pi_t \leftarrow \texttt{PPO-Update}(\pi_{t-1}, r^{\mathrm{RTO}}, D_t)$.
end for
Output: π_T.

3.1 Span-level DRO objective

Span partition. For the i-th response, we partition the answer-token positions $[H_i]$ into S_i contiguous spans

$$\mathcal{G}_{i,1}, \ldots, \mathcal{G}_{i,S_i}, \qquad \mathcal{G}_{i,s} \subseteq [H_i].$$

We place span cut points at low-confidence rollout positions, i.e., positions whose stored probability $p_{i,h}^{\mathrm{old}}$ falls below a cut probability threshold p_{cut}, subject to a minimum span length and a maximum number of spans per response. Low-probability tokens often mark transitions between reasoning steps or syntactic units, so they form natural span boundaries; the full procedure is deferred to Appendix C. We let

$$\mathcal{S}_{\mathcal{B}} := \{(i, s) : i \in [n],\ s \in [S_i]\}$$

index all spans in the minibatch $\mathcal{B}$.

Span loss. Reusing the per-token PPO loss $\ell_{i,h}(\theta)$ in (1), the loss of span $\mathcal{G}_{i,s}$ is its mean token loss

$$L_{i,s}(\theta) := \frac{1}{|\mathcal{G}_{i,s}|} \sum_{h \in \mathcal{G}_{i,s}} \ell_{i,h}(\theta), \tag{4}$$

which recovers the response-level loss (2) when $S_i = 1$.

Span-level DRO. We assign each response equal mass $1/n$ and split it uniformly across its spans, defining the response-balanced nominal span distribution

$$\mathbb{P}_{\mathcal{S}}(i, s) := \frac{1}{nS_i}, \qquad (i, s) \in \mathcal{S}_{\mathcal{B}}. \tag{5}$$

Equal nominal weighting within each response treats a span as one local loss unit, regardless of its token count. For radius $\rho \geq 0$, let $\Omega_\rho(\mathbb{P}_{\mathcal{S}}) := \{\mathbb{Q} : D_f(\mathbb{Q} \| \mathbb{P}_{\mathcal{S}}) \leq \rho\}$ be an ambiguity set of distributions over the observed spans. DRTO replaces the mean PPO loss with the robust value

$$R(\theta; \rho) := \sup_{\mathbb{Q} \in \Omega_\rho(\mathbb{P}_{\mathcal{S}})} \mathbb{E}_{(i,s) \sim \mathbb{Q}}\big[L_{i,s}(\theta)\big], \tag{6}$$

which reduces to $L_{\mathrm{PPO}}(\theta)$ when $\rho = 0$ and $S_i \equiv 1$. The ambiguity set models reweightings of the observed span losses; we evaluate whether this local robustness transfers to natural distribution shifts. The following sections instantiate (6) using KL and Pearson χ^2 divergences.

3.2 DRTO on a KL-divergence ambiguity set

We first instantiate (6) with a KL-divergence ambiguity set

$$\Omega_\rho^{\mathrm{KL}}(\mathbb{P}_{\mathcal{S}}) = \{\mathbb{Q} :\ D_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_{\mathcal{S}}) \leq \rho\}. \tag{7}$$

On the finite span set, the inner maximization in (6) admits an exact one-dimensional dual Hu and Hong (2012). For any $\eta > 0$, we define the entropic risk

$$\Psi_\eta(\theta) := \frac{1}{\eta} \log \mathbb{E}_{(i,s) \sim \mathbb{P}_{\mathcal{S}}}[\exp(\eta L_{i,s}(\theta))], \tag{8}$$

Algorithm 2 KL-DRTO

Input: $\mathcal{D}$; $\beta_{1:3}, \rho > 0$; `DPO`; `PPO-Update`.
Train $\pi_{\text{dpo}} \leftarrow \texttt{DPO}(\mathcal{D})$; init $\pi_0 = \pi_{\text{ref}}$.
for $t = 1, \ldots, T$ **do**
  Sample D_t and generate $y \sim \pi_{t-1}(\cdot|x)$.
  Compute $\{r_h^{\text{RTO}}\}$ by (3).
  Build spans $\mathcal{S}_{\mathcal{B}}$ and $\{L_{i,s}\}$ via (4).
  Solve the dual in (10) for its optimal span weights.
  For a finite $\eta^\star$, $\mathbb{Q}^\star = \mathbb{Q}_{\eta^\star}$ via (9).
  Set $L_{\text{KL}}(\theta) = \sum_{(i,s)} \mathbb{Q}^\star(i,s) L_{i,s}(\theta)$.
  $\pi_t \leftarrow \texttt{PPO-Update}(\pi_{t-1}, r^{\text{RTO}}, L_{\text{KL}})$.
end for
Output: π_T.

Algorithm 3 χ^2-DRTO

Input: $\mathcal{D}$; $\beta_{1:3} > 0$, $\rho, \delta > 0$; `DPO`; `PPO-Update`.
Train $\pi_{\text{dpo}} \leftarrow \texttt{DPO}(\mathcal{D})$; init $\pi_0 = \pi_{\text{ref}}$.
for $t = 1, \ldots, T$ **do**
  Sample D_t and generate $y \sim \pi_{t-1}(\cdot|x)$.
  Compute $\{r_h^{\text{RTO}}\}$ by (3).
  Build spans $\mathcal{S}_{\mathcal{B}}$ and $\{L_{i,s}\}$ via (4).
  $\bar{L} = \mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[L_{i,s}]$, $\text{Var}(L) = \mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[(L_{i,s} - \bar{L})^2]$.
  Set $L_{\chi^2}(\theta) = \bar{L} + \sqrt{\rho}\sqrt{\text{Var}(L) + \delta}$.
  $\pi_t \leftarrow \texttt{PPO-Update}(\pi_{t-1}, r^{\text{RTO}}, L_{\chi^2})$.
end for
Output: π_T.

and the corresponding exponential-tilt weights

$$\mathbb{Q}_\eta(i,s) := \frac{\mathbb{P}_{\mathcal{S}}(i,s)\exp(\eta L_{i,s}(\theta))}{\sum_{(j,r)\in\mathcal{S}_{\mathcal{B}}} \mathbb{P}_{\mathcal{S}}(j,r)\exp(\eta L_{j,r}(\theta))}. \tag{9}$$

Theorem 3.1 (Guarantee of KL-DRTO)**.** *Fix any minibatch $\mathcal{B}$ with span set $\mathcal{S}_{\mathcal{B}}$, and assume $L_{i,s}(\theta)$ is finite for all $(i,s) \in \mathcal{S}_{\mathcal{B}}$. Then the KL-robust value* (6) *admits the exact dual representation*

$$R_{\text{KL}}(\theta;\rho) = \inf_{\eta>0} \left\{ \Psi_\eta(\theta) + \frac{\rho}{\eta} \right\}. \tag{10}$$

Moreover, if the infimum in (10) *is attained at some $\eta^\star \in (0,\infty)$, then the worst-case reweighting is $\mathbb{Q}^\star = \mathbb{Q}_{\eta^\star}$, and*

$$R_{\text{KL}}(\theta;\rho) = \sum_{(i,s)\in\mathcal{S}_{\mathcal{B}}} \mathbb{Q}^\star(i,s)\, L_{i,s}(\theta). \tag{11}$$

Proof. See Appendix A.1. □

Theorem 3.1 reduces the KL-ball adversary in (6) to a scalar minimization over $\eta > 0$ and yields an explicit worst-case reweighting over spans. Algorithm 2 therefore implements KL-DRTO by computing $\eta^\star$ and $\mathbb{Q}^\star$ from the minibatch span losses $\{L_{i,s}(\theta)\}$ and running RTO with the weighted loss $L_{\text{KL}}(\theta) = \sum_{(i,s)} \mathbb{Q}^\star(i,s)\, L_{i,s}(\theta)$.

3.3 DRTO on a χ^2-divergence ambiguity set

We now instantiate (6) with the Pearson χ^2-divergence Namkoong and Duchi (2017)

$$\Omega_\rho^{\chi^2}(\mathbb{P}_{\mathcal{S}}) = \left\{ \mathbb{Q} :\ \mathbb{E}_{(i,s)\sim\mathbb{P}_{\mathcal{S}}} \left[\left(\frac{\mathbb{Q}(i,s)}{\mathbb{P}_{\mathcal{S}}(i,s)} - 1 \right)^2 \right] \le \rho \right\}, \tag{12}$$

and the corresponding χ^2 span-robust value is

$$R_{\chi^2}(\theta;\rho) := \sup_{\mathbb{Q}\in\Omega_\rho^{\chi^2}(\mathbb{P}_{\mathcal{S}})} \mathbb{E}_{(i,s)\sim\mathbb{Q}}[L_{i,s}(\theta)]\,. \tag{13}$$

A tractable relaxation depends only on the span mean and standard deviation under $\mathbb{P}_{\mathcal{S}}$,

$$\bar{L}(\theta) := \mathbb{E}_{(i,s)\sim\mathbb{P}_{\mathcal{S}}}[L_{i,s}(\theta)], \qquad \sigma(\theta) := \sqrt{\text{Var}_{(i,s)\sim\mathbb{P}_{\mathcal{S}}}(L_{i,s}(\theta))}.$$

We thus have the following result.

Table 1: Accuracy of `Llama3-8B` across five benchmarks. Superscripts report absolute change (percentage points) compared to standard RTO. Best results are in **bold**.

Models	**Accuracy**				
	GSM8K	**GSM (CoT)**	**GSM (ES)**	**MATH-500**	**GPQA**
RTO	73.2	74.4	66.0	18.0	26.3
SFT	61.4$^{-11.8}$	64.8$^{-9.6}$	63.6$^{-2.4}$	17.4$^{-0.6}$	24.8$^{-1.5}$
PPO	73.2$^{0.0}$	75.2$^{+0.8}$	65.6$^{-0.4}$	17.0$^{-1.0}$	25.8$^{-0.5}$
DPO	73.9$^{+0.7}$	76.4$^{+2.0}$	66.4$^{+0.4}$	20.2$^{+2.2}$	20.7$^{-5.6}$
FlowRL	75.9$^{+2.7}$	76.2$^{+1.8}$	66.0$^{0.0}$	19.2$^{+1.2}$	**28.8**$^{+2.5}$
DAPO	78.0$^{+4.8}$	76.8$^{+2.4}$	64.0$^{-2.0}$	19.4$^{+1.4}$	25.3$^{-1.0}$
χ^2-DRTO	**78.7**$^{+5.5}$	**79.5**$^{+5.1}$	66.8$^{+0.8}$	**22.4**$^{+4.4}$	27.3$^{+1.0}$
KL-DRTO	77.3$^{+4.1}$	77.4$^{+3.0}$	**67.6**$^{+1.6}$	18.8$^{+0.8}$	26.3$^{0.0}$

Theorem 3.2 (Relaxation for χ^2-DRTO)**.** *Fix any minibatch $\mathcal{B}$ with span set $\mathcal{S}_\mathcal{B}$ and nominal distribution $\mathbb{P}_\mathcal{S}$, and assume $L_{i,s}(\theta) \in \mathbb{R}$ for all $(i,s) \in \mathcal{S}_\mathcal{B}$. Then for any θ,*

$$R_{\chi^2}(\theta;\rho) \;\le\; \bar{L}(\theta) + \sigma(\theta)\sqrt{\rho}. \tag{14}$$

Moreover, if $\sigma(\theta) > 0$ and the nonnegativity condition

$$1 + \sqrt{\rho}\,\frac{L_{i,s}(\theta) - \bar{L}(\theta)}{\sigma(\theta)} \ge 0, \qquad \forall (i,s) \in \mathcal{S}_\mathcal{B}, \tag{15}$$

holds, then the bound (14) *is tight.*

Proof. See Appendix A.2. □

The relaxation in (14) yields the stabilized span-level surrogate

$$L_{\chi^2}(\theta) := \bar{L}(\theta) + \sqrt{\rho}\,\sqrt{\mathrm{Var}_{\mathbb{P}_\mathcal{S}}(L_{i,s}(\theta)) + \delta}, \tag{16}$$

where $\delta > 0$ is a small numerical stabilizer. As $\delta \to 0$, (16) matches the bound in (14), which is tight whenever (15) holds. Algorithm 3 implements χ^2-DRTO by computing (16) from the span losses' weighted mean and variance under $\mathbb{P}_\mathcal{S}$.

4 Experimental results

In this section, we evaluate DRTO under natural distribution shifts and on in-distribution data. We use one generated response per prompt and measure math accuracy and code execution, while training uses the preference-derived token reward in (3).

4.1 Robustness of DRTO

We evaluate KL-DRTO and χ^2-DRTO against several widely used RLHF fine-tuning objectives on two complementary reasoning domains: mathematical reasoning with `Llama3-8B` and code reasoning with `Qwen3-8B`. Training configurations are provided in Appendix C.

Math reasoning setup. The math pipeline uses the `Llama3-8B` backbone. Its online RL stage uses 10,000 samples from `OpenMathInstruct` (Toshniwal et al., 2024), a dataset spanning diverse math topics and solution formats. We report 5-shot flexible-match accuracy via `lm-evaluation-harness` (Gao et al., 2024). The five math benchmarks jointly probe in-distribution generalization and a range of distribution shifts: `GSM8K` (grade-school arithmetic, nearest in-distribution), `GSM(CoT)` (same problems with a chain-of-thought trigger, prompt-format shift), `GSM(ES)` (Spanish translation, language shift), `MATH-500` (harder competition-style problems), and `GPQA` (graduate-level science, difficulty and topic shift).

Table 2: Coding accuracy of `Qwen3-8B` (greedy pass@1). SFT denotes the released `Qwen3-8B` reference without additional fine-tuning in this study. Best results in **bold**.

Models	HumanEval	MBPP	LiveCodeBench
RTO	87.80	60.60	37.84
SFT	$84.76^{-3.04}$	$69.40^{+8.80}$	$40.54^{+2.70}$
PPO	$84.15^{-3.65}$	$68.60^{+8.00}$	$37.84^{0.00}$
DPO	$78.05^{-9.75}$	$\mathbf{70.60}^{+10.00}$	$24.13^{-13.71}$
FlowRL	$84.76^{-3.04}$	$59.60^{-1.00}$	$39.96^{+2.12}$
DAPO	$85.98^{-1.82}$	$62.60^{+2.00}$	$\mathbf{40.93}^{+3.09}$
χ^2-DRTO	$\mathbf{88.41}^{+0.61}$	$69.00^{+8.40}$	$39.96^{+2.12}$
KL-DRTO	$85.37^{-2.43}$	$70.20^{+9.60}$	$40.54^{+2.70}$

Table 3: Relative training time on the math pipeline (RTO: 10 h).

Method	vs. RTO
RTO	1.00×
χ^2-DRTO	1.16×
KL-DRTO	1.17×
DPO	0.21×
PPO	0.70×
DAPO	3.59×
FlowRL	4.86×

Code reasoning setup. For code reasoning, all methods share the `Qwen3-8B` backbone. We use 5,000 prompts from `UltraInteract` (Yuan et al., 2025) for the RL stage, covering diverse programming problems and solution styles. We initialize the RTO and DRTO actors from a DPO-tuned policy and use a smaller `Qwen3-1.7B` reward model. We evaluate greedy pass@1 on `HumanEval` (Chen et al., 2021), `MBPP` (Austin et al., 2021), and `LiveCodeBench` (Jain et al., 2025) via the `OpenCompass` harness (OpenCompass Contributors, 2023), covering function-level synthesis, broader Python programming patterns, and contamination-resistant competitive problems beyond the training distribution.

We compare DRTO with various baseline models, including (i) RTO (Zhong et al., 2025), the token-level RLHF baseline that our methods extend; (ii) supervised fine-tuning (SFT) (Ouyang et al., 2022) on the corresponding math training data, with the released `Qwen3-8B` checkpoint as the code reference; (iii) DPO (Rafailov et al., 2023), a classic preference-based framework; (iv) PPO (Schulman et al., 2017), a classic policy-gradient RLHF baseline; (v) FlowRL (Zhu et al., 2026), a recent flow-matching policy optimization objective; and (vi) DAPO (Yu et al., 2025), a recent decoupled-clipping with dynamic-sampling RL objective. Our main methods are reported at the bottom of Tables 1 and 2.

DRTO improves robustness across both math and code reasoning. Tables 1 and 2 report results. Both DRTO variants match or exceed RTO on all five math benchmarks, covering prompt-format and language shifts as well as broader out-of-distribution transfer. The largest math gains come from χ^2-DRTO, including an improvement of 4.4 percentage points on `MATH-500`. On code reasoning, χ^2-DRTO also improves all three benchmarks over RTO and achieves the best `HumanEval` accuracy, while KL-DRTO is stronger on `MBPP` and `LiveCodeBench` among the two variants. Appendix E.2 reports a three-seed ablation for both DRTO variants.

Runtime comparison. We compare end-to-end training times on the math pipeline in Table 3. Both DRTO variants remain close to RTO in this comparison, at $1.16\times$ for χ^2-DRTO and $1.17\times$ for KL-DRTO, whereas DAPO and FlowRL are several times more expensive. A representative qualitative example is provided in Appendix D.

4.2 Performance analysis

We conduct ablations on the token reward, span granularity, aggregation rule, and hyperparameters. We report training diagnostics in Appendix B and complete parameter sweeps and seed results in Appendix E.

The vital role of token-wise reward: DRTO vs DRPPO. To examine the contribution of the token-wise reward, we apply the same span objectives to PPO actor losses while retaining PPO's trajectory-level reward. We train the resulting KL-DRPPO and χ^2-DRPPO models under the setup in Section 4.1. Table 4 shows that both DRTO variants consistently outperform their DRPPO counterparts, with the largest gap on `MATH-500`. The comparison highlights the complementary roles

Table 4: Comparison of DRPPO, trajectory-level DRTO (no span), and span-level DRTO. Best results in each column are in **bold**.

Model	GSM8K	MATH-500	GPQA
χ^2-DRPPO	77.4	14.2	26.8
KL-DRPPO	77.0	15.8	25.8
χ^2-DRTO (w/o span)	77.7	18.6	25.8
KL-DRTO (w/o span)	76.8	18.6	26.3
χ^2-DRTO	**78.7**	**22.4**	**27.3**
KL-DRTO	77.3	18.8	26.3

Table 5: In-distribution evaluation after training on GSM8K.

Method	GSM8K
SFT	63.3
PPO	78.1
DPO	76.8
RTO	**80.2**
KL-DRTO	80.1
χ^2-DRTO	79.8

Table 6: Additional aggregation and granularity ablations on the math pipeline. The fine-span run uses χ^2 with $\rho = 0.1$, $L_{\min} = 1$, and $S_{\max} = 64$. Fixed high-loss weighting replaces the DRO aggregation.

Configuration	GSM8K	MATH-500	GPQA
RTO	73.2	18.0	26.3
Response-balanced span mean, $\rho = 0$	76.6	18.4	25.8
Fixed high-loss span weighting	77.2	17.2	26.8
Fine spans, $L_{\min} = 1$, $S_{\max} = 64$	77.0	17.8	25.3
KL-DRTO	77.3	18.8	26.3
χ^2-DRTO	**78.7**	**22.4**	**27.3**

of the two components: the token-wise reward provides local learning signals, while span-level DRO adjusts how those signals contribute to the actor update.

Span-level vs. trajectory-level DRTO. A second ablation examines the contribution of span segmentation. We train trajectory-level variants that retain the RTO token reward and the same divergence, but assign one robust weight to each complete response. Table 4 shows that span-level DRTO improves both variants on `GSM8K` and `MATH-500`, while remaining comparable on `GPQA`. A single trajectory weight treats all parts of a response alike. Span-level reweighting instead lets the objective emphasize individual reasoning segments, retaining the local structure used by the token reward.

In-distribution case. To verify that robust objectives do not distort behavior on the training distribution, we additionally train on the `GSM8K` training split and evaluate on its test split. Table 5 shows that both KL-DRTO and χ^2-DRTO closely track RTO and remain ahead of PPO and DPO. DRTO therefore retains strong in-distribution performance while improving robustness under shift.

Robust aggregation and finer spans. We compare DRTO with the response-balanced span mean and a rule that assigns fixed extra weight to spans in the highest loss quantile. Both controls retain the DRTO training setup and change only the aggregation rule. Table 6 shows that both DRTO variants outperform these controls on `MATH-500`, supporting the contribution of the robust objective beyond span averaging or fixed weighting. We also train a finer span configuration. The default span setting performs better on all three benchmarks, supporting our choice to retain local context within each robust unit. Appendix E.3 gives the control setup.

Hyperparameter sensitivity and training seeds. We vary the ambiguity radius, maximum span count, minimum span length, and cut threshold to examine the sensitivity of DRTO. The χ^2 variant retains its `GSM8K` and `MATH-500` gains at both tested positive radii, and the default span cap performs best among the tested caps. We also examine training variability through a three-seed ablation of both DRTO variants. We provide the complete sweeps and seed results in Appendix E.

Choosing the divergence. We use χ^2-DRTO as the default choice for its consistent gains over RTO across the evaluated math and code benchmarks. KL-DRTO offers stronger results on `GSM(ES)`, `MBPP`, and `LiveCodeBench`. We recommend choosing the divergence and radius using held-out validation; the two numerical radii describe different geometries.

5 Conclusion

We introduced *Distributionally Robust Token Optimization* (DRTO), a framework that incorporates distributionally robust optimization into token-level RLHF by constructing divergence-based ambiguity sets around the empirical span distribution. We derive an exact KL dual and a χ^2 upper bound that is tight under the stated condition. The two variants instantiate the same span-level robust optimization principle through different divergence geometries. As a drop-in replacement in standard RLHF pipelines, DRTO improves over standard RTO on most out-of-distribution math and code evaluations with no increase in model size or architectural change.

Limitations. Our evaluation covers two 8B backbones (`Llama3-8B`, `Qwen3-8B`) and single-response settings. Most comparisons use a single training run; Appendix E.2 reports a three-seed ablation for both DRTO variants. The ambiguity set concerns observed span losses; transfer to natural prompt shifts is evaluated empirically. Span settings and radii affect performance. The evaluation does not address adversarial prompts or jailbreak safety, and DRTO shares the rollout cost of on-policy RLHF. We leave larger models, multi-turn interaction, open-ended preference evaluation, and multi-reward RLHF to future work.

Acknowledgments and Disclosure of Funding

The research has been partially supported by the NSF under grants CCF-2200052, ECCS-2317079, DEB-2433726, and ITE-2607506, the DOE under grant DE-AC02-05CH11231, the NIH under grants UM1TR005308 and U19AG068753, and by Boston University. The authors declare no competing interests.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022. URL `https://arxiv.org/abs/2212.08073`.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL `https://web.stanford.edu/~boyd/cvxbook/`.

Ruidi Chen and Ioannis Ch. Paschalidis. Distributionally robust learning. *Foundations and Trends in Optimization*, 4(1–2):1–243, 2020. doi: 10.1561/2400000026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

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*, volume 235 of *Proceedings of Machine Learning Research*, pages 9722–9744. PMLR, 2024. URL https://proceedings.mlr.press/v235/cui24f.html.

John C. Duchi, Peter W. Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 46(3):946–969, 2021. doi: 10.1287/moor.2020.1085.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Zhaolin Hu and L. Jeff Hong. Kullback–Leibler divergence constrained distributionally robust optimization. Technical report, Optimization Online, 2012. URL https://optimization-online.org/2012/11/3677/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In *Advances in Neural Information Processing Systems*, volume 37, pages 126265–126296, 2024. doi: 10.52202/079017-4011. URL https://openreview.net/forum?id=vI1WqFn15v.

Jiaming Hu, Haoyu Wang, Debarghya Mukherjee, and Ioannis Ch. Paschalidis. CCFC: Core & core-full-core dual-track defense for LLM jailbreak protection, 2025. URL https://arxiv.org/abs/2508.14128.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Wenkai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: A Ray-based Easy-to-use, Scalable and High-performance RLHF Framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 656–666, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-demos.48. URL https://aclanthology.org/2025.emnlp-demos.48/.

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D. Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J. Foster. Correcting the mythos of KL-regularization: Direct alignment without overoptimization via χ^2 preference optimization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hXm0Wu2U9K.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Yanshi Li, Shaopan Xiong, Gengru Chen, Xiaoyang Li, Yijia Luo, Xingyuan Bu, Yingshui Tan, Wenbo Su, and Bo Zheng. Adaptive segment-level reward: Bridging the gap between action and reward space in alignment. *arXiv preprint arXiv:2411.00809*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-Reward-V2: Scaling Preference Data Curation via Human-AI Synergy. *arXiv preprint arXiv:2507.01352*, 2025. URL https://arxiv.org/abs/2507.01352.

Llama Team, AI @ Meta. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Hongseok Namkoong and John C. Duchi. Variance-based regularization with convex objectives. In *Advances in Neural Information Processing Systems*, volume 30, pages 2975–2984, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5a142a55461d5fef016acfb927fee0bd-Abstract.html.

OpenCompass Contributors. OpenCompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 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. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, 2022. doi: 10.5802/ojmo.15.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free RLHF. In *Advances in Neural Information Processing Systems*, volume 37, pages 37100–37137, 2024. doi: 10.52202/079017-1171. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4147dfaa46cd7e20a2aecb91097ae8cc-Abstract-Conference.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*, 2025.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

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

Naufal Suryanto, Muzammal Naseer, Pengfei Li, Syed Talal Wasim, Jinhui Yi, Juergen Gall, Paolo Ceravolo, and Ernesto Damiani. RedSage: A Cybersecurity Generalist LLM. In *International Conference on Learning Representations*, 2026.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. OpenMathInstruct-1: A 1.8 million math instruction tuning dataset. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=Mbd3QxXjq5. Datasets and Benchmarks Track.

Ramon van Handel. Probability in high dimension. Lecture notes, Princeton University, 2016. URL https://web.math.princeton.edu/~rvan/APC550.pdf.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CbfsKHiWEn.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. GradSafe: Detecting jailbreak prompts for LLMs via safety-critical gradient analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 507–518, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.30. URL https://aclanthology.org/2024.acl-long.30/.

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. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yueqin Yin, Shentao Yang, Yujia Xie, Ziyi Yang, Yuting Sun, Hany Hassan Awadalla, Weizhu Chen, and Mingyuan Zhou. Segmenting text and learning their rewards for improved RLHF in language model. *Transactions on Machine Learning Research*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. Advancing LLM reasoning generalists with preference trees. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 58348–58365. PMLR, 2024. URL https://proceedings.mlr.press/v235/zeng24c.html.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. DPO meets PPO: Reinforced token optimization for RLHF. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 78498–78521. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhong25b.html.

Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In *Advances in Neural Information Processing Systems*, volume 37, pages 40184–40211, 2024. doi: 10.52202/079017-1270. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/46ed503889ab232c21c1162340ee17b2-Abstract-Conference.html.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, Qizheng Zhang, Lin Chen, Fanghao Shao, Bo Xue, Yunchong Song, Zhenjie Yang, Ganqu Cui, Ning Ding, Jianfeng Gao, Xiaodong Liu, Bowen Zhou, Hongyuan Mei, and Zhouhan Lin. FlowRL: Matching reward distributions for LLM reasoning. In *International Conference on Learning Representations*, 2026.

A Appendix of proofs

The derivations use standard KL variational duality and χ^2 mean–variance bounds (Hu and Hong, 2012; Namkoong and Duchi, 2017) for the span actor objectives in Section 3.

A.1 Proof of Theorem 3.1

We prove the exact one-dimensional dual representation of the span-level KL-robust value and the induced exponential-tilt form of the worst-case span reweighting when the dual infimum is attained at a finite temperature.

Let $m = (i, s)$ index a span in $\mathcal{S}_{\mathcal{B}}$, write $z_m = L_{i,s}(\theta)$, and let $\mathbb{P}_{\mathcal{S}}(m)$ be the response-balanced nominal probability. If $\rho = 0$, then the constraint $D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}_{\mathcal{S}}) \le 0$ forces $\mathbb{Q} = \mathbb{P}_{\mathcal{S}}$, so $R_{\mathrm{KL}}(\theta; 0) = \sum_m \mathbb{P}_{\mathcal{S}}(m)\, z_m$. Moreover, $\Psi_\eta(\theta) \to \sum_m \mathbb{P}_{\mathcal{S}}(m)\, z_m$ as $\eta \downarrow 0$, hence (10) holds when $\rho = 0$. We now assume $\rho > 0$.

The inner maximization in (6) can be written as

$$\max_{\mathbb{Q}\in\Delta(\mathcal{S}_{\mathcal{B}})} \sum_m \mathbb{Q}(m) z_m \quad \text{where} \quad D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}_{\mathcal{S}}) \le \rho.$$

The point $\mathbb{Q} = \mathbb{P}_{\mathcal{S}}$ is strictly feasible whenever $\rho > 0$, so Slater's condition (Boyd and Vandenberghe, 2004) holds and strong duality applies. Let $\nu \ge 0$ be the Lagrange multiplier for the KL constraint. Then

$$R_{\mathrm{KL}}(\theta; \rho) = \inf_{\nu\ge 0} \left\{ \nu\rho + \sup_{\mathbb{Q}\in\Delta(\mathcal{S}_{\mathcal{B}})} \left(\sum_m \mathbb{Q}(m) z_m - \nu D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}_{\mathcal{S}}) \right) \right\}.$$

For $\nu > 0$, set $\eta = 1/\nu$. By the Gibbs variational principle (van Handel, 2016),

$$\sup_{\mathbb{Q}\in\Delta(\mathcal{S}_{\mathcal{B}})} \left(\sum_m \mathbb{Q}(m) z_m - \frac{1}{\eta} D_{\mathrm{KL}}(\mathbb{Q}\|\mathbb{P}_{\mathcal{S}}) \right) = \frac{1}{\eta} \log \sum_m \mathbb{P}_{\mathcal{S}}(m)\, e^{\eta z_m} = \Psi_\eta(\theta),$$

and the maximizer is

$$\mathbb{Q}_\eta(m) = \frac{\mathbb{P}_{\mathcal{S}}(m)\, e^{\eta z_m}}{\sum_{m'} \mathbb{P}_{\mathcal{S}}(m')\, e^{\eta z_{m'}}},$$

which is exactly (9). Substituting $\nu = 1/\eta$ gives (10); the case $\nu = 0$ is recovered as the $\eta \to \infty$ limit. If the infimum is attained at a finite $\eta^\star$, strong duality and the KKT conditions imply that the corresponding primal optimizer is $\mathbb{Q}_{\eta^\star}$, which gives (11).

A.2 Proof of Theorem 3.2

We first show the following bound, which also characterizes an optimizer when the nonnegativity condition holds.

Theorem A.1 (χ^2 span-robust bound)**.** *Fix a minibatch $\mathcal{B}$ and its span set $\mathcal{S}_{\mathcal{B}}$. Let*

$$\bar{L}(\theta) = \mathbb{E}_{(i,s)\sim\mathbb{P}_{\mathcal{S}}}[L_{i,s}(\theta)], \qquad \sigma(\theta) = \sqrt{\mathrm{Var}_{(i,s)\sim\mathbb{P}_{\mathcal{S}}}(L_{i,s}(\theta))}.$$

Then for any $\rho > 0$,

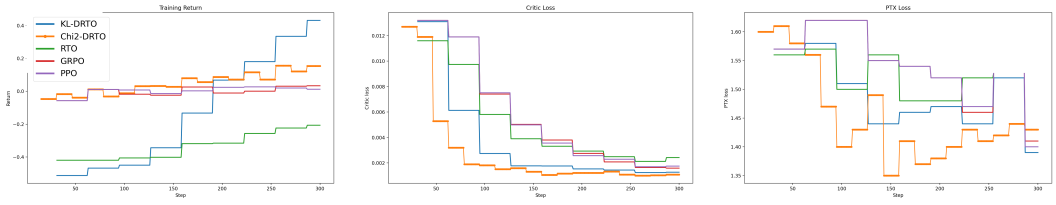
$$R_{\chi^2}(\theta; \rho) \le \bar{L}(\theta) + \sigma(\theta)\sqrt{\rho}. \tag{17}$$

Moreover, given $z_{i,s}(\theta) = L_{i,s}(\theta) - \bar{L}(\theta)$, if $\sigma(\theta) > 0$ and

$$1 + \sqrt{\rho}\, \frac{z_{i,s}(\theta)}{\sigma(\theta)} \ge 0 \quad \textit{for all } (i, s), \tag{18}$$

then the bound in (17) *is tight and the supremum in* (13) *equals $\bar{L}(\theta) + \sigma(\theta)\sqrt{\rho}$. In this case, an optimal adversary is*

$$\mathbb{Q}^\star(i, s) = \mathbb{P}_{\mathcal{S}}(i, s) \left(1 + \sqrt{\rho}\, \frac{z_{i,s}(\theta)}{\sigma(\theta)} \right). \tag{19}$$



(a) Training return comparison. (b) Critic loss during training. (c) PTX loss comparison.

Figure 3: Training dynamics comparison across methods.

Proof. Let $w_{i,s} = \mathbb{Q}(i,s)/\mathbb{P}_{\mathcal{S}}(i,s)$. Then $\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[w] = 1$ and $\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[(w-1)^2] \le \rho$. Adding and subtracting $\bar{L}(\theta)$ inside the inner expectation and using $\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[w] = 1$ gives the mean-shift identity

$$\sum_{(i,s)} \mathbb{Q}(i,s)\, L_{i,s}(\theta) = \mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[w(L_{i,s}(\theta) - \bar{L}(\theta))] + \bar{L}(\theta) = \bar{L}(\theta) + \mathbb{E}_{\mathbb{P}_{\mathcal{S}}}\big[(w-1)(L_{i,s}(\theta) - \bar{L}(\theta))\big],$$

where the second equality uses the fact that $\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[L_{i,s}(\theta) - \bar{L}(\theta)] = 0$, so the contribution of the centered random variable under the constant weight 1 vanishes. By Cauchy–Schwarz,

$$\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}\big[(w-1)(L_{i,s}(\theta) - \bar{L}(\theta))\big] \le \sqrt{\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[(w-1)^2]}\sqrt{\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[(L_{i,s}(\theta) - \bar{L}(\theta))^2]} \le \sqrt{\rho}\,\sigma(\theta),$$

which proves (17).

If $\sigma(\theta) = 0$, then all span losses are equal under $\mathbb{P}_{\mathcal{S}}$, and the theorem is immediate. Now assume $\sigma(\theta) > 0$ and (18) holds. Define $w^\star_{i,s} = 1 + \sqrt{\rho}\, z_{i,s}(\theta)/\sigma(\theta)$ and $\mathbb{Q}^\star(i,s) = \mathbb{P}_{\mathcal{S}}(i,s)\, w^\star_{i,s}$. Then $\mathbb{Q}^\star$ is a valid distribution, $\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[w^\star] = 1$, and

$$\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[(w^\star - 1)^2] = \mathbb{E}_{\mathbb{P}_{\mathcal{S}}}\left[\rho \frac{z_{i,s}(\theta)^2}{\sigma(\theta)^2}\right] = \rho,$$

so $\mathbb{Q}^\star \in \Omega^{\chi^2}_\rho(\mathbb{P}_{\mathcal{S}})$. Finally,

$$\sum_{(i,s)} \mathbb{Q}^\star(i,s)\, L_{i,s}(\theta) = \bar{L}(\theta) + \sqrt{\rho}\frac{\mathbb{E}_{\mathbb{P}_{\mathcal{S}}}[z_{i,s}(\theta)^2]}{\sigma(\theta)} = \bar{L}(\theta) + \sigma(\theta)\sqrt{\rho},$$

which matches the upper bound and proves tightness. □

Theorem 3.2 follows by applying Theorem A.1 to $R_{\chi^2}(\theta;\rho)$.

B Training dynamics

We examine training return, critic loss, and PTX loss in Figure 3.

Convergence and optimization effectiveness. Both DRTO variants show an overall increase in training return in Figure 3a.

Critic stability. Figure 3b reports the critic loss. Both DRTO variants show decreasing critic loss, with lower values than PPO and RTO toward the end of the displayed runs.

Preserving the pretraining objective. Figure 3c reports the auxiliary PTX loss during training. For both DRTO variants, the final PTX loss is below its initial recorded value.

C Implementation details

This appendix provides implementation details for all baselines and robust variants in our DRTO study, including the training pipeline, shared hyperparameters, and method-specific configurations.

Training pipeline. For math experiments, we use `OpenRLHF/Llama-3-8b-sft-mixture`, based on Llama 3 (Llama Team, AI @ Meta, 2024), to initialize the actor. We use `RTO-RL/Llama3.2-1B-RewardModel` as the reward model and critic initialization. For RTO and DRTO, we use `RTO-RL/Llama3-8B-DPO` as the frozen policy π_{dpo} in the token reward.

Datasets. We use UltraFeedback (Cui et al., 2024) to train the math DPO baseline, with the hyperparameters listed below. We use prompt-only examples from OpenMathInstruct-1 (Toshniwal et al., 2024) for online RL (PPO/RTO/DRTO). For the math pipeline, both KL-DRTO and χ^2-DRTO use `max_samples=10000`. We use `seed=42` unless stated otherwise.

Shared RLHF setup. Across PPO/RTO/DRTO runs, we apply a cosine learning rate schedule with 3% warmup steps and 10% minimum learning rate. We use DeepSpeed ZeRO-2 with `adam_offload`, `bf16`, `flash_attn`, and `gradient_checkpointing`.

PPO implementation details. Our PPO-based runs follow the OpenRLHF implementation (J. Hu et al., 2025) and enable reward normalization. We use GAE with `lambd=0.95` and `gamma=1.0`. For DRTO, span cut points are selected from stored rollout probabilities with minimum span length 16, maximum span number 8, and cut probability threshold $p_{\mathrm{cut}} = 0.05$; each span loss is the mean token-level PPO loss in that span, and the nominal span distribution is response-balanced as in (5). The DRO statistics ($\bar{L}$, σ for χ^2-DRTO; $\eta^\star$ and $\mathbb{Q}^\star$ for KL-DRTO) are computed jointly over the spans in the current actor-loss minibatch. For KL-DRTO, we detach the optimized weights $\mathbb{Q}^\star$ during the actor update, giving the robust gradient through the differentiable span losses. For χ^2-DRTO, the gradient of (16) assigns coefficient $\mathbb{P}_{\mathcal{S}}(i,s)\big[1+\sqrt{\rho}(L_{i,s}-\bar{L})/\sqrt{\sigma^2+\delta}\big]$ to each span gradient. The unstabilized coefficients coincide with worst-case probabilities when (15) holds. We list the method-specific hyperparameters below. The aggregation controls use the DRTO configuration.

DPO baseline (UltraFeedback)	
Learning rate	5e-7
Batch size	256
DPO KL coefficient (β)	0.1

RTO (OpenMath)	
Rollout / train batch size	256 / 256
Actor / critic learning rate	1e-5 / 5e-5
Initial KL coefficient	0.01
DPO reward scale / clip	0.05 / 0.05
PTX coefficient	0.05

PPO (OpenMath)	
Rollout / train batch size	512 / 256
Actor / critic learning rate	1e-5 / 5e-5
Initial KL coefficient	0.01
PPO clip (ϵ)	0.2
PTX coefficient	0.05

KL-DRTO (OpenMath)	
Rollout / train batch size	512 / 256
Actor / critic learning rate	1e-5 / 5e-5
Initial KL coefficient	0.005
DPO reward scale / clip	0.05 / 0.05
PTX coefficient	0.05
Span min len / max num	16 / 8
Cut probability p_{cut}	0.05
KL radius ρ	0.01

χ^2-DRTO (OpenMath)	
Rollout / train batch size	512 / 256
Actor / critic learning rate	1e-5 / 5e-5
Initial KL coefficient	0.005
DPO reward scale / clip	0.05 / 0.05
PTX coefficient	0.05
Span min len / max num	16 / 8
Cut probability p_{cut}	0.05
χ^2 radius ρ	0.1

Code pipeline. For coding experiments, we use the released `Qwen3-8B` checkpoint (Yang et al., 2025) for the SFT reference without additional fine-tuning in this study. We initialize the RTO and DRTO actors from `RedSage-Qwen3-8B-DPO` (Suryanto et al., 2026) and use `Skywork-Reward-V2-Qwen3-1.7B` (Liu et al., 2025) as the reward model and critic initialization. We use 5,000 prompt-only examples from `UltraInteract` (Yuan et al., 2025) for the RL stage. We evaluate `HumanEval` (Chen et al., 2021), `MBPP` (Austin et al., 2021; Liu et al., 2023), and `LiveCodeBench` (Jain et al., 2025) with `OpenCompass` (OpenCompass Contributors, 2023) using greedy decoding (pass@1). Table 13 identifies the model and software sources. For the online RL stage, the optimizer, learning rates, rollout and training batch sizes, KL/PTX coefficients, and

Table 7: Example responses on the same prompt. RTO misses the pack price multiplication, while both DRTO variants return the correct total cost.

Question. Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for $5.00. How much does he spend on yogurt over 30 days?

RTO.

```
Terry eats 2 yogurts a day, so over 30 days he eats 2*30=«2*30=60»60 yogurts.
They are currently on sale at 4 yogurts for $5.00, so Terry spends 60/4=$«60/4=15»15.00 on
yogurt over 30 days.
#### 15
```

χ^2-DRTO ($\rho = 0.1$).

```
Terry eats 2 yogurts a day, so over 30 days he eats 2*30=«2*30=60»60 yogurts.
They are currently on sale at 4 yogurts for $5.00, so Terry spends $5.00 for every 4
yogurts.
To find out how much Terry spends on 60 yogurts, we can divide 60 by 4 to find out how
many sets of 4 yogurts he needs:  60/4=«60/4=15»15 sets of 4 yogurts.
Since each set of 4 yogurts costs $5.00, Terry spends 15*$5.00=«15*5=75»$75.00 on yogurt
over 30 days.
#### 75
```

KL-DRTO ($\rho = 0.01$).

```
Terry eats 2 yogurts a day, so over 30 days he eats 2*30=«2*30=60»60 yogurts.
The sale is 4 yogurts for $5.00, so the price per yogurt is $5.00/4 = $«5/4=1.25»1.25.
Therefore, Terry spends 60 * $1.25 = $«60*1.25=75»75 on yogurt over 30 days.
#### 75
```

GAE settings follow the corresponding math configurations above. We use the span objectives and construction rules in Section 3 and the shared precision and memory settings in Table 12.

D Response comparison

To further illustrate qualitative differences among methods, we present a representative evaluation example and compare the responses produced by RTO and both DRTO variants on the same prompt in Table 7.

E Additional experiments

We conduct additional experiments on the `Llama3-8B` math pipeline to examine the ambiguity radius, span construction, aggregation rule, and training seed. We evaluate `GSM8K`, `MATH-500`, and `GPQA` with the same protocol as the main experiments.

E.1 Ambiguity radius and span settings

We vary one parameter at a time and hold the remaining settings fixed. Our defaults are $L_{\min} = 16$, $S_{\max} = 8$, $p_{\text{cut}} = 0.05$, $\rho = 0.1$ for χ^2-DRTO, and $\rho = 0.01$ for KL-DRTO. We use these settings in both pipelines.

Table 8 shows that χ^2-DRTO retains its GSM8K and MATH-500 gains at both tested positive radii. The default span cap gives the strongest results on all three benchmarks among the tested caps. A shorter minimum span length improves MATH-500, while the default gives stronger GSM8K and GPQA results. These differences guide the choice of span settings for each task.

We favor χ^2-DRTO as a practical default across the evaluated math and code benchmarks. KL-DRTO offers stronger MBPP and LiveCodeBench results among the two variants, and we recommend choosing the divergence using held-out validation. The smallest tested KL radius, 0.001, gives the strongest results among the tested KL radii. We interpret the radii separately because the two divergences use different reweighting geometries.

Table 8: Hyperparameter sweeps on `Llama3-8B`. We vary one setting at a time and retain the other defaults. A dagger marks an original default. Scores are accuracies in percent.

Configuration	GSM8K	MATH-500	GPQA
RTO	73.2	18.0	26.3
χ^2 radius ρ			
$\rho = 0$: response-balanced span mean	76.6	18.4	25.8
$\rho = 0.1^\dagger$	78.7	22.4	27.3
$\rho = 0.2$	78.7	22.0	26.3
KL radius ρ			
$\rho = 0.001$	77.7	19.4	26.8
$\rho = 0.01^\dagger$	77.3	18.8	26.3
$\rho = 0.05$	76.6	17.8	26.3
Maximum span count, χ^2 with $\rho = 0.1$			
$S_{\max} = 1$: complete response	77.7	18.6	25.8
$S_{\max} = 8^\dagger$	78.7	22.4	27.3
$S_{\max} = 16$	76.6	18.8	25.3
Minimum span length and cut threshold, χ^2 with $\rho = 0.1$			
$L_{\min} = 8$	76.1	23.4	25.8
$p_{\text{cut}} = 0.2$	76.0	20.0	21.7

Table 9: Training-seed ablation on the math pipeline. We report accuracy (%) as mean $\pm$ sample standard deviation across three training seeds.

Method	GSM8K	MATH-500	GPQA
χ^2-DRTO	77.5 ± 1.0	20.3 ± 2.3	26.6 ± 0.6
KL-DRTO	77.1 ± 0.2	19.0 ± 0.2	26.0 ± 0.3

E.2 Training-seed ablation

We conduct a seed ablation for both DRTO variants using seeds 42, 123, and 456. Table 9 reports mean accuracy and sample standard deviation across the three runs, using the $n - 1$ denominator with $n = 3$.

E.3 Additional aggregation controls

For fixed high-loss weighting, we retain the DRTO training setup and change only the span aggregation rule. We assign extra weight to spans in the highest quantile of detached actor losses, then normalize the weights. We fix the weighting strength without using the test benchmarks.

We can apply both robust objectives directly to token losses. We use spans to average isolated token variation while retaining more local detail than complete responses. We also train a finer-grained configuration with $L_{\min} = 1$ and $S_{\max} = 64$. It retains detected boundaries and a cap on the number of units per response. We report both controls in Table 6.

F Span interpretation and extensions

F.1 Nominal weights and local uncertainty

We assign each response total nominal mass $1/n$ and divide that mass equally among its spans in (5). This choice treats each local region as a unit independently of its token count. Response balancing fixes the nominal distribution; the adversary can redistribute mass across responses as well as spans. An alternative that weights spans by token count is

$$\mathbb{P}_{\mathcal{S}}^{\text{len}}(i, s) := \frac{|\mathcal{G}_{i,s}|}{nH_i}. \tag{20}$$

At zero radius, this alternative gives

$$R^{\mathrm{len}}(\theta;0) = \sum_{(i,s)\in\mathcal{S}_{\mathcal{B}}} \mathbb{P}_{\mathcal{S}}^{\mathrm{len}}(i,s) L_{i,s}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{H_i} \sum_{h\in[H_i]} \ell_{i,h}(\theta) = L_{\mathrm{PPO}}(\theta). \tag{21}$$

Our nominal distribution instead gives the response-balanced mean of span losses at $\rho = 0$, retaining the effect of segmentation.

For a fixed batch and partition, (6) bounds the expected loss under every span distribution in its ambiguity set. This uncertainty class changes the relative masses of observed spans. Changes in format, language, topic, or difficulty may alter the mix of routine and difficult regions in generated responses. We therefore evaluate whether protection against local reweighting transfers to these natural shifts; the formal bound concerns the observed spans.

F.2 Span boundaries and examples

We use the probability of the realized token because it measures the surprise of the sampled action and is already available from rollouts. Predictive entropy measures uncertainty over the full next-token distribution and can be large when several continuations are plausible. These signals can select different boundaries. In a sample of 500 responses from the trained χ^2-DRTO model, the mean realized-token probability is 0.021 at selected cuts and 0.731 across all response tokens. This diagnostic confirms that the cut rule selects low-probability positions.

Minimal example. We consider two responses: A has two spans and B has three. Each response receives total nominal mass $1/2$, giving mass $1/4$ to each span of A and $1/6$ to each span of B. Table 10 shows the two adversaries at the paper radii. Both increase the relative weight of larger losses, with B3 receiving the largest proportional increase over its nominal mass.

Table 10: Nominal and worst-case span masses for the minimal example, with $\rho_{\chi^2} = 0.1$ and $\rho_{\mathrm{KL}} = 0.01$. Masses are rounded to three decimals.

Span	$L_{i,s}$	Nominal $\mathbb{P}_{\mathcal{S}}$	χ^2 $\mathbb{Q}^\star$	KL $\mathbb{Q}^\star$
A1	0.0	0.250	0.169	0.215
A2	1.0	0.250	0.308	0.274
B1	0.0	0.167	0.113	0.143
B2	0.5	0.167	0.159	0.162
B3	1.5	0.167	0.251	0.206

Logged diagnostic minibatch. We also report a minibatch from a small diagnostic run, with $n = 8$ responses and $M = 45$ spans, $\bar{L} = 0.3867$, and $\sigma = 0.8811$, using $L_{\min} = 16$, $S_{\max} = 8$, and $p_{\mathrm{cut}} = 0.05$. The response in Table 11 has two spans, each with nominal mass $1/(8 \times 2) = 0.0625$. We compute both adversaries jointly over all 45 spans, evaluating the KL adversary on the same logged losses for comparison. The displayed global masses therefore need not sum to one within this response. Both spans exceed the batch mean, and Span 2 receives more weight.

Table 11: One response from the logged diagnostic batch, with $\rho_{\chi^2} = 0.1$ and $\rho_{\mathrm{KL}} = 0.01$.

	$L_{i,s}$	$\mathbb{P}_{\mathcal{S}}$	$\mathbb{Q}^\star_{\chi^2}$	$\mathbb{Q}^\star_{\chi^2}/\mathbb{P}_{\mathcal{S}}$	$\mathbb{Q}^\star_{\mathrm{KL}}$	$\mathbb{Q}^\star_{\mathrm{KL}}/\mathbb{P}_{\mathcal{S}}$
Span 1	1.1953	0.0625	0.0806	1.290×	0.0699	1.118×
Span 2	3.0156	0.0625	0.1215	1.944×	0.0918	1.470×

Both examples satisfy (15); the smallest χ^2 multiplier in the logged batch is $0.132 > 0$. We therefore report valid exact adversaries from (19). The stabilized training objective (16) is an upper-bound surrogate. As $\delta \to 0$, it equals the exact robust value when the nonnegativity condition holds, or when $\sigma = 0$. Its gradient coefficients can otherwise be signed, so their interpretation differs from the probabilities in these tables.

F.3 KL boundary cases and computation

At $\rho = 0$, the optimum is $\mathbb{Q}^\star = \mathbb{P}_\mathcal{S}$. Let A contain the spans attaining the largest loss. If $\rho \geq -\log \mathbb{P}_\mathcal{S}(A)$, the conditional distribution $\mathbb{P}_\mathcal{S}(\cdot \mid A)$ is optimal, corresponding to the $\eta \to \infty$ limit in (10). For constant losses, this choice is the nominal distribution. For nonconstant losses and $0 < \rho < -\log \mathbb{P}_\mathcal{S}(A)$, the minimizer is finite. These cases complete the specification of the scalar dual.

With M spans, computing the weighted mean and variance requires $O(M)$ work once span losses are available. Each evaluation of the KL scalar objective and its weights also requires $O(M)$ work; the total depends on the number of solver iterations. The paper settings give $M \leq 8n$. These aggregation costs depend on the span count rather than directly on model size. We report the training resources and recorded computation in Appendix G.1.

F.4 Extensions and evaluation scope

We can extend nominal weighting to conversations by assigning equal mass to each conversation, dividing it equally across turns, and then across spans within each turn. For conversation i with T_i turns and $S_{i,t}$ spans in turn t, this gives

$$\mathbb{P}_{\mathrm{conv}}(i, t, s) := \frac{1}{nT_i S_{i,t}}. \tag{22}$$

An interactive evaluation must also account for how early actions change later states, prompts, and rewards.

We use preference information through the DPO policies and learned reward models described in Appendix C. We evaluate math accuracy and code execution to assess the resulting policies. The same actor objective can use a learned reward for open-ended tasks; that extension should evaluate preference quality together with reward model or judge reliability.

G Resources and asset sources

G.1 Training resources

We use two GPUs per final RLHF training job on nodes with approximately 750 GB of host RAM. We train with DeepSpeed ZeRO-2, optimizer offload, `bf16`, FlashAttention, and gradient checkpointing. We use LoRA (Hu et al., 2022) with rank 32 and $\alpha = 64$. Table 12 summarizes this configuration.

Table 12: Training configuration for the main math and code RLHF runs.

Resource	Setting	Resource	Setting
GPUs per run	2	Parallelism	ZeRO-2
Host RAM	$\sim$ 750 GB	Optimizer offload	Enabled
Precision	`bf16`	LoRA rank / α	32 / 64

The recorded coding RLHF runs selected for comparison take approximately 7 to 14 hours each. Across the collected training logs, we estimate approximately 1.1 thousand GPU-hours for training, including preliminary runs, ablations, additional seeds, and parameter sweeps.

G.2 Model, software, and dataset sources

We list the models and software we use in Table 13 and the datasets in Table 14. The tables identify their sources and stated licenses or release terms.

Table 13: Official sources and stated licenses or release terms for model and software assets.

Asset and source	License or release terms
`OpenRLHF/Llama-3-8b-sft-mixture`	Upstream base: Meta Llama 3 Community License[a]
`RTO-RL/Llama3-8B-DPO` (Zhong et al., 2025)	Upstream base: Meta Llama 3 Community License[a]
`RTO-RL/Llama3.2-1B-RewardModel` (Zhong et al., 2025)	Upstream base: Llama 3.2 Community License[a]
`Qwen/Qwen3-8B`	Apache-2.0
`Skywork/Skywork-Reward-V2-Qwen3-1.7B`	Apache-2.0
`RISys-Lab/RedSage-Qwen3-8B-DPO`	Authors' research-use statement
`OpenRLHF/OpenRLHF` (code)	Apache-2.0
`EleutherAI/lm-evaluation-harness`	MIT
`open-compass/opencompass`	Apache-2.0

[a]These entries identify the licenses of the upstream base models. The OpenRLHF release derives from `RLHFlow/LLaMA3-SFT` and Meta Llama 3; the RTO-RL model cards identify the corresponding Llama 3 and Llama 3.2 bases.

Table 14: Dataset sources and publisher license or release statements.

Dataset / release	Official source	License or source terms
OpenMathInstruct-1	NVIDIA dataset release	NVIDIA License
UltraFeedback (Cui et al., 2024)	Hugging Face H4 binarized release	MIT
UltraInteract (Yuan et al., 2025)	OpenBMB/Eurus project	MIT (dataset)
GSM8K (Cobbe et al., 2021)	OpenAI dataset release	MIT
MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024)	OpenAI PRM800K test split	MIT (MATH data)
GPQA (Rein et al., 2024)	Original dataset release	CC BY 4.0
HumanEval	OpenAI dataset release	MIT
MBPP	Google Research release	CC BY 4.0
LiveCodeBench (code generation)	LiveCodeBench project	Authors' academic-use statement[a]

[a]The LiveCodeBench paper describes the academic use of programming contest material in Appendix A.1. We link that source statement separately from the benchmark's software license.

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: We introduce span-level robust objectives for RTO in Section 3 and evaluate them under natural math and code distribution shifts in Section 4.

   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 the evaluated backbones, tasks, robustness scope, and on-policy training setting in the Limitations paragraph of Section 5.

   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: Theorems 3.1 and 3.2 state the assumptions and tightness conditions. We provide full proofs in Appendices A.1 and A.2.

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 specify the algorithms in Section 3, the datasets and evaluation protocols in Section 4, and the training and implementation settings in Appendix C.

   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: Our implementation and instructions are available at `https://github.com/noc-lab/DRTO`.

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 describe the models, datasets, baselines, and evaluation protocols in Section 4.1. Appendix C reports training settings, and Appendix E.1 reports the defaults and sensitivity experiments.

   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: [Yes]

   Justification: We report mean $\pm$ sample standard deviation across three training seeds for each DRTO variant in Table 9. Appendix E.2 defines the statistic.

   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: We report training resources, coding run times, and estimated total GPU-hours in Appendix G.1, and the math runtime comparison in Table 3.

   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: We use publicly released models and datasets and conduct automated benchmark evaluation.

   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: [No]

    Justification: We focus on the formulation and evaluation of robust RLHF objectives; broader societal impacts are outside the scope of this study.

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: We release research code that uses existing models and datasets. We do not release a new pretrained foundation model or a newly scraped dataset.

    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: We credit the original assets and list their sources and stated licenses or release terms in Tables 13 and 14.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Our code documentation describes environment setup, the DRTO implementation, and training commands.

    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: We use existing datasets and automated benchmark evaluation; the study does not include new crowdsourcing or human-subject experiments.

    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: The study does not include new research with human subjects.

    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: [No]

    Justification: We primarily used LLM for writing and editing in this research.

    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.