Adaptive Robust Estimator for Policy Optimization in Reinforcement Learning

Zhongyi Li[1,*] **Wan Tian**[2,9,*] **Jingyu Chen**[3], **Kangyao Huang**[4]
Huiming Zhang[1,5], **Hui Yang**[6], **Tao Ren**[6], **Ruijie Wang**[7]
Fuzhen Zhuang[1,†] **Yijie Peng**[8,9,10,11,†] **Yikun Ban**[7,†]

[1]School of Artificial Intelligence, Beihang University
[2]Wangxuan Institute of Computer Technology, Peking University
[3]Chinese Academy of Sciences
[4]Tsinghua University
[5]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing
[6]Peking University
[7]School of Computer Science & Engineering, Beihang University
[8]Department of AI for Management,
School of Management and Engineering, Nanjing University
[9]Advanced Institute of Information Technology, Peking University
[10]Hangzhou IndEngine Intelligence Co., Ltd.
[11]Xiangjiang Laboratory

Abstract

Reinforcement learning (RL) has become a key ingredient in post-training large language models (LLMs), with Group Relative Policy Optimization (GRPO) and its variants widely adopted for improving reasoning performance. Despite their success, these methods rely on batch-level reward normalization, where the empirical mean can be severely distorted by noisy, skewed, heavy-tailed, or contaminated rewards. Such distortion directly affects advantage estimation and may lead to unstable or inefficient policy updates. We propose an *Adaptive Robust Estimator* (ARE), a plug-and-play robustification module for GRPO-style policy optimization. ARE replaces the empirical batch mean with a two-level robust location estimator: adaptive loss minimization suppresses extreme rewards within each block, while median-of-means aggregation limits the influence of corrupted blocks. This design preserves the original policy optimization objective while improving the reliability of reward centering. We prove that ARE is consistent and achieves high-probability deviation guarantees under both finite-variance and heavy-tailed reward distributions. Experiments on mathematical reasoning benchmarks show that ARE improves GRPO-based training in both single-agent and multi-agent settings, with particularly clear gains under noisy and out-of-distribution evaluations. We further validate its generality on embodied vision-and-language navigation tasks, where ARE improves training stability and downstream navigation (VLN) performance.

1 Introduction

LLMs have recently achieved substantial progress on complex reasoning tasks, including mathematical problem solving, logical inference, and multi-step decision making [Shao et al., 2024, Lightman et al., 2024, Xu et al., 2025]. These tasks often require models to search over long reasoning trajectories and optimize for task-level correctness rather than merely imitate supervised demonstrations. RL

[*]Equal contribution.
[†]Corresponding authors.

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

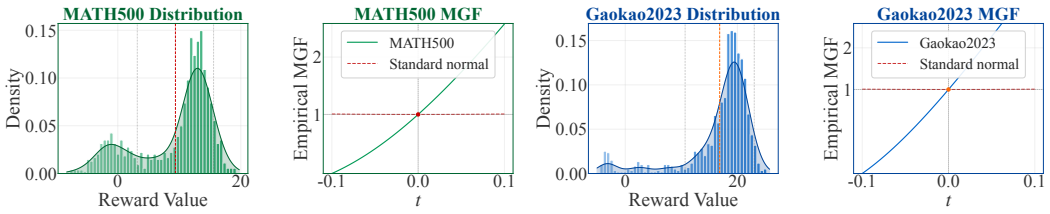


Figure 1: Reward-model scores on responses generated by a large language model for randomly selected problems from MATH500 and Gaokao2023. The results illustrate that reward-model scores can be skewed and non-light-tailed.

therefore plays an increasingly important role in LLM post-training, as it enables optimization with respect to reward signals derived from human preferences, verifiable answers, or learned evaluators [Ouyang et al., 2022, Shao et al., 2024, Lightman et al., 2024, Xu et al., 2025]. This reward-driven formulation allows models to improve reasoning quality, alignment, and task-specific performance beyond supervised fine-tuning.

Following the success of RL in LLM post-training, policy optimization has become a central paradigm for improving reasoning performance. PPO and its variants provide a basic algorithmic foundation for optimizing language models with preference-, outcome-, or verifier-based rewards [Ouyang et al., 2022, Xu et al., 2025]. More recently, GRPO has become widely adopted in reasoning-oriented LLM training, owing to its strong empirical performance and improved memory efficiency over standard PPO-style methods [Shao et al., 2024]. A key feature of GRPO is that it replaces critic-based advantage estimation with group-relative reward normalization, making it particularly suitable for large-scale reasoning tasks. Building on this line of work, related methods such as Dr.GRPO, GSPO and DAPO further improve training efficiency, scalability, and optimization stability in long-chain reasoning settings [Liu et al., 2025, Zheng et al., 2025, Yu et al., 2025]. Despite their algorithmic differences, these methods commonly rely on batch- or group-level reward normalization to construct advantages, making the reliability of reward statistics central to stable policy optimization.

This reliance on empirical reward statistics creates a vulnerability under realistic reward distributions. Such normalization is reliable when rewards are well behaved, but it can become fragile in LLM post-training, where reward signals may be noisy, skewed, heavy-tailed, or contaminated. In particular, the empirical mean used for reward centering is highly sensitive to extreme reward values, and a small number of outlying samples can distort normalized advantages and destabilize policy updates. Recent studies show that reward misspecification and corrupted feedback can severely impair standard RLHF pipelines, and that common regularization strategies such as KL penalties are insufficient to eliminate heavy-tailed failure modes [Kwa et al., 2024, Bukharin et al., 2024]. Related findings in inference-time alignment further suggest that extreme reward realizations can induce overly optimistic or pessimistic decision biases [Hsu et al., 2026]. Empirical reward-model scores in Figure 1 further illustrate such skewed and non-light-tailed behavior on mathematical reasoning benchmarks. These observations highlight the need for robust reward-centering mechanisms that can stabilize advantage estimation under realistic, non-light-tailed reward distributions.

To address these limitations, we propose the *Adaptive Robust Estimator* (ARE), which aims to improve the robustness of reward centering in policy optimization. ARE replaces the empirical mean in batch-level reward normalization with a robust location estimator. Built on the classical MoM framework [Lecué and Lerasle, 2020, Humbert et al., 2022], ARE strengthens standard MoM by replacing each blockwise sample mean with an adaptive-loss-based robust estimate [Barron, 2019, Jung et al., 2024], followed by median aggregation across blocks. This design provides two-level robustness: the adaptive loss suppresses extreme rewards within each block, while median aggregation limits the influence of corrupted blocks. Moreover, the adaptive loss adjusts its shape to the empirical residual distribution, smoothly interpolating between a near-quadratic regime for well-behaved rewards and a heavy-tailed regime for contaminated rewards. Therefore, ARE stabilizes the reward-centering step used in advantage estimation while preserving the original policy optimization objective.

We emphasize that ARE is a plug-and-play robustification module for policy optimization. It can be incorporated into existing GRPO-style training pipelines with minimal modification, without changing the policy optimization objective, model architecture, or task-specific reward function. In multi-agent

settings, ARE can also be applied without modifying the underlying interaction protocol. We further provide theoretical guarantees for ARE, including consistency and high-probability deviation bounds under finite-variance and heavy-tailed reward regimes. We evaluate ARE on mathematical reasoning benchmarks in both single-agent and multi-agent settings, including homogeneous and heterogeneous agent pairs across different model families and scales [Wan et al., 2025]. To further examine its generality beyond language reasoning, we also apply ARE to an aerial VLN task inspired by recent VLM-based navigation systems [Cai et al., 2025, Hong et al., 2025]. The results show that ARE improves the stability and generalization of GRPO-based training, with particularly clear gains under noisy rewards, out-of-distribution benchmarks, and unseen navigation environments.

2 Methodology

2.1 GRPO-based Policy Optimization

We begin with the standard GRPO-style objective used in LLM post-training. Given a batch of trajectories with returns $\{R_i\}_{i=1}^B$, GRPO constructs normalized advantages using batch-level empirical reward statistics: $\mu_B = \frac{1}{B}\sum_{i=1}^B R_i, \sigma_B^2 = \frac{1}{B}\sum_{i=1}^B (R_i - \mu_B)^2$. The normalized advantage is defined as $A_i = \frac{R_i - \mu_B}{\sigma_B + \epsilon_{\text{std}}}$, where $\epsilon_{\text{std}} > 0$ is a small constant for numerical stability. The policy is then optimized through a clipped surrogate objective:

$$\mathcal{L}(\theta) = \mathbb{E}\Big[\min\left(\eta_i(\theta)A_i,\ \text{clip}(\eta_i(\theta), 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}})A_i\right)\Big] - \beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}),$$

where $\eta_i(\theta)$ denotes the importance sampling ratio, $\varepsilon_{\text{clip}}$ is the clipping radius, and β controls the KL regularization strength. This formulation stabilizes policy updates when reward statistics are reliable, but the empirical mean can be strongly affected by heavy-tailed or contaminated rewards, leading to distorted advantage estimates.

2.2 Adaptive Robust Estimator

ARE is a plug-and-play robustification module that replaces the empirical batch mean used for reward centering with a robust location estimate. In the policy-optimization setting, the samples to be robustly centered are the batch rewards or returns; for notational simplicity, we denote them by $X_1, \ldots, X_n$. ARE builds on the classical MoM estimator [Lecué and Lerasle, 2020], which partitions samples into blocks and aggregates blockwise statistics by a median. Standard MoM is robust to a fraction of corrupted blocks, but each blockwise statistic is still a sample mean and can therefore be sensitive to within-block outliers. We further discuss the robustness of the MoM estimator under both random and adversarial contamination in Appendix C. ARE strengthens this framework by introducing robustness at both levels.

Specifically, we partition $X_1, \ldots, X_n$ into K disjoint blocks $B_1, \ldots, B_K$. For each block, ARE computes a robust location estimate by minimizing an adaptive robust loss:

$$\widetilde{X}_j \in \arg\min_{x\in\mathbb{R}} \frac{1}{|B_j|}\sum_{i\in B_j} \rho(X_i - x; \alpha, c)\,, \qquad j = 1, \ldots, K. \tag{1}$$

Here $\rho(\cdot;\alpha,c)$ denotes the adaptive robust loss introduced by Barron [2019], defined as $\rho(e;\alpha,c) = |\alpha-2|/\alpha\left[\left(1+(e/c)^2/|\alpha-2|\right)^{\alpha/2} - 1\right]$, where α controls the shape of the loss and $c > 0$ controls its scale, with standard continuous extensions at special values of α. Smaller values of α induce heavier-tailed and more robust behavior, while $\alpha = 2$ recovers the quadratic loss. We provide a detailed discussion of this adaptive loss family in Appendix E. ARE is then defined as the median of the blockwise robust estimates:

$$\widetilde{\mu} = \text{median}\left(\widetilde{X}_1, \ldots, \widetilde{X}_K\right). \tag{2}$$

This construction provides two complementary forms of robustness: intra-block robustness through adaptive loss minimization, and inter-block robustness through median aggregation. The adaptive loss downweights extreme residuals within each block, while the median limits the influence of corrupted blocks. As a result, ARE provides a stable estimate of the reward center under noisy, heavy-tailed, or contaminated reward distributions.

The main computational challenge lies in solving (1), since the adaptive loss can be nonconvex for robust choices of α. For clarity, consider a single block of size m:

$$\widetilde{X} \in \arg\min_{x\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \rho(X_i - x; \alpha, c)\,. \tag{3}$$

Directly tuning (α, c) by cross-validation or Lepski-type procedures is computationally expensive. We therefore adopt an alternating scheme that estimates the adaptive-loss parameters and the robust location iteratively. Starting from $\widetilde{X}^{(0)}$ and $(\alpha^{(0)}, c^{(0)})$, we update

$$\alpha^{(t+1)}, c^{(t+1)} \in \arg\min_{\alpha\in\mathcal{A},\, c>0} \left\{ \frac{1}{m}\sum_{i=1}^{m} \rho\Big(X_i - \widetilde{X}^{(t)}; \alpha, c\Big) + \log c + \log Z(\alpha) \right\}, \tag{4}$$

$$\widetilde{X}^{(t+1)} \in \arg\min_{x\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \rho\Big(X_i - x; \alpha^{(t+1)}, c^{(t+1)}\Big)\,, \tag{5}$$

until convergence. Here $Z(\alpha)$ is the normalization constant induced by the adaptive-loss density, and $\mathcal{A} \subseteq (-\infty, 2]$ denotes the admissible range of the shape parameter. Section 3 provides efficient solvers for (4) and (5).

2.3 Robust Policy Optimization with ARE

We incorporate ARE into GRPO by replacing the empirical batch mean μ_B with the robust estimate $\widetilde{\mu}_B$ computed from the batch returns: $A_i^{\mathrm{ARE}} = \frac{R_i - \widetilde{\mu}_B}{\sigma_B + \epsilon_{\mathrm{std}}}$. Here we keep the empirical standard deviation σ_B unchanged for a minimal plug-and-play modification, while it can also be replaced by a robust scale estimator when scale contamination is a concern. The resulting optimization objective is

$$\mathcal{L}_{\mathrm{ARE}}(\theta) = \mathbb{E}\Big[\min\left(\eta_i(\theta)A_i^{\mathrm{ARE}},\ \mathrm{clip}(\eta_i(\theta), 1-\varepsilon_{\mathrm{clip}}, 1+\varepsilon_{\mathrm{clip}})A_i^{\mathrm{ARE}}\right)\Big] - \beta\, D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}).$$

Thus, ARE modifies only the reward-centering step used in advantage construction, while preserving the original policy objective, clipping mechanism, and KL regularization. Besides the GRPO algorithm, ARE can also be applied to policy optimization algorithms such as Dr.GRPO and REINFORCE++.

3 Optimization Perspective

In this section, we describe the optimization procedure for the two subproblems in ARE: estimating the adaptive-loss parameters and updating the robust location estimate. Both steps are performed within each block of the MoM partition.

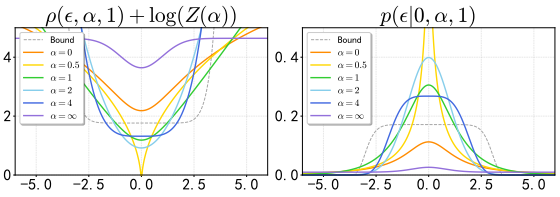


Figure 2: Density and negative log-likelihood induced by the adaptive loss. The left shows the negative log-likelihood $-\log p(\epsilon; \alpha, 1) = \rho(\epsilon; \alpha, 1) + \log Z(\alpha)$ for different shape parameters, and the right shows the corresponding density $p(\epsilon; \alpha, 1) = \exp\{-\rho(\epsilon; \alpha, 1)\}/Z(\alpha)$.

3.1 Estimation of Robust and Scale Parameters

We first examine why directly estimating the adaptive-loss parameters by minimizing the empirical loss can be ill-posed. The following lemma characterizes the monotonicity of the adaptive loss with respect to the shape parameter α.

Lemma 3.1. *Fix $e \in \mathbb{R}$ and $c > 0$. For every $\alpha \neq 0, 2$, $\frac{\partial}{\partial\alpha}\rho(e; \alpha, c) \geq 0$, and the inequality is strict whenever $e \neq 0$.*

Lemma 3.1 shows that, for nonzero residuals, the adaptive loss is monotone increasing in α. Hence, directly minimizing the empirical loss over α tends to drive α toward the smallest admissible value, producing an overly robust and degenerate solution. To avoid this collapse, we follow Barron [2019], Jung et al. [2024] and estimate the adaptive-loss parameters through a likelihood-based calibration.

Specifically, we consider the density $p(\epsilon;\alpha,c)=\frac{\exp\{-\rho(\epsilon;\alpha,c)\}}{cZ(\alpha)}$, $Z(\alpha)=\int_{-\infty}^{\infty}\exp\{-\rho(u;\alpha,1)\}\,du$, where $Z(\alpha)$ is the normalization constant. The corresponding negative log-likelihood is $-\log p(\epsilon;\alpha,c)=\rho(\epsilon;\alpha,c)+\log c+\log Z(\alpha)$. As illustrated in Figure 2, decreasing α flattens the tails of the induced density and reduces the penalty on large residuals, while the normalization term prevents the shape parameter from collapsing to an excessively small value. This likelihood-based formulation therefore balances robustness to outliers with fidelity to the bulk of the residual distribution. Given a current location estimate $\widetilde{X}^{(0)}$, we estimate α and c by minimizing the sample negative log-likelihood:

$$\alpha^{(1)},\, c^{(1)} \in \arg\min_{\alpha\in\mathcal{A},\, c>0}\left\{\sum_{i=1}^{m}\rho\big(X_i-\widetilde{X}^{(0)};\alpha,c\big)+m\big(\log c+\log Z(\alpha)\big)\right\}, \tag{6}$$

where $\mathcal{A}\subseteq(-\infty,2]$ denotes the admissible range of the shape parameter. Unlike prior adaptive robust losses that typically fix or separately estimate the scale parameter, (6) jointly calibrates both α and c, which improves adaptivity but also makes the parameter-estimation step more challenging.

3.2 Estimation of the Mean

After obtaining $\alpha^{(1)}$ and $c^{(1)}$, we update the robust location estimate by solving $\min_{x\in\mathbb{R}}\frac{1}{m}\sum_{i=1}^{m}\rho\big(X_i-x;\alpha^{(1)},c^{(1)}\big)$. This problem is generally nonconvex for robust choices of $\alpha^{(1)}$. We therefore combine the graduated nonconvexity (GNC) paradigm [Black and Rangarajan, 1996] with an iteratively reweighted least-squares (IRLS) update. Let $r_i(x)=X_i-x$. At a fixed iterate $x^{(t)}$, matching the gradient of the adaptive loss leads to a weighted least-squares surrogate:

$$x^{(t+1)}=\arg\min_{x\in\mathbb{R}}\frac{1}{2m}\sum_{i=1}^{m}w_i^{(t)}\big(X_i-x\big)^2, \tag{7}$$

where $w_i^{(t)}=\frac{1}{r}\frac{\partial\rho(r;\alpha^{(1)},c^{(1)})}{\partial r}\Big|_{r=r_i(x^{(t)})}$. The minimizer of (7) has the closed form

$$x^{(t+1)}=\frac{\sum_{i=1}^{m}w_i^{(t)}X_i}{\sum_{i=1}^{m}w_i^{(t)}}. \tag{8}$$

In practice, the weight at $r=0$ is defined by its continuous limit. If the weights are treated as unconstrained free variables and optimized jointly with x, the weighted least-squares objective can degenerate by setting all weights to zero. To prevent this, we introduce an outlier-process regularizer through the Black–Rangarajan duality [Black and Rangarajan, 1996]. Define $\phi(z)=\rho(c\sqrt{z};\alpha,c)$. For robust choices of α, ϕ is concave in z, with $\phi'(z)$ decreasing from its value near the origin to zero in the tails. This property yields an outlier-process representation $\rho(r;\alpha,c)=\min_{\omega\in(0,1]}\left\{\frac{1}{2}\omega\left(\frac{r}{c}\right)^2+\Phi_\rho(\omega)\right\}$, where ω is a normalized weight and Φ_ρ is the corresponding Black–Rangarajan regularizer. Consequently, the robust location update can be written as the regularized problem

$$\widetilde{X}^{(1)}=\arg\min_{x\in\mathbb{R},\,\omega_i\in(0,1]}\frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{2}\omega_i\left(\frac{X_i-x}{c^{(1)}}\right)^2+\Phi_\rho(\omega_i)\right]. \tag{9}$$

Directly optimizing (9) can still be sensitive to initialization because the target loss is nonconvex. We therefore use GNC to start from a convex quadratic surrogate and gradually recover the target adaptive loss. Specifically, we introduce a shape mapping $f(\beta,\alpha^{(1)})$ controlled by a continuation parameter β, such that $f(\beta,\alpha^{(1)})\to 2$ at the beginning of continuation, $f(\beta,\alpha^{(1)})\to\alpha^{(1)}$ at the target loss. At each continuation stage, we solve the corresponding IRLS subproblem using (8), and then update β to increase the degree of nonconvexity. Details of the continuation schedule and stopping criteria are provided in Appendix F.

4 Theoretical Properties

We organize this section as follows. We first study the mean estimator with a fixed shape parameter $\alpha\in(0,1]$ and a sample-size–dependent scale c_m. We then establish two complementary regimes: (i) a finite-variance regime, in which the estimator is consistent and asymptotically normal at

the classical $\sqrt{m}$ rate; and (ii) a heavy-tailed regime, where only a $(1+\epsilon)$-moment is assumed and the estimator attains a high-probability deviation bound of order $\big(\log(1/\delta)/m\big)^{\epsilon/(1+\epsilon)}$ for the single-block estimator in (3). Finally, we study the asymptotic theory of the median-of-means aggregation—namely, the ARE in (2)—based on the collection of block means. Throughout, the same estimating equation underlies all results; the difference between regimes is driven by how c_m is scaled relative to the stochastic fluctuations of the empirical score.

Let $X_1, \dots, X_m$ be i.i.d. real-valued random variables. The goal is to estimate the mean $\mu := \mathbb{E}[X_1]$ (assumed to exist). Fix a shape parameter $\alpha \in (0, 1]$ and let $\kappa_\alpha := 2 - \alpha \in [1, 2)$. We use the adaptive robust loss family $\rho(\varepsilon; \alpha, c) := \frac{\kappa_\alpha}{\alpha}\left[\big(1 + (\varepsilon/c)^2/\kappa_\alpha\big)^{\alpha/2} - 1\right], c > 0$, with score (influence) function $\psi(\varepsilon; \alpha, c) := \frac{\partial}{\partial \varepsilon}\rho(\varepsilon; \alpha, c) = \frac{\varepsilon}{c^2}\big(1 + (\varepsilon/c)^2/\kappa_\alpha\big)^{\alpha/2-1}$. Heuristically, ρ is locally quadratic while ψ downweights large residuals when $|\varepsilon|$ exceeds the scale c. Given a scale $c_m > 0$ (possibly depending on m), define the empirical risk $\widehat{R}_m(x) := \frac{1}{m}\sum_{i=1}^m \rho(X_i - x; \alpha, c_m)$, and the estimating equation $\widehat{g}_m(x) := \frac{d}{dx}\widehat{R}_m(x) = -\frac{1}{m}\sum_{i=1}^m \psi(X_i - x; \alpha, c_m)$.

Let $R_m(x) := \mathbb{E}[\rho(X - x; \alpha, c_m)]$ and $g_m(x) := R_m'(x) = -\mathbb{E}[\psi(X - x; \alpha, c_m)]$. Our proofs compare $\widehat{g}_m$ to its population counterpart g_m on a neighborhood of μ to localize a stable root. For $\alpha < 1$, $\widehat{g}_m(x) = 0$ may have multiple roots due to non-convexity. We therefore define a *central root* in a controlled interval. Let $\widehat{\mu}_{\mathrm{pil}}$ be any pilot estimator and choose a radius $r_m > 0$ such that $r_m \to \infty$ and $r_m = o(c_m)$. Define $\mathcal{I}_m := [\widehat{\mu}_{\mathrm{pil}} - r_m,\ \widehat{\mu}_{\mathrm{pil}} + r_m]$. The only role of $\mathcal{I}_m$ is to rule out spurious distant roots while keeping μ inside with high probability.

Definition 4.1 (Central-root estimator)**.** If the equation $\widehat{g}_m(x) = 0$ has a unique solution in $\mathcal{I}_m$, denote it by $\widetilde{X}_m$.

4.1 Finite-Variance Regime

Assumption 4.2 (Finite variance)**.** $\mathbb{E}[X_1] = \mu$ and $\mathbb{E}[(X_1 - \mu)^2] = \sigma^2 \in (0, \infty)$.

We use a purely asymptotic deterministic tuning: for a fixed $\gamma > 0$, let $c_m := m^{1/2+\gamma}, \quad r_m := m^{\gamma/4}$. With this choice, c_m diverges fast enough that the score is essentially linear on $\mathcal{I}_m$, recovering the classical $\sqrt{m}$ behavior.

Theorem 4.3. *Assume Assumption 4.2, fix $0 < \alpha \le 1$ and $\gamma > 0$, and let $c_m = m^{1/2+\gamma}, r_m = m^{\gamma/4}$. Assume the pilot estimator satisfies*

$$\mathbb{P}\big(|\widehat{\mu}_{\mathrm{pil}} - \mu| \le r_m/2\big) \to 1. \tag{10}$$

Let $\widetilde{X}_m$ be the unique root of $\widehat{g}_m(x) = 0$ in $I_m = [\widehat{\mu}_{\mathrm{pil}} - r_m, \widehat{\mu}_{\mathrm{pil}} + r_m]$, whenever this root exists and is unique. Then there exists an event $\mathscr{E}_m$ with $\mathbb{P}(\mathscr{E}_m) \to 1$ such that, on $\mathscr{E}_m$, the equation $\widehat{g}_m(x) = 0$ has a unique root in I_m. Moreover, $\widetilde{X}_m \xrightarrow{p} \mu$, $\sqrt{m}\,(\widetilde{X}_m - \mu) \Rightarrow N(0, \sigma^2)$.

4.2 Heavy-Tailed Regime

For any confidence level $\delta \in (0, 1/2)$, choose $c_m(\delta) := \tau\left(\frac{v_{1+\epsilon}\, m}{\log(2/\delta)}\right)^{\frac{1}{1+\epsilon}}, \tau > 0$, and take any $r_m \to \infty$ with $r_m = o(c_m(\delta))$ (e.g., $r_m = c_m(\delta)^{1/2}$). Here $c_m(\delta)$ is tuned to balance (controlled) bias from downweighting large residuals with concentration of the empirical score.

Theorem 4.4. *Fix $0 < \alpha \le 1$ and $0 < \epsilon \le 1$. Assume $\mathbb{E}(X_1) = \mu, \mathbb{E}|X_1 - \mu|^{1+\epsilon} \le v^{1+\epsilon} < \infty$. Let $\delta \in (0, 1/2)$ and set $t_\delta := \log(4/\delta), c_m := \tau\Big((v^{1+\epsilon} m)/t_\delta\Big)^{1/(1+\epsilon)}$, where $\tau > 0$ is fixed. Let $r_m \to \infty$ satisfy $r_m = o(c_m)$, and assume the pilot estimator satisfies*

$$\mathbb{P}\big(|\widehat{\mu}_{\mathrm{pil}} - \mu| \le r_m/2\big) \ge 1 - \delta/4. \tag{11}$$

Let $\widetilde{X}_m$ be the unique root of $\widehat{g}_m(x) = 0$ in $I_m = [\widehat{\mu}_{\mathrm{pil}} - r_m, \widehat{\mu}_{\mathrm{pil}} + r_m]$, whenever this root exists and is unique. Then, for all sufficiently large m, $\mathbb{P}\Big(|\widetilde{X}_m - \mu| \le C_{\alpha,\epsilon}(\tau)\Big(\frac{v^{1+\epsilon} t_\delta}{m}\Big)^{\epsilon/(1+\epsilon)}\Big) \ge 1 - \delta$, where one may take

$$C_{\alpha,\epsilon}(\tau) := \frac{2}{\lambda_\alpha}\tau^{-\epsilon} + \frac{2^{2+\epsilon/2}\max\{1, L_\alpha\}}{\lambda_\alpha}\tau^{(1-\epsilon)/2} + \frac{8L_\alpha}{3\lambda_\alpha}\tau. \tag{12}$$

This single-block deviation bound is the basic ingredient for median-of-means aggregation, which amplifies constant-success probability to an arbitrary confidence level.

4.3 Asymptotic Theory for ARE

We now aggregate across blocks using a sample median. In the finite-variance regime, a block-level CLT combined with the classical median CLT yields a $\sqrt{n}$ limit and the usual $\pi/2$ variance inflation, giving a constant asymptotic relative efficiency.

Theorem 4.5. *Assume Assumption 4.2 and fix $0 < \alpha \le 1$. Let $k = k_n \to \infty$, $m = \lfloor n/k \rfloor \to \infty$, and compute each block estimator with the finite-variance tuning of Theorem 4.3. Let F_m denote the law of a generic block estimator $\widetilde{X}_m^{(m)}$, let q_m be its median, and let f_m be its density. Assume: (i)$\sqrt{n}\,(q_m - \mu) \to 0$; (ii) $f_m(q_m) = \dfrac{\sqrt{m}}{\sigma\sqrt{2\pi}}\,(1 + o(1))$; (iii) the triangular-array sample-median CLT applies to the i.i.d. row $\widetilde{X}_{1,m}, \ldots, \widetilde{X}_{k,m}$ with common law F_m, so that $\sqrt{k}\,(\widetilde{\mu} - q_m) \Rightarrow N\Big(0, \frac{1}{4f_m(q_m)^2}\Big)$. Then $\sqrt{n}\,(\widetilde{\mu} - \mu) \Rightarrow N\big(0, \frac{\pi}{2}\sigma^2\big)$. Consequently the asymptotic relative efficiency with respect to the sample mean is $2/\pi$.*

Theorem 4.6. *Assume the hypotheses of Theorem 4.4. Let $k := \left\lceil 8 \log \frac{2}{\delta} \right\rceil, m := \left\lfloor \frac{n}{k} \right\rfloor, \delta_0 := \frac{1}{4}$. On each block, build the central-root estimator with confidence level δ_0 and the same tuning rule as in Theorem 4.4. Assume the block pilots satisfy the analogue of* (11) *independently across blocks. Then, for all sufficiently large n, $\mathbb{P}\Big(|\widetilde{\mu} - \mu| \le C_{\alpha,\epsilon}^{\mathrm{ARE}}(\tau)\Big(\frac{v^{1+\epsilon}\log(2/\delta)}{n}\Big)^{\epsilon/(1+\epsilon)}\Big) \ge 1 - \delta$, where one may take*

$$C_{\alpha,\epsilon}^{\mathrm{ARE}}(\tau) := C_{\alpha,\epsilon}(\tau)\,[32 \log 16]^{\epsilon/(1+\epsilon)}. \tag{13}$$

In particular, the original statement with unspecified constants C_1, C_2 follows by absorbing the explicit numerical factors into (13)*.*

5 Experiments on Single-Agent

5.1 Mathematical reasoning

We evaluate the effectiveness of ARE in single-agent RL on mathematical reasoning tasks. Mathematical reasoning is a representative task for examining RL optimization, as it typically involves long-chain reasoning, sparse reward signals, and noise introduced by reward models. We train on MATH 7.5k, using `internlm2-1.8b-reward` as the reward model, and evaluate on MATH500 [Lightman et al., 2024], AIME2024, AIME2025, MinervaMath [Lewkowycz et al., 2022], and Gaokao2023 [Zhang et al., 2024] to assess the reasoning ability of the trained model across diverse benchmarks. Detailed settings can be referred to in Section I.1.

Table 1 reports mean accuracy and standard deviation over three random seeds for Qwen3-4B-Base and OLMo3-7B-Instruct. With Qwen3-4B-Base and Dr.GRPO, ARE increases mean MinervaMath accuracy from 12.57% to 21.45% and Gaokao2023 accuracy from 5.57% to 14.46%. Appendix I.4 reports rollout, entropy, and median-centering comparisons. For the REINFORCE++ run in Figure 3, total training time increases from 331.5 to 525.25 minutes (+58.45%); the additional computation comes from blockwise parameter calibration and GNC–IRLS updates.

Table 1: (Single-Agent) Accuracy (%): mean ± standard deviation over three random seeds.

Model	Method	MATH500	AIME2024	AIME2025	MinervaMath	Gaokao2023
Qwen3-4B-Base	Dr.GRPO	72.04 ± 0.73	12.06 ± 0.56	8.04 ± 1.44	12.57 ± 1.57	5.57 ± 0.55
	Dr.GRPO (ARE)	$\mathbf{72.71 \pm 0.61}$	$\mathbf{12.22 \pm 0.84}$	$\mathbf{9.79 \pm 0.63}$	$\mathbf{21.45 \pm 0.43}$	$\mathbf{14.46 \pm 0.54}$
	REINFORCE++	72.37 ± 0.91	8.33 ± 0.75	6.95 ± 0.61	19.60 ± 0.57	14.28 ± 0.69
	REINFORCE++ (ARE)	$\mathbf{73.50 \pm 0.23}$	$\mathbf{9.58 \pm 1.10}$	$\mathbf{8.72 \pm 1.07}$	$\mathbf{21.31 \pm 1.58}$	$\mathbf{14.81 \pm 0.52}$
OLMo3-7B-Instruct	Dr.GRPO	85.95 ± 0.70	42.40 ± 0.85	35.40 ± 0.75	21.08 ± 0.93	15.45 ± 0.65
	Dr.GRPO (ARE)	$\mathbf{86.70 \pm 0.62}$	$\mathbf{43.25 \pm 0.78}$	$\mathbf{36.75 \pm 0.68}$	$\mathbf{21.81 \pm 1.18}$	$\mathbf{16.60 \pm 0.59}$
	REINFORCE++	85.50 ± 0.78	44.40 ± 0.72	36.60 ± 0.68	20.83 ± 0.56	15.45 ± 0.60
	REINFORCE++ (ARE)	$\mathbf{86.42 \pm 0.66}$	$\mathbf{45.68 \pm 0.83}$	$\mathbf{38.02 \pm 0.72}$	$\mathbf{21.57 \pm 0.77}$	$\mathbf{16.25 \pm 0.63}$

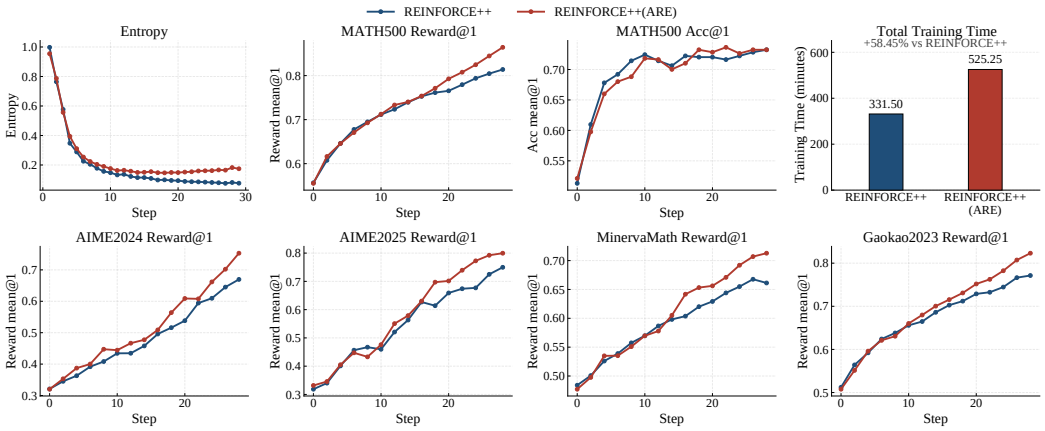


Figure 3: (Single-Agent) Dynamic comparison between REINFORCE++ and REINFORCE++(ARE) in the single-agent mathematical reasoning setting, including training entropy, reward, accuracy, and total training time.

5.2 Embodied VLN Evaluation

To assess whether ARE transfers beyond pure language reasoning, we evaluated it on an embodied aerial VLN task by applying ARE to a visual language model. We built on FlightGPT [Cai et al., 2025] and replaced its standard GRPO optimization with ARE, fine-tuning a Qwen2-VL-2B-Instruct backbone [Wang et al., 2024] with downscaled image resolutions to fit available compute. In VLN (Figure 4), the agent follows a natural-language instruction D and visual observations from a 3D environment E, starting from $(\mathbf{p}_0, \theta_0)$ and executing discrete actions $a_t \in \mathcal{A}$; an episode succeeds if the final position $\mathbf{p}_T$ is within a threshold δ of the target.

Following Cai et al. [2025], we reported four standard metrics: Navigation Error (NE), the Euclidean distance between the final position and the ground-truth target (lower is better); Success Rate (SR), the fraction of episodes whose final position is within 20 meters of the target (higher is better); Oracle Success Rate (OSR), the fraction of episodes that reach within 20 meters of the target at any time (higher is better); and Success weighted by Path Length (SPL), which penalizes unnecessarily long trajectories (higher is better).

Figure 5 shows that ARE achieved consistently higher training rewards than FlightGPT, suggesting improved optimization stability under the same VLN setup; although we used a smaller backbone than in Cai et al. [2025] (leading to lower absolute performance), the comparison to the GRPO baseline remained controlled. Consistently, Table 2 shows that ARE outperformed both the base model and FlightGPT across all splits, with the largest gains on unseen environments (e.g., NE 145.56→124.38 on $\text{val}_{\text{unseen}}$ and 140.78→129.71 on $\text{test}_{\text{unseen}}$) while also improving SR/OSR/SPL, indicating better localization, higher success, and more efficient trajectories.

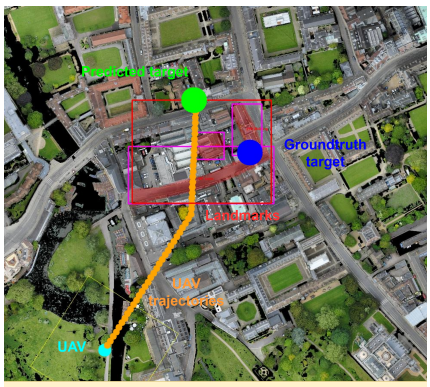


Figure 4: Visualization of the navigation task. The UAV trajectory, predicted target, and ground-truth target are annotated.

Table 2: (Single Agent) Single-Agent Aerial VLN performance across seen and unseen splits.

Method	VS				VUS				TUS			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Base	193.92	0.3300	1.0000	0.0000	190.84	1.3300	1.3300	0.0100	195.95	1.3300	2.3300	0.0100
FlightGPT	148.19	2.6700	13.670	0.0200	145.56	3.6700	18.000	0.0300	140.78	2.3300	17.330	0.0200
ARE	**139.49**	**3.0000**	**14.330**	**0.0300**	**124.38**	**6.3300**	**18.330**	**0.0600**	**129.71**	**4.6700**	**18.330**	**0.0400**

VS: val_seen, VUS: val_unseen, TUS: test_unseen. NE is lower better; SR, OSR, and SPL are higher better.

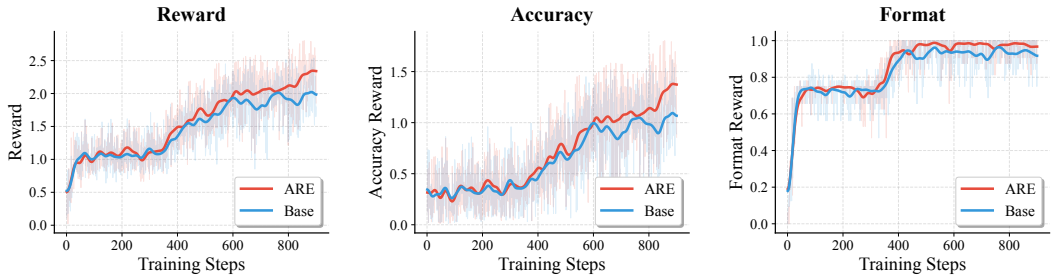


Figure 5: (Single-Agent) Training reward curves of ARE and FlightGPT.

Table 3: (Multi-Agent) ReMA accuracy (%): mean ± standard deviation over three random seeds.

Method	MATH500	AIME2024	AMC23	MinervaMath	Gaokao2023
ReMA	54.60 ± 2.71	7.78 ± 1.92	30.83 ± 1.44	12.25 ± 1.18	35.49 ± 0.91
ReMA (ARE)	**55.47 ± 2.37**	**11.11 ± 1.92**	**31.67 ± 1.44**	**12.99 ± 0.76**	**35.93 ± 1.05**

6 Experiments on Multi-Agent

6.1 The experiment on the ReMA collaborative model

In order to verify the scalability of ARE in a non-customized multi-agent reinforcement learning framework, we further embedded it into the ReMA [Wan et al., 2025] collaborative training framework. Unlike DACR, the design of ReMA involves two serially collaborating agents. We maintained the original collaborative structure and interaction sequence of ReMA. The purpose of this was to decouple the coupling influence between the interaction framework and the optimizer.

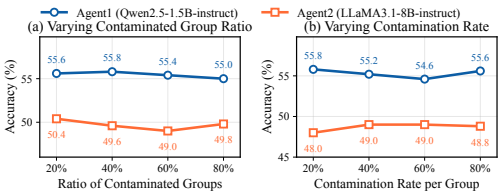


Figure 6: (Multi-Agent) The performance of the proposed method under different noise conditions is illustrated: the left figure shows contamination by noise groups of varying proportions, while the right figure depicts contamination ratios within the same group.

Table 3 reports mean accuracy and standard deviation over three random seeds. ARE increases mean accuracy from 54.60% to 55.47% on MATH500 and from 7.78% to 11.11% on AIME2024. The gains on AMC23, MinervaMath, and Gaokao2023 are 0.84, 0.74, and 0.44 percentage points, respectively.

6.2 The experiment on the DACR collaborative model

We evaluate the proposed ARE and DACR (Section B) on challenging mathematical reasoning benchmarks. We train on MATH 7.5k [Hendrycks et al., 2021] and test on other mathematical reasoning dataset. We compare against a baseline that matches our collaboration setting but replaces ARE with standard GRPO. We organize the empirical study around three research questions, examining overall effectiveness under noisy rewards.

To evaluate robustness under noisy rewards, we inject heavy-tailed, peaked Cauchy noise into the reward model and train on mathematical reasoning datasets. As shown in Table 4, our method consistently improves accuracy over the GRPO-based baseline.

We further stress-test robustness under diverse contamination patterns (Figure 6): (i) group-level corruption where a fraction of groups are selected and 20% of samples within each selected group are contaminated, and (ii) within-group corruption where the contamination ratio varies inside the same group. Both settings evaluate accuracy on MATH500. Across all conditions, our method degrades gracefully and remains consistently more robust to reward contamination than the baseline.

Table 4: (Multi-Agent) Comparison of accuracy rates on the mathematical reasoning. **GroupA**: qwen2.5-1.5b-instruct, qwen2.5-7b-instruct, **GroupB**: llama3.1-8b-instruct, qwen2.5-7b-instruct, **GroupC**: qwen2.5-1.5b-instruct, llama3.1-8b-instruct.

Method	MATH500	AIME24	AIME25	AMC23	Gaokao23	Minerva
Qwen2.5(1.5B)	35.80	-	-	30.00	23.50	5.150
baseline-A	56.00	-	-	22.50	45.50	13.20
baseline-C	55.00	**3.300**	-	22.50	45.50	13.60
ARE-A	**56.60**	**3.300**	-	**32.50**	**46.00**	14.00
ARE-C	55.60	**3.300**	**3.300**	27.50	45.50	**15.40**
Llama3.1(8B)	32.60	-	-	5.000	3.390	7.350
baseline-B	49.40	3.300	-	20.00	39.70	17.60
baseline-C	49.20	6.700	3.300	22.50	39.20	17.30
ARE-B	51.00	**6.700**	-	22.50	**42.60**	**19.90**
ARE-C	**50.40**	6.700	-	**27.50**	40.30	18.80
Qwen2.5(7B)	68.00	6.667	10.00	37.50	42.04	18.38
baseline-A	**77.60**	**16.70**	6.700	50.00	**62.60**	27.60
baseline-B	77.40	13.30	10.00	52.50	61.30	28.70
ARE-A	**77.60**	10.00	10.00	**60.00**	61.80	28.30
ARE-B	77.40	13.30	**10.00**	57.70	61.60	**30.50**

7 Conclusion

We study a critical yet underexplored problem in reinforcement learning-based post-training of large language models: the instability of policy optimization under noisy, biased, and heavy-tailed reward distributions. To address this issue, we propose ARE, a plug-and-play module that combines the MoM structure with an adaptive robust loss function. ARE can be seamlessly integrated into a wide range of policy optimization algorithms without modifying the underlying objective. Across both single-agent and multi-agent settings, as well as mathematical reasoning and embodied VLN tasks, ARE consistently improves training stability, accuracy, and generalization, outperforming standard baselines. Nevertheless, this work has several limitations. The computational overhead of ARE and its adaptive parameterization can be further optimized, and its interaction with other components, such as reward modeling and credit assignment, has not yet been systematically explored. Future work will address these limitations toward achieving fully end-to-end robust training.

Acknowledgments and Disclosure of Funding

This research work is supported by the National Key Research and Development Program of China under Grant NOs. 2024YFF0729003, the Fundamental Research Funds for the Central Universities. Huiming Zhang is supported in part by the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing and Beihang University under the Youth Talent Start-up Funding Project (No. KG16447001). Wan Tian is supported in part by the Natural Science Foundation of Hangzhou (No. 2025SZRJJ1388) and the Beijing Outstanding Young Scientist Program (No. JWZQ20240101027). Yijie Peng is supported by the National Natural Science Foundation of China (NSFC) (No. 72325007), the Xiangjiang Laboratory Key Project (No. 23XJ02004), and the Science and Technology Innovation Program of Hunan Province (No. 2024RC7003).

References

Jonathan T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, 19(1):57–91, July 1996. ISSN 0920-5691.

Alexander Bukharin, Ilgee Hong, Haoming Jiang, Zichong Li, Qingru Zhang, Zixuan Zhang, and Tuo Zhao. Robust reinforcement learning from corrupted human feedback. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. Flightgpt: Towards generalizable and interpretable uav vision-and-language navigation with vision-language models, 2025. URL `https://arxiv.org/abs/2505.12835`.

John E Dennis Jr and Roy E Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-simulation and Computation*, 7(4):345–359, 1978.

Donald Geman and Stuart Geman. Bayesian image analysis. In *Disordered systems and biological organization*, pages 301–319. Springer, 1986.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

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, 2021. URL `https://arxiv.org/abs/2103.03874`.

Thomas Hitchcox and James Richard Forbes. Mind the gap: Norm-aware adaptive robust loss for multivariate least-squares problems. *IEEE Robotics and Automation Letters*, 7(3):7116–7123, 2022.

Haodong Hong, Yanyuan Qiao, Sen Wang, Jiajun Liu, and Qi Wu. General scene adaptation for vision-and-language navigation. In *International Conference on Learning Representations*, 2025.

Hsiang Hsu, Eric Lei, and Chun-Fu Chen. Best-of-tails: Bridging optimism and pessimism in inference-time alignment. *arXiv preprint arXiv:2603.06797*, 2026. doi: 10.48550/arXiv.2603.06797.

Peter J Huber. Robust statistics. In *International encyclopedia of statistical science*, pages 1248–1251. Springer, 2011.

Pierre Humbert, Batiste Le Bars, and Ludovic Minvielle. Robust kernel density estimation with median-of-means principle. In *International Conference on Machine Learning*, pages 9444–9465. PMLR, 2022.

Kyungmin Jung, Thomas Hitchcox, and James Richard Forbes. An adaptive graduated nonconvexity loss function for robust nonlinear least-squares solutions. *IEEE Transactions on Robotics*, 2024.

Thomas Kwa, Drake Thomas, and Adrià Garriga-Alonso. Catastrophic goodhart: Regularizing rlhf with kl divergence does not mitigate heavy-tailed reward misspecification. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yvan G Leclerc. Constructing simple stable descriptions for image partitioning. *International journal of computer vision*, 3(1):73–102, 1989.

Guillaume Lecué and Matthieu Lerasle. Robust machine learning by median-of-means: Theory and practice. *The Annals of Statistics*, 48(2):906–931, 2020. doi: 10.1214/19-AOS1828.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL `https://arxiv.org/abs/2206.14858`.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Edwards Harri, 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.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. doi: 10.48550/arXiv.2503.20783.

Lu Lu, Yi Yu, Rodrigo C de Lamare, and Xiaomin Yang. Tukey's biweight m-estimate with conjugate gradient adaptive learning. *IEEE Signal Processing Letters*, 29:1117–1121, 2022.

MiniMax. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025. doi: 10.48550/arXiv.2506.13585.

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, 2022.

Helmut Rieder. *Robust Asymptotic Statistics: Volume I*. Springer Science & Business Media, 2012.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Qiang Sun. Do we need to estimate the variance in robust mean estimation? *arXiv preprint arXiv:2107.00118*, 2021.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. Rema: Learning to meta-think for llms with multi-agent reinforcement learning, 2025. URL `https://arxiv.org/abs/2503.09501`.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.

Lihu Xu, Fang Yao, Qiuran Yao, and Huiming Zhang. Non-Asymptotic Guarantees for Robust Statistical Learning under Infinite Variance Assumption. *Journal of Machine Learning Research*, 24(92):1–46, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark, 2024. URL `https://arxiv.org/abs/2305.12474`.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

A Related Work

A.1 Policy Optimization

Modern policy optimization methods for LLMs can be traced back to the trust-region line in RL. Trust Region Policy Optimization (TRPO) introduces an explicit trust-region constraint, typically enforced through a KL-based approximation, in order to guarantee stable policy improvement under constrained updates [Schulman et al., 2015]. PPO subsequently simplifies this idea by replacing the hard trust-region constraint with a clipped surrogate objective, achieving a favorable balance among stability, simplicity, and empirical performance [Schulman et al., 2017]. In the context of language model alignment, Ouyang et al. [2022] establishes PPO as the canonical optimization backbone in RLHF, making policy optimization a central component of post-training for instruction-following and alignment.

As RL began to be used for long-chain reasoning in LLMs, the limitations of standard PPO-style optimization became increasingly apparent, especially in terms of memory cost, critic dependence, and optimization instability under long or highly variable responses. To reduce reliance on value modeling, DeepSeekMath introduces GRPO, which replaces critic-based advantage estimation with groupwise relative reward normalization and demonstrates strong performance in mathematical reasoning [Shao et al., 2024]. Building on GRPO, subsequent work focuses on improving training stability and scalability in long-chain reasoning settings. DAPO refines the GRPO-style pipeline from a systems-and-algorithm perspective, emphasizing decoupled clipping, dynamic sampling, and long-CoT training efficiency [Yu et al., 2025]. In parallel, Liu et al. [2025] show that standard GRPO may suffer from optimization bias, especially length-related bias induced by its normalization design, and propose Dr. GRPO as a corrective variant.

A more recent line of work revisits the granularity at which policy ratios and clipping should be defined. GSPO argues that token-level importance ratios in GRPO-style methods can accumulate noise as sequence length grows, and therefore advocates sequence-level likelihood ratios and sequence-level clipping for more stable and efficient optimization [Zheng et al., 2025]. Similarly, MiniMax-M1 introduces CISPO, which clips importance sampling weights rather than token updates, further highlighting that recent policy optimization research is shifting from the original token-level PPO/GRPO formulation toward more stable sequence-level or weight-level trust-region surrogates [MiniMax, 2025]. Overall, the evolution from TRPO and PPO to GRPO, DAPO, Dr. GRPO, GSPO, and CISPO reflects a clear trend: policy optimization for modern LLMs is increasingly shaped by the need to handle long-horizon reasoning, reduce variance, and improve robustness and scalability in large-scale post-training.

A.2 Robust Estimation Methods

Robust estimation methods are designed to maintain statistical reliability when data deviate from idealized assumptions, such as in the presence of outliers, model misspecification, or heavy-tailed noise [Xu et al., 2023, Huber, 2011, Rieder, 2012]. Classical estimators—e.g., the sample mean or ordinary least squares—can be highly sensitive to a small fraction of corrupted observations, leading to unstable inference and degraded predictive performance. After decades of development, robust estimation methods can be categorized—according to their evolutionary trajectory—into the design of robust loss functions, adaptive procedures, tuning-free approaches, and median-of-means (MOM) estimators.

The core idea of robust loss functions is to minimize the influence of abnormally large per-sample residuals or gradients on the overall objective [Jung et al., 2024]. Representative choices include the Huber [Huber, 2011], pseudo-Huber [Hastie et al., 2001], Cauchy [Black and Anandan, 1996], Tukey's biweight [Lu et al., 2022], and Geman–McClure (GM) [Geman and Geman, 1986] losses. A key caveat, however, is the risk of over-robustness: overly aggressive down-weighting of extreme observations may suppress rare yet genuine signals. Moreover, the landscape of robust losses is broad, and selecting an appropriate form typically relies heavily on expert knowledge and empirical tuning. To mitigate this issue, Barron [2019] proposed an adaptive loss (as shown in (14)) which, via a shape parameter α governing tail behavior and robustness and a scale parameter c controlling the width of the near-zero quadratic basin, unifies many classical losses within a single continuously adjustable—and potentially automatically learnable—framework. Nevertheless, both classes of approaches may suffer from substantial computational overhead for tuning-parameter selection (e.g.,

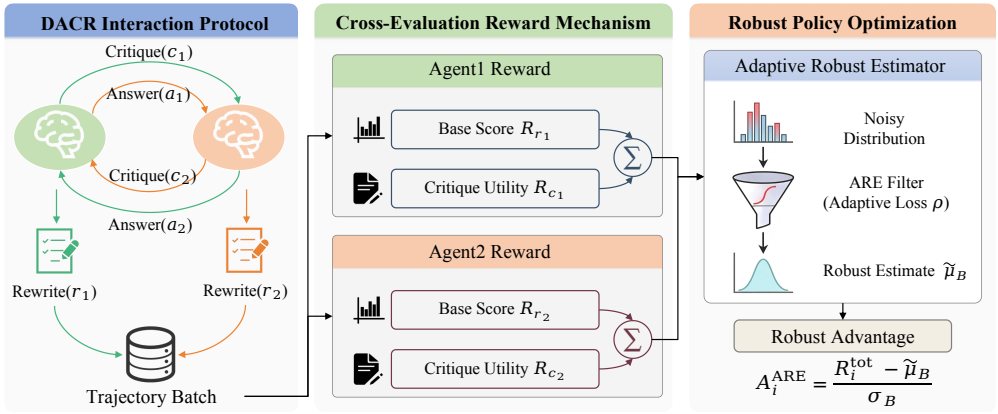


Figure 7: Overview of the DACR framework. Two agents π_1 and π_2 act as both solvers and peer evaluators. Each agent first produces an initial answer, then critiques its partner, and finally revises its own solution based on received feedback. The reward mechanism assigns credit based on both individual accuracy and the improvement induced in the partner's revised answer.

common choices such as Lepski's method and cross-validation can be computationally expensive), and the resulting objectives are often nonconvex [Jung et al., 2024], which can lead to poor local optima, sensitivity to initialization, and training instability.

To address these two major challenges, a third line of work—tuning-free methods—has emerged. The key idea is to determine the value of the tuning parameter (typically depending on the target estimation accuracy and the population second moment of the data) under which the resulting estimator can achieve a sub-Gaussian rate of convergence. A representative example is the work of Sun [2021], which builds on the pseudo-Huber loss and derives an explicit criterion for choosing the tuning parameter. A fourth class of approaches that has attracted considerable attention in recent years is the MoM estimator [Lecué and Lerasle, 2020, Humbert et al., 2022]. The procedure first partitions the sample into multiple groups, computes the sample mean within each group, and then takes the median of these group-wise means. Under the mere existence of a finite second moment, the MoM estimator can achieve sub-Gaussian rates of convergence. In Appendix C, we further discuss that, under both random contamination and adversarial contamination, the MoM estimator may still fail to be robust; moreover, attaining strong empirical performance typically requires a careful trade-off between the number of blocks and the sample size within each block.

B DACR Interaction Framework

In this appendix, we present the full design of the Dual-Agent Collaborative Reasoning (DACR) framework used in our experiments. DACR provides a structured interaction protocol together with a cross-evaluation reward mechanism for training collaborative reasoning agents.

B.1 Dual-Agent Interaction Protocol

DACR decomposes the reasoning process into a dual-agent system, where two policies π_1^θ and π_2^ϕ alternate between generative and evaluative roles. Given a problem q, the interaction proceeds through a structured three-stage trajectory:

1. **Independent Answering.** Each agent independently generates an initial reasoning chain and answer:
$$a_1 \sim \pi_1(q), \qquad a_2 \sim \pi_2(q).$$
2. **Mutual Critique.** The agents exchange their answers and provide critiques:
$$c_1 \sim \pi_1(q, a_2), \qquad c_2 \sim \pi_2(q, a_1).$$
3. **Revision.** Each agent refines its answer conditioned on peer feedback:
$$r_1 \sim \pi_1(q, a_1, c_2), \qquad r_2 \sim \pi_2(q, a_2, c_1).$$

This protocol explicitly separates reasoning and evaluation, allowing agents to receive external feedback signals rather than relying solely on their own reasoning trajectories. Such structured interaction helps expose hidden errors and encourages iterative refinement.

B.2 Cross-Evaluation Reward Mechanism

To train the agents, we adopt a reward design that captures both individual performance and collaborative contribution. Let $\mathrm{S}(y, q)$ denote a scoring function that evaluates the correctness of a response y to problem q.

We define stage-wise rewards for each agent $i \in \{1, 2\}$:

$$R_{\mathrm{ans},i} = \mathrm{S}(a_i, q), \qquad R_{\mathrm{rw},i} = \mathrm{S}(r_i, q).$$

To quantify collaboration, we introduce a cross-improvement term:

$$\Delta_1 = R_{\mathrm{rw},2} - R_{\mathrm{ans},2}, \qquad \Delta_2 = R_{\mathrm{rw},1} - R_{\mathrm{ans},1}.$$

The total reward for agent i is defined as

$$R_i^{\mathrm{total}} = R_{\mathrm{rw},i} + \gamma\, \Delta_i,$$

where $\gamma > 0$ controls the trade-off between individual accuracy and collaborative contribution.

This reward formulation encourages agents not only to produce correct answers, but also to generate informative critiques that improve their partner's reasoning. As a result, DACR promotes both accuracy and cooperative behavior within the training process.

DACR serves as a general interaction and reward framework for collaborative reasoning. Importantly, it is orthogonal to the choice of policy optimization algorithm and can be combined with different training objectives. In this work, we pair DACR with the proposed ARE-based robust policy optimization method (Section 2), enabling stable learning under noisy and heavy-tailed reward signals.

C Discussion on the MOM Estimator and its Robustness

In this section, we review the Median-of-Means (MoM) estimator as our baseline and discuss its key limitations under two contamination regimes: *random* and *adversarial* contamination.

Let $X_1, \ldots, X_n$ be i.i.d. real-valued random variables with mean $\mu = \mathbb{E}[X_i]$. The empirical mean $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ is statistically optimal under light-tailed distributions (e.g., sub-Gaussian), yet it can be highly unstable in the presence of heavy tails or contaminated samples. As noted in Section 1, MoM provides a canonical robust alternative by combining blockwise averaging with a robust aggregation step. Concretely, fix $k \geq 2$ and randomly partition $[n]$ into k disjoint blocks $B_1, \ldots, B_k$ of size $m = \lfloor n/k \rfloor$. Define the block means

$$\overline{X}_j = \frac{1}{|B_j|} \sum_{i \in B_j} X_i, \qquad j = 1, \ldots, k,$$

and aggregate them via the median:

$$\widehat{\mu} = \mathrm{median}(\overline{X}_1, \ldots, \overline{X}_k).$$

This construction exhibits a transparent robustness mechanism: blockwise averaging reduces the variance contributed by the uncontaminated observations, while the median discards the influence of a minority of corrupted blocks. In particular, under the mild assumption $\sigma^2 = \mathrm{Var}(X_i) < \infty$, MoM achieves a sub-Gaussian-type concentration bound. For any $\delta \in (0, 1)$, choosing $k \simeq c_0 \log(1/\delta)$ yields

$$\mathbb{P}\left(\left|\widehat{\mu} - \mu\right| \leq C_0 \sigma \sqrt{\frac{\log(1/\delta)}{n}} \right) \geq 1 - \delta,$$

for universal constants $c_0, C_0 > 0$ [Sun, 2021]. Hence, even without sub-Gaussian tails, MoM retains the optimal $n^{-1/2}$ rate and the canonical $\sqrt{\log(1/\delta)}$ confidence scaling. Practically, k controls the

robustness–efficiency trade-off: increasing k improves tolerance to corruption through the median at the cost of smaller block sizes. A common default is $k = \lceil c_0 \log(1/\delta) \rceil$ (with c_0 in the range 2–8); when δ is not specified, the heuristic $k \approx \lceil \log n \rceil$ is often used. We now turn to a simple analysis and discussion.

In random contamination, assume each observation is independently corrupted with probability $\kappa \in (0, 1)$. Randomly partition the n samples into k disjoint blocks, each of size $m = \lfloor n/k \rfloor$. Call a block *clean* if none of its m samples is contaminated. Under independent contamination, the probability that a fixed block is clean is

$$p_0 = \Pr(\text{a block is clean}) = (1 - \kappa)^m,$$

where we ignore the minor imbalance due to the floor operation and possible leftovers. Let $X_j = \mathbf{1}\{\text{block } j \text{ is clean}\}$ for $j = 1, \ldots, k$. Then $X_j \sim \text{Bernoulli}(p_0)$ i.i.d., and the number of clean blocks is

$$S := \sum_{j=1}^{k} X_j \sim \text{Binomial}(k, p_0).$$

A key mechanism behind MoM is that the median is taken over a *majority* of blocks; thus, if more than half of the blocks are clean (i.e., $S > k/2$), then the median is computed from predominantly uncontaminated block means. This motivates the condition $p_0 > 1/2$. When $p_0 > 1/2$, a Chernoff–Hoeffding (KL-form) bound yields

$$\Pr(S \le k/2) \le \exp\Big(-k \,\mathrm{KL}\big(1/2 \,\|\, p_0\big)\Big),$$

where the binary KL divergence is

$$\mathrm{KL}\big(1/2 \,\|\, p_0\big) = \frac{1}{2} \ln \frac{1/2}{p_0} + \frac{1}{2} \ln \frac{1/2}{1 - p_0}.$$

Hence, provided $p_0 > 1/2$, the failure event "at most half of the blocks are clean" has probability exponentially small in k.

The condition $p_0 > 1/2$ is equivalent to

$$(1 - \kappa)^m > \frac{1}{2} \quad \Longleftrightarrow \quad m < \frac{\ln 2}{-\ln(1 - \kappa)}.$$

For small κ, using $-\ln(1 - \kappa) \approx \kappa$, this becomes the convenient rule of thumb

$$m \lesssim \frac{\ln 2}{\kappa}.$$

Thus, smaller blocks (smaller m, larger k) make it easier to ensure that a majority of blocks are clean, which is precisely the regime in which MoM can retain sub-Gaussian-type deviation bounds under only finite second moments.

It is important to note that "smaller blocks are cleaner" does not mean "the smaller the better." Decreasing m increases $p_0 = (1 - \kappa)^m$ and improves the probability of having a clean majority, but it also increases the variance of each within-block mean because fewer samples are averaged within a block. Therefore, obtaining good estimation accuracy requires balancing the number of blocks k and the block size m.

In the adversarial contamination, an attacker can strategically place corrupted observations across blocks. In particular, it suffices to inject a single extremely large outlier into each of $\lceil k/2 \rceil$ distinct blocks to arbitrarily shift the MoM output. Indeed, within any contaminated block of size m, the block mean takes the form

$$\bar{X}_{\text{block}} = \frac{1}{m} \sum_{i=1}^{m} X_i,$$

so a single outlier of magnitude M shifts the block mean by

$$\Delta = \frac{M}{m}.$$

Although the multiplicative factor $1/m$ attenuates the effect, the attacker may choose M arbitrarily large, hence Δ can be made arbitrarily large. Consequently, once at least half of the blocks are contaminated, the median over block means can be driven to an arbitrary value. This yields a finite-sample breakdown point on the order of

$$\epsilon_{\mathrm{bd}} \approx \frac{\lceil k/2 \rceil}{n},$$

since contaminating $\lceil k/2 \rceil$ blocks requires only $\lceil k/2 \rceil$ adversarial points among n observations. In particular, if k is fixed (or grows sublinearly in n), then $\epsilon_{\mathrm{bd}} \to 0$ as $n \to \infty$, indicating extremely poor robustness in the strict breakdown-point sense. Equivalently, even when the overall contamination proportion is very small, MoM can be completely compromised if more than half of the blocks each contain at least one outlier.

Finally, the choice of k exhibits an intrinsic tension. If k is too large, then $m = \lfloor n/k \rfloor$ becomes too small and the variance of each block mean inflates; if k is too small, then blocks are too large and it becomes easier for contamination to hit a majority of blocks (especially under random contamination). Therefore, MoM entails a structural trade-off between k and m that is difficult to calibrate in practice.

D Simulation Studies of the ARE

In the main text, we validate the effectiveness of the ARE estimator through several concrete LLM applications. In this section, we further demonstrate its performance via simulation studies.

We consider one-dimensional robust mean estimation with ground-truth mean $\mu^\star = 1$. For each sample size $n \in \{200, 500, 1000, 2000, 5000\}$, we generate i.i.d. samples $X_1, \ldots, X_n$ under five data-generating processes (DGP1–DGP5) that progressively increase distributional difficulty from light tails to heavy tails and explicit corruption. DGP1 (Gaussian) serves as a light-tailed baseline:

$$X_i = \mu^\star + \sigma Z_i, \qquad Z_i \sim \mathcal{N}(0,1),$$

with default $\sigma = 1$. DGP2 (centered log-normal) captures skewed heavy tails while preserving the mean: letting $Y_i \sim \mathcal{N}(0,1)$,

$$X_i = \mu^\star + \Big(\exp(\tau Y_i) - \exp(\tau^2/2)\Big),$$

where $\tau > 0$ controls tail-heaviness and skewness (default $\tau = 1.5$). DGP3 (Student-t) models symmetric heavy tails with finite variance:

$$X_i = \mu^\star + sT_i, \qquad T_i \sim t_\nu, \qquad \nu > 2,$$

with default $(s, \nu) = (1, 4)$; in our implementation $T_i = Z_i/\sqrt{V_i/\nu}$ with $Z_i \sim \mathcal{N}(0,1)$ and $V_i \sim \chi^2(\nu)$. DGP4 (symmetric Pareto) probes the infinite-variance regime while keeping the mean finite:

$$X_i = \mu^\star + S_i U_i,$$

where $S_i \in \{-1, +1\}$ is Rademacher (equiprobable) and $U_i \geq 1$ has Pareto tail $\mathbb{P}(U_i > u) = u^{-\alpha}$ for $u \geq 1$, with $\alpha \in (1, 2)$ (default $\alpha = 1.5$). Finally, DGP5 (contamination) explicitly injects outliers on top of a clean baseline $X_i^{(0)} \sim \mathcal{N}(\mu^\star, \sigma_0^2)$ with default $\sigma_0 = 1$. Under random (mean-preserving) contamination, independently for each i we set $X_i = X_i^{(0)}$ with probability $1 - \kappa$, and with probability κ replace

$$X_i \leftarrow \mu^\star + S_i M,$$

where $S_i \in \{-1, +1\}$ is equiprobable (default $(\kappa, M) = (0.05, 100)$). Under adversarial one-sided contamination, after drawing the clean sample we replace exactly $\lfloor \kappa n \rfloor$ points by the outlier $\mu^\star + M$. To further stress-test median-of-means (MoM) style aggregation, we also consider a block-aware adversary: we deterministically partition the first mk samples into k blocks of equal size $m = \lfloor n/k \rfloor$ (discarding leftovers), then select $r = \lceil k/2 \rceil$ blocks and inject one outlier $\mu^\star + M$ into each selected block (optionally capped by a global budget $\lfloor \kappa n \rfloor$).

We compare the estimators `l2`, `Cauchy`, `GM`, `Adapt`, `AMB` (adaptive Maxwell–Boltzmann [Hitchcox and Forbes, 2022]), their GNC variants `GNC_Adapt`, `GNC_AMB`, `GNC_TLS` (truncated least squares), and the MoM-integrated variant `ARE`. Each method is initialized at the sample mean $\mu^{(0)} = \frac{1}{n}\sum_{i=1}^n X_i$

and returns a scalar estimate $\widehat{\mu}$. For each pair (DGP, n), we run $R = 100$ Monte Carlo replications. To ensure fair comparisons, we use common random numbers: in replication $r \in \{0, \ldots, R-1\}$ we set the seed to `base_seed` $+ r$ (default `base_seed` $= 12345$), generate one dataset, and evaluate all estimators on the same data. Performance is summarized by the error $e = \widehat{\mu} - \mu^\star$ through MAE and MSE,

$$\mathrm{MAE} = \mathbb{E}[|e|], \qquad \mathrm{MSE} = \mathbb{E}[e^2],$$

together with robust quantiles of $|e|$ (Median, Q90, and Q95 computed via linear interpolation on sorted absolute errors). Tables 5–9 report the experimental results for DGP1–DGP5 across all sample sizes and estimators.

Table 5: Results on DGP1 (Gaussian): robust error summaries over $R = 100$ replications.

n	Metric	l2	Cauchy	GM	Adapt	AMB	GNC_Adapt	GNC_AMB	GNC_TLS	ARE
	MAE	0.057287	0.074298	0.082726	0.089879	0.086309	0.089576	0.086304	0.067855	0.093291
	MSE	0.005173	0.008747	0.010770	0.012688	0.011817	0.012600	0.011710	0.007302	0.013051
200	Median	0.047519	0.062271	0.068645	0.074465	0.070920	0.074508	0.072363	0.056955	0.080019
	Q90	0.117608	0.152255	0.173105	0.185062	0.181983	0.185578	0.178264	0.140570	0.187690
	Q95	0.137931	0.199400	0.216909	0.229500	0.224137	0.225030	0.220215	0.163779	0.224390
	MAE	0.034494	0.044633	0.049594	0.053918	0.052523	0.053909	0.052433	0.041655	0.063132
	MSE	0.001838	0.002937	0.003645	0.004326	0.004069	0.004322	0.004044	0.002647	0.006196
500	Median	0.029021	0.040605	0.043591	0.046434	0.045928	0.046208	0.046618	0.035880	0.052503
	Q90	0.069328	0.089981	0.100150	0.106283	0.105652	0.106068	0.105136	0.085231	0.132819
	Q95	0.081448	0.099868	0.114402	0.126156	0.119561	0.126863	0.119259	0.097621	0.152423
	MAE	0.024141	0.032439	0.036162	0.039273	0.038195	0.039322	0.038274	0.029353	0.044525
	MSE	0.000877	0.001590	0.001982	0.002343	0.002228	0.002344	0.002231	0.001330	0.003116
1000	Median	0.020658	0.028489	0.031086	0.034594	0.034696	0.034421	0.034927	0.025598	0.037707
	Q90	0.048351	0.064046	0.073582	0.079218	0.079125	0.079847	0.079148	0.059341	0.089249
	Q95	0.057840	0.077597	0.085396	0.092410	0.090472	0.092422	0.089938	0.068942	0.108791
	MAE	0.018659	0.025467	0.028122	0.030252	0.029671	0.030232	0.029631	0.022884	0.035059
	MSE	0.000534	0.000973	0.001192	0.001395	0.001326	0.001394	0.001323	0.000812	0.001860
2000	Median	0.016846	0.022130	0.024437	0.026106	0.026217	0.026044	0.026388	0.020445	0.029204
	Q90	0.038356	0.052066	0.056480	0.061172	0.058984	0.061170	0.058737	0.046448	0.072581
	Q95	0.044255	0.059942	0.066329	0.071238	0.069291	0.071240	0.068512	0.055781	0.082386
	MAE	0.011756	0.015152	0.016491	0.017618	0.017328	0.017611	0.017323	0.014490	0.020988
	MSE	0.000215	0.000365	0.000439	0.000507	0.000490	0.000507	0.000489	0.000317	0.000683
5000	Median	0.010020	0.012336	0.013943	0.014923	0.014729	0.014922	0.014766	0.013010	0.017922
	Q90	0.023354	0.032167	0.035496	0.037974	0.037047	0.037991	0.036445	0.027374	0.041582
	Q95	0.027684	0.036030	0.040371	0.043431	0.043515	0.043434	0.042303	0.033452	0.049732

Table 6: Results on DGP2 (Centered LogNormal): robust error summaries over $R = 100$ replications.

n	Metric	l2	Cauchy	GM	Adapt	AMB	GNC_Adapt	GNC_AMB	GNC_TLS	ARE
	MAE	0.440732	2.383597	2.428866	2.437117	2.441007	2.457052	2.348575	2.228842	2.442037
	MSE	0.319861	5.684898	5.905568	5.977074	5.964692	6.039979	5.522237	4.972983	5.968493
200	Median	0.374759	2.385970	2.433266	2.458548	2.446208	2.460386	2.348119	2.225001	2.446614
	Q90	0.835407	2.453506	2.501665	2.523363	2.513925	2.521825	2.452893	2.326485	2.530520
	Q95	1.092987	2.474307	2.514922	2.537888	2.530098	2.537393	2.473458	2.343185	2.551792
	MAE	0.284348	2.384501	2.431850	2.456983	2.446077	2.457022	2.342375	2.230460	2.456446
	MSE	0.139719	5.686893	5.914873	6.037721	5.984414	6.037914	5.489974	4.976731	6.035749
500	Median	0.233371	2.385287	2.431715	2.457309	2.446462	2.457377	2.335733	2.230171	2.458230
	Q90	0.599346	2.426119	2.470482	2.496481	2.488803	2.496624	2.425090	2.284673	2.504184
	Q95	0.708262	2.434107	2.482159	2.508471	2.499906	2.508461	2.444517	2.296450	2.517809
	MAE	0.211270	2.385299	2.432640	2.457747	2.448701	2.457779	2.329159	2.231928	2.457179
	MSE	0.077294	5.690247	5.918278	6.041038	5.996701	6.041196	5.426657	4.982442	6.038572
1000	Median	0.172539	2.384034	2.432140	2.457033	2.448279	2.457028	2.324848	2.231803	2.457498
	Q90	0.424567	2.417506	2.463839	2.489640	2.481998	2.489626	2.387608	2.270945	2.494162
	Q95	0.530385	2.423898	2.470188	2.493565	2.486350	2.493547	2.406430	2.283688	2.506638
	MAE	0.156645	2.385716	2.433062	2.458169	2.449900	2.458201	2.318107	2.231655	2.458427
	MSE	0.043052	5.691973	5.920085	6.042870	6.002305	6.043028	5.374561	4.980825	6.044297
2000	Median	0.121270	2.386073	2.433150	2.457977	2.449910	2.458020	2.313820	2.231244	2.459061
	Q90	0.320521	2.409282	2.456259	2.480855	2.473205	2.480893	2.358750	2.261899	2.485424
	Q95	0.394085	2.417107	2.461496	2.485910	2.478026	2.485956	2.375311	2.270079	2.491608
	MAE	0.105697	2.385195	2.432413	2.457464	2.449503	2.457496	2.307456	2.231414	2.458300
	MSE	0.018055	5.689290	5.916752	6.039241	6.000178	6.039394	5.324652	4.979428	6.043399
5000	Median	0.091034	2.385240	2.432084	2.457331	2.449294	2.457379	2.306014	2.231975	2.458453
	Q90	0.216724	2.399676	2.445936	2.470685	2.462998	2.470644	2.330204	2.251057	2.474467
	Q95	0.262409	2.405306	2.450735	2.475080	2.466920	2.475065	2.338020	2.255622	2.477549

Table 7: Results on DGP3 (Student-t): robust error summaries over $R = 100$ replications.

n	Metric	l2	Cauchy	GM	Adapt	AMB	GNC_Adapt	GNC_AMB	GNC_TLS	ARE
200	MAE	0.079329	0.077027	0.085139	0.091804	0.089159	0.091806	0.088340	0.075760	0.106217
	MSE	0.010140	0.009076	0.011105	0.013011	0.012191	0.013010	0.011964	0.009004	0.017967
	Median	0.065749	0.066426	0.074241	0.079843	0.077080	0.079817	0.076304	0.065225	0.086817
	Q90	0.165889	0.156022	0.175618	0.185670	0.181116	0.185723	0.181239	0.155254	0.216114
	Q95	0.197331	0.181987	0.201629	0.219082	0.212402	0.219154	0.211227	0.181042	0.265095
500	MAE	0.053192	0.050686	0.055188	0.059119	0.057425	0.059175	0.057249	0.050691	0.070144
	MSE	0.004389	0.003959	0.004725	0.005423	0.005150	0.005426	0.005107	0.004040	0.007499
	Median	0.047555	0.042452	0.048275	0.049438	0.048677	0.049481	0.048193	0.041320	0.063075
	Q90	0.108891	0.100059	0.111604	0.121137	0.118760	0.120968	0.118494	0.103940	0.144723
	Q95	0.124714	0.122820	0.130709	0.143039	0.141928	0.143011	0.136721	0.123777	0.162166
1000	MAE	0.036055	0.034389	0.036687	0.038872	0.038216	0.038801	0.038277	0.035842	0.045915
	MSE	0.002049	0.001818	0.002119	0.002405	0.002328	0.002397	0.002328	0.001978	0.003375
	Median	0.030219	0.029908	0.031209	0.032017	0.031530	0.031926	0.031413	0.030449	0.039262
	Q90	0.076757	0.071994	0.077547	0.081623	0.079673	0.081599	0.080059	0.072346	0.096297
	Q95	0.087859	0.085480	0.095375	0.100148	0.098976	0.100240	0.098355	0.083406	0.117685
2000	MAE	0.027005	0.025412	0.027685	0.029742	0.029301	0.029659	0.029244	0.026076	0.035955
	MSE	0.001130	0.001019	0.001203	0.001371	0.001331	0.001361	0.001327	0.001082	0.002001
	Median	0.024092	0.021646	0.023765	0.025259	0.025824	0.025264	0.025472	0.022653	0.029967
	Q90	0.053873	0.050953	0.057300	0.062042	0.060372	0.061152	0.059646	0.054660	0.073035
	Q95	0.063653	0.059921	0.065569	0.071133	0.069951	0.070878	0.070029	0.063390	0.083720
5000	MAE	0.016767	0.015550	0.016721	0.017673	0.017497	0.017613	0.017462	0.016209	0.022485
	MSE	0.000438	0.000380	0.000446	0.000505	0.000494	0.000503	0.000493	0.000417	0.000775
	Median	0.014579	0.013322	0.014540	0.015093	0.015065	0.015025	0.015005	0.013681	0.019660
	Q90	0.033488	0.031232	0.033738	0.035248	0.035180	0.035245	0.035151	0.033712	0.045893
	Q95	0.041146	0.037998	0.041529	0.044344	0.043462	0.044339	0.043444	0.041594	0.051865

Table 8: Results on DGP4 (Symmetric Pareto): robust error summaries over $R = 100$ replications.

n	Metric	l2	Cauchy	GM	Adapt	AMB	GNC_Adapt	GNC_AMB	GNC_TLS	ARE
200	MAE	0.412362	1.438328	1.518497	1.522176	1.517095	1.500790	1.564941	1.595852	1.326587
	MSE	0.590463	2.072604	2.413169	2.459166	2.413946	2.255401	2.458244	2.645237	1.767638
	Median	0.253832	1.436787	1.502406	1.498419	1.502898	1.501094	1.564170	1.650789	1.325492
	Q90	0.848063	1.514769	1.572911	1.565906	1.574786	1.566497	1.664394	1.763932	1.438845
	Q95	1.155875	1.537037	1.602260	1.596185	1.600452	1.599216	1.694309	1.795882	1.461990
500	MAE	0.346482	1.439051	1.505764	1.513167	1.516621	1.497524	1.574814	1.638352	1.339161
	MSE	0.441919	2.072407	2.268632	2.356217	2.371738	2.243820	2.482337	2.718181	1.798361
	Median	0.214237	1.435626	1.503844	1.501487	1.503984	1.495642	1.576356	1.655792	1.345156
	Q90	0.679670	1.488627	1.554075	1.548618	1.551994	1.545145	1.633025	1.723029	1.425720
	Q95	1.001716	1.505555	1.565078	1.559460	1.565327	1.556368	1.648180	1.741259	1.447595
1000	MAE	0.404736	1.435188	1.568055	1.499761	1.502030	1.496583	1.581408	1.650191	1.172465
	MSE	6.035140	2.060631	4.486813	2.249927	2.260321	2.244539	2.502444	2.742182	1.592772
	Median	0.174446	1.433928	1.502667	1.497949	1.503413	1.497246	1.585403	1.658140	1.350682
	Q90	0.617388	1.473958	1.538876	1.532762	1.537795	1.533496	1.626974	1.709116	1.419330
	Q95	0.977243	1.485230	1.548406	1.543379	1.549137	1.545707	1.641047	1.721744	1.435046
2000	MAE	0.277339	1.483650	1.553332	1.549032	1.553667	1.498893	1.583741	1.640998	1.358529
	MSE	1.570783	3.414067	3.633623	3.623838	3.619150	2.247003	2.509354	2.721683	1.850885
	Median	0.134661	1.434549	1.503553	1.499047	1.504036	1.497680	1.589020	1.660958	1.364235
	Q90	0.499724	1.461581	1.527771	1.521979	1.528136	1.521673	1.621502	1.693832	1.421070
	Q95	0.723037	1.468161	1.534138	1.527576	1.533661	1.527547	1.628375	1.702626	1.432571
5000	MAE	0.172371	1.433381	1.503061	1.498617	1.504033	1.498603	1.585251	1.610063	1.376493
	MSE	0.291315	2.054726	2.259312	2.245967	2.262231	2.245924	2.518299	2.668343	1.896178
	Median	0.095233	1.433980	1.502981	1.498358	1.503860	1.498899	1.592539	1.660890	1.377615
	Q90	0.312437	1.447404	1.516828	1.511865	1.517150	1.511360	1.608947	1.678992	1.423776
	Q95	0.446499	1.452532	1.519658	1.515645	1.521086	1.515568	1.613800	1.682779	1.436759

Table 9: Results on DGP5 (Random contamination): robust error summaries over $R = 100$ replications.

n	Metric	l2	Cauchy	GM	Adapt	AMB	GNC_Adapt	GNC_AMB	GNC_TLS	ARE
200	MAE	1.275762	0.076195	0.085042	0.092851	0.088960	0.092902	0.082093	0.069203	0.101623
	MSE	2.589767	0.009235	0.011385	0.013490	0.012538	0.013503	0.010758	0.007540	0.016209
	Median	1.034393	0.063265	0.071296	0.077451	0.074361	0.077476	0.064493	0.058816	0.085344
	Q90	2.562352	0.154705	0.171824	0.187716	0.182085	0.187708	0.170956	0.143705	0.219979
	Q95	3.040650	0.198719	0.218241	0.232770	0.224837	0.232767	0.207915	0.179176	0.251584
500	MAE	0.803141	0.045911	0.051108	0.055784	0.054351	0.055812	0.051871	0.043391	0.065892
	MSE	0.999150	0.003058	0.003778	0.004492	0.004241	0.004495	0.003863	0.002878	0.006678
	Median	0.647632	0.042344	0.047195	0.050444	0.050155	0.050443	0.046949	0.037627	0.055327
	Q90	1.648696	0.088117	0.097799	0.107246	0.102712	0.107400	0.100482	0.087705	0.133173
	Q95	1.987630	0.103240	0.113667	0.125033	0.119585	0.124933	0.113398	0.103239	0.162270
1000	MAE	0.559301	0.033795	0.037628	0.040745	0.039726	0.040750	0.038179	0.030827	0.047724
	MSE	0.478545	0.001704	0.002122	0.002517	0.002390	0.002517	0.002215	0.001423	0.003510
	Median	0.493729	0.028837	0.032411	0.035934	0.034792	0.035898	0.033473	0.027246	0.038480
	Q90	1.097783	0.068452	0.077436	0.084156	0.082611	0.084147	0.076178	0.059305	0.097886
	Q95	1.313985	0.080211	0.088760	0.097425	0.095241	0.097411	0.092335	0.073964	0.115841
2000	MAE	0.402444	0.025673	0.028418	0.030721	0.030015	0.030728	0.029719	0.023478	0.036463
	MSE	0.252472	0.000999	0.001222	0.001433	0.001362	0.001434	0.001334	0.000846	0.002092
	Median	0.342546	0.022232	0.024743	0.026295	0.026008	0.026305	0.026590	0.020136	0.030277
	Q90	0.814902	0.052501	0.057734	0.063119	0.060448	0.063057	0.059638	0.046563	0.077859
	Q95	0.983756	0.063525	0.067954	0.074476	0.071649	0.074469	0.071677	0.054432	0.091296
5000	MAE	0.254088	0.015730	0.017206	0.018488	0.018124	0.018487	0.018104	0.014815	0.023329
	MSE	0.099986	0.000389	0.000470	0.000546	0.000526	0.000546	0.000520	0.000330	0.000835
	Median	0.214689	0.013191	0.014152	0.015257	0.014829	0.015312	0.014979	0.013506	0.020551
	Q90	0.528963	0.031760	0.035809	0.038966	0.038035	0.038986	0.038560	0.027583	0.047545
	Q95	0.641245	0.037724	0.041986	0.045497	0.044011	0.045536	0.043942	0.032985	0.055688

Across the five data-generating processes (DGP1–DGP5), the dominant pattern is that `ARE` offers its clearest benefits precisely in the regimes where robustness is genuinely needed: skewed heavy tails, infinite-variance tails, and explicit contamination. In the skewed heavy-tailed DGP2 (centered log-normal) and the infinite-variance DGP4 (symmetric Pareto with tail index $\xi \in (1, 2)$), several non-MoM robust losses (e.g., `Cauchy`/`GM`/`Adapt`/`AMB` and their GNC counterparts) exhibit a pronounced non-vanishing error floor: their MAE and Median remain essentially flat as n grows, with high quantiles (Q90/Q95) also failing to improve materially. In contrast, `ARE` consistently suppresses the tail risk in these settings, yielding markedly smaller Q90/Q95 and, in DGP4, substantially improved overall error summaries at representative sample sizes (e.g., a striking reduction in MAE/MSE at $n = 1000$). This behavior is consistent with the two-layer robustness of the method: the inner GNC mechanism stabilizes the optimization of a nonconvex robust objective, while the outer median-of-means aggregation truncates the influence of a minority of blocks containing extreme observations, thereby preventing the estimator from being dominated by rare but catastrophic samples.

The advantage becomes even more transparent under explicit contamination (DGP5). Under random contamination, the sample mean (`l2`) suffers a dramatic degradation in all metrics, while `ARE` maintains small errors and noticeably better tail quantiles, indicating strong protection against sporadic outliers. Under adversarial contamination, `l2` is effectively "locked" by the injected outliers: its MAE stays near 5 and does not meaningfully decrease with n, reflecting a complete breakdown of classical averaging. In sharp contrast, `ARE` remains accurate and continues to improve with n, with particularly strong control of Q90/Q95, which highlights its robustness to worst-case corruption consistent with MoM-type guarantees (so long as a majority of blocks remain uncontaminated). Importantly, in the benign baselines DGP1 (Gaussian) and DGP3 (Student-t with finite variance), `ARE` incurs only a modest constant-factor efficiency loss relative to the best light-tail methods (e.g., `l2` or `GNC_TLS`) and does not exhibit instability. Overall, these results suggest that `ARE` trades a small amount of efficiency in easy regimes for substantial and reliable gains in heavy-tailed and contaminated regimes, with its most consistent improvements appearing in the high-quantile error metrics (Q90/Q95) that directly capture tail-risk robustness.

E Introduction to Adaptive Losses.

In this paper, we employ the adaptive loss function introduced by Barron [2019], defined as follows:

$$\rho(\epsilon; \alpha, c) = \frac{|\alpha - 2|}{\alpha} \left(\left(1 + \frac{(\epsilon/c)^2}{|\alpha - 2|} \right)^{\alpha/2} - 1 \right), \tag{14}$$

where $\alpha \in (-\infty, 2]$ is a shape parameter controlling the loss's robustness, while $c > 0$ is a scale parameter governing the quadratic bowl near $\epsilon = 0$. Through simple analysis, we can immediately recover several common losses—such as the L2, Charbonnier/pseudo-Huber/L1-L2, Cauchy/Lorentzian [Black and Anandan, 1996], Geman–McClure [Geman and Geman, 1986], and Welsch/Leclerc losses [Dennis Jr and Welsch, 1978, Leclerc, 1989]—as special cases of the adaptive loss $\rho(\epsilon; \alpha, c)$ in (14) by setting α to particular values, as summarized as follows,

$$\rho(\epsilon; \alpha, c) = \begin{cases} \frac{1}{2}(\epsilon/c)^2 & \alpha = 2 \\ \sqrt{(\epsilon/c)^2 + 1} - 1 & \alpha = 1 \\ \log\left(\frac{1}{2}(\epsilon/c)^2 + 1\right) & \alpha = 0 \\ 2(\epsilon/c)^2 / \left((\epsilon/c)^2 + 4\right) & \alpha = -2 \\ 1 - \exp\left(-\frac{1}{2}(\epsilon/c)^2\right) & \alpha = -\infty \end{cases}, \quad \frac{\partial \rho}{\partial \epsilon}(\epsilon; \alpha, c) = \begin{cases} \epsilon/c^2 & \alpha = 2 \\ (\epsilon/c^2)/\sqrt{(\epsilon/c)^2 + 1} & \alpha = 1 \\ 2\epsilon/(\epsilon^2 + 2c^2) & \alpha = 0 \\ (\epsilon/c^2)\left((\epsilon/c)^2/4 + 1\right)^{-3/2} & \alpha = -2 \\ (\epsilon/c^2)\exp\left(-\frac{1}{2}(\epsilon/c)^2\right) & \alpha = -\infty \end{cases}$$

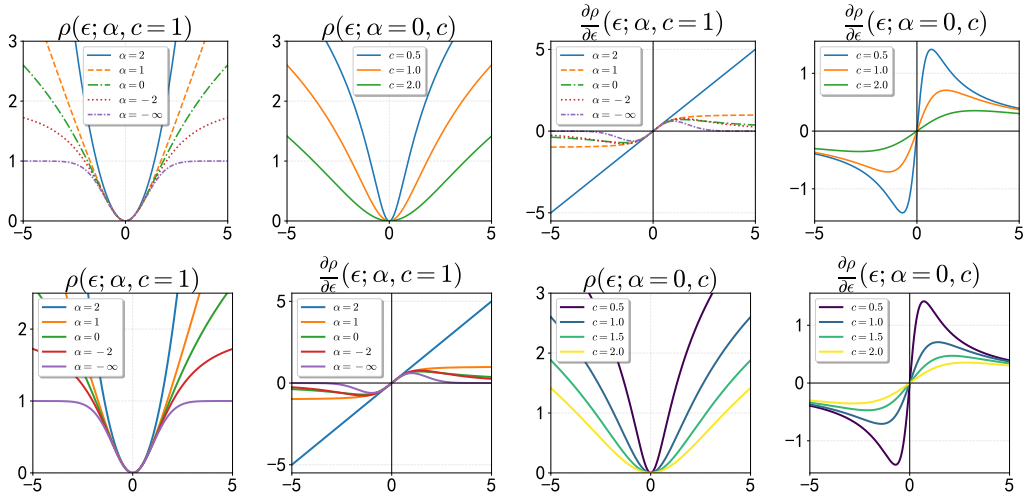


Figure 8: **Adaptive robust loss $\rho(\epsilon; \alpha, c)$ and its gradient under different parameter settings.** From left to right and top to bottom, the four panels show: (*top-left*) the loss $\rho(\epsilon; \alpha, c)$ for varying shape α with fixed $c = 1$; (*top-right*) the corresponding gradient $\partial\rho/\partial\epsilon$ for varying α with fixed $c = 1$; (*bottom-left*) the loss for varying scale c with fixed $\alpha = 0$; (*bottom-right*) the corresponding gradient for varying c with fixed $\alpha = 0$.

The adaptive loss $\rho(\epsilon; \alpha, c)$ provides a unified two-parameter family that spans a broad spectrum of robust penalties through a shape parameter $\alpha \in (-\infty, 2]$ and a scale parameter $c > 0$. As illustrated in Figure 8, decreasing α continuously interpolates from the quadratic ℓ_2 loss ($\alpha = 2$), which assigns unbounded influence to large residuals, to increasingly robust losses whose growth slows down for large $|\epsilon|$. This continuum makes α a direct knob for robustness: smaller α reduces the marginal penalty on large errors and thus mitigates the effect of outliers or heavy-tailed noise. The scale parameter c controls the width of the near-zero quadratic regime. For fixed α (e.g., $\alpha = 0$), larger c enlarges the approximately quadratic region around $\epsilon = 0$, thereby behaving more like least squares on moderate residuals and improving statistical efficiency under mild noise. Conversely, smaller c triggers earlier deviation from quadratic growth, increasing robustness by treating moderate-to-large residuals more conservatively. Hence, c mediates a local efficiency–robustness trade-off by determining the transition point at which the loss begins to downweight large errors.

The gradients $\partial\rho/\partial\epsilon$ make the robustness mechanism explicit through the induced influence function. When α is small, the gradient decays rapidly as $|\epsilon|$ grows and can approach zero for very large residuals, effectively suppressing the contribution of extreme observations during optimization. This bounded (or rapidly vanishing) influence stabilizes gradient-based updates and prevents optimization from being dominated by a small number of corrupted samples. The scale c further adjusts the decay rate of $\partial\rho/\partial\epsilon$: smaller c yields faster saturation (stronger outlier rejection but potentially slower progress when many residuals are moderately large), whereas larger c preserves larger gradients over a wider range (often faster convergence locally but weaker robustness). Overall, the (α, c) parameterization enables data-dependent control of robustness versus efficiency within a single

coherent framework, which is particularly useful for robust estimation and learning under heavy-tailed or contaminated data.

F The GNC Paradigm for Mean Estimation in Subsection 3.2

This appendix details the graduated nonconvexity (GNC) procedure used to solve the nonconvex subproblem (5) (equivalently (9)) after obtaining $(\alpha^{(1)}, c^{(1)})$. We present (i) a GNC surrogate for the adaptive loss, (ii) the induced IRLS weights and the associated Black–Rangarajan (BR) regularization, and (iii) a complete GNC-IRLS algorithm with practical continuation schedules and stopping criteria.

Let $f(\beta, \alpha^{(1)})$ be a shape mapping controlled by a continuation parameter β. We construct a one-parameter family of surrogate losses by replacing the shape parameter $\alpha^{(1)}$ with $f(\beta, \alpha^{(1)})$:

$$\rho_\beta(\epsilon; \alpha^{(1)}, c^{(1)}) \;\coloneqq\; \rho\Big(\epsilon;\; f(\beta, \alpha^{(1)}),\; c^{(1)}\Big)\,, \tag{15}$$

where $\rho(\cdot; \alpha, c)$ denotes the adaptive robust loss used in the main text. To avoid ambiguity and ensure numerical stability, we recommend writing ρ_β in the standard Barron form with a scale parameter:

$$\rho_\beta(\epsilon; \alpha^{(1)}, c^{(1)}) = \frac{|f-2|}{f}\left[\left(\frac{(\epsilon/c^{(1)})^2}{|f-2|}+1\right)^{\frac{f}{2}} - 1\right], \qquad f \equiv f(\beta, \alpha^{(1)}). \tag{16}$$

This definition guarantees the correct quadratic limit as $f \to 2$ and matches the commonly used scaling convention (cf. Barron-style adaptive losses). If a different scaling convention is adopted in the main text, (16) should be adjusted accordingly; the derivations below remain identical up to constant factors.

A valid shape mapping must satisfy

$$\lim_{\beta\to\beta_1} f(\beta, \alpha^{(1)}) = 2, \qquad \lim_{\beta\to\beta_2} f(\beta, \alpha^{(1)}) = \alpha^{(1)}, \qquad f(\beta, \alpha^{(1)}) \le 2, \tag{17}$$

where β_1, β_2 are surrogate-dependent constants (or endpoints) determined by the chosen parameterization of β. Intuitively, $f(\beta, \alpha^{(1)}) \to 2$ yields a convex quadratic surrogate, while $f(\beta, \alpha^{(1)}) \to \alpha^{(1)}$ recovers the original adaptive loss (typically nonconvex). We consider three concrete choices of $f(\beta, \alpha^{(1)})$:

Example 1 (Polynomial decay; $\beta:\ \infty \to 1$).

$$f_1(\beta, \alpha^{(1)}) = 2 - \frac{2-\alpha^{(1)}}{\beta^p}, \qquad p > 0, \tag{18}$$

so that $\beta \to \infty$ implies $f_1 \to 2$ and $\beta \to 1$ implies $f_1 \to \alpha^{(1)}$.

Example 2 (Exponential transition; $\beta:\ 0^+ \to \infty$).

$$f_2(\beta, \alpha^{(1)}) = \alpha^{(1)} e^{-1/\beta} + 2\big(1 - e^{-1/\beta}\big), \tag{19}$$

so that $\beta \to 0^+$ implies $f_2 \to 2$ and $\beta \to \infty$ implies $f_2 \to \alpha^{(1)}$.

Example 3 (Rational form; $\beta:\ 0^+ \to \infty$).

$$f_3(\beta, \alpha^{(1)}) = \frac{2 + \alpha^{(1)}\beta^q}{1+\beta^q}, \qquad q > 0, \tag{20}$$

so that $\beta \to 0^+$ implies $f_3 \to 2$ and $\beta \to \infty$ implies $f_3 \to \alpha^{(1)}$.

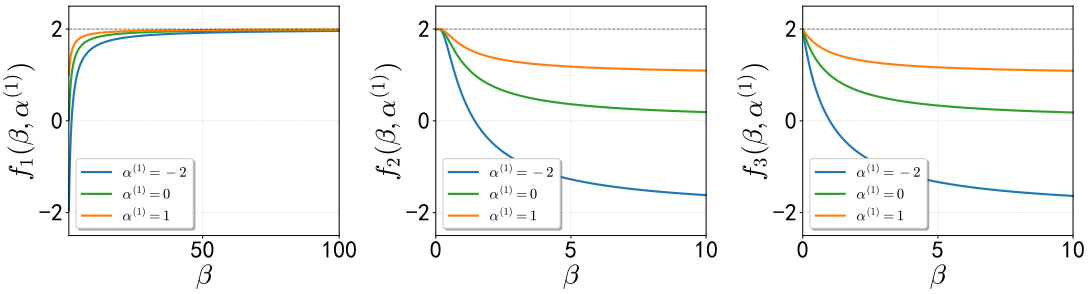


Figure 9: Three instantiations of the interpolation function $f(\beta, \alpha^{(1)})$: (a) polynomial decay $f_1 = 2 - \frac{2-\alpha^{(1)}}{\beta^p}$ with $\beta : \infty \to 1$; (b) exponential transition $f_2 = \alpha^{(1)} e^{-1/\beta} + 2(1 - e^{-1/\beta})$ with $\beta : 0^+ \to \infty$; (c) rational form $f_3 = \frac{2+\alpha^{(1)}\beta^q}{1+\beta^q}$ with $\beta : 0^+ \to \infty$, all smoothly interpolating between the quadratic regime ($f = 2$) and the adaptive regime ($f = \alpha^{(1)}$) for different values of $\alpha^{(1)} \in \{-2, 0, 1\}$.

Let $\epsilon_i = X_i - X$ and consider the surrogate objective

$$\min_{X \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^{m} \rho_\beta(\epsilon_i; \alpha^{(1)}, c^{(1)}). \tag{21}$$

At fixed $(\beta, \alpha^{(1)}, c^{(1)})$, IRLS solves a sequence of weighted least-squares problems

$$\min_{X \in \mathbb{R}} \frac{1}{2m} \sum_{i=1}^{m} w_i \, \epsilon_i^2, \tag{22}$$

with weights obtained by matching derivatives w.r.t. X:

$$w_i \;=\; \frac{1}{\epsilon_i} \frac{\partial}{\partial \epsilon_i} \rho_\beta(\epsilon_i; \alpha^{(1)}, c^{(1)}). \tag{23}$$

With ρ_β defined in (16), straightforward differentiation yields

$$w_i = \frac{1}{(c^{(1)})^2} \left(\frac{(\epsilon_i / c^{(1)})^2}{|f-2|} + 1 \right)^{\frac{f}{2}-1}, \qquad f = f(\beta, \alpha^{(1)}). \tag{24}$$

For numerical robustness, it is often convenient to use the following special-case expressions when f is close to singular values (and otherwise use (24)):

$$w_i = \begin{cases} \frac{1}{(c^{(1)})^2}, & f = 2, \\ \frac{2}{\epsilon_i^2 + 2(c^{(1)})^2}, & f = 0, \\ \frac{1}{(c^{(1)})^2} \left(\frac{(\epsilon_i / c^{(1)})^2}{|f-2|} + 1 \right)^{\frac{f}{2}-1}, & \text{otherwise}, \end{cases} \tag{25}$$

and to clip w_i to $(0, 1]$ when required by the BR-duality constraint.

Optimizing (22) jointly over $(X, (w_i)_{i=1}^m)$ without regularization degenerates to $w_i = 0$ for all i. To prevent this, we exploit the Black–Rangarajan duality [Black and Rangarajan, 1996]. Under standard conditions (e.g., concavity of $\phi(z) := \rho_\beta(c^{(1)}\sqrt{z})$ in z), there exists an outlier-process penalty $\Phi_{\rho_\beta}(\cdot)$ such that

$$\rho_\beta(\epsilon) = \min_{w \in (0,1]} \frac{1}{2} w \epsilon^2 + \Phi_{\rho_\beta}(w). \tag{26}$$

Plugging (26) into (21) yields the BR-regularized formulation

$$\min_{X \in \mathbb{R},\, w_i \in (0,1]} \frac{1}{2m} \sum_{i=1}^{m} \Big(w_i \epsilon_i^2 + \Phi_{\rho_\beta}(w_i) \Big), \tag{27}$$

which matches (9) with the surrogate loss ρ_β. At fixed X, minimizing (27) over w_i recovers the IRLS update (23). For fixed weights, (22) admits a closed-form minimizer:

$$X \leftarrow \frac{\sum_{i=1}^m w_i X_i}{\sum_{i=1}^m w_i}. \tag{28}$$

The GNC continuation progressively changes β so that $f(\beta, \alpha^{(1)})$ moves from the quadratic regime ($f \approx 2$) to the target regime ($f \approx \alpha^{(1)}$).

Example 1 schedule ($\beta : \infty \to 1$). We initialize β_0 at a large value (e.g., $\beta_0 \gg 1$ so that $f_1(\beta_0, \alpha^{(1)}) \approx 2$) and decrease it geometrically toward 1:

$$\beta_k = 1 + \frac{\beta_{k-1} - 1}{\gamma}, \qquad \gamma > 1. \tag{29}$$

This update ensures $\beta_k \downarrow 1$ and hence $f_1(\beta_k, \alpha^{(1)}) \downarrow \alpha^{(1)}$.

Example 2/3 schedule ($\beta : 0^+ \to \infty$). We initialize β_0 small (e.g., $\beta_0 \ll 1$ so that $f_{2/3}(\beta_0, \alpha^{(1)}) \approx 2$) and increase it geometrically:

$$\beta_k = \gamma \beta_{k-1}, \qquad \gamma > 1, \tag{30}$$

so that $\beta_k \uparrow \infty$ and $f_{2/3}(\beta_k, \alpha^{(1)}) \to \alpha^{(1)}$.

We terminate the outer continuation once

$$\left| f(\beta, \alpha^{(1)}) - \alpha^{(1)} \right| \le \varepsilon_f, \tag{31}$$

or after a maximum number of continuation steps.

Algorithm 1 summarizes the full procedure. The method alternates IRLS updates of $(X, (w_i)_{i=1}^m)$ at a fixed β, and then updates β to increase nonconvexity. This yields a robust solver with improved basins of attraction compared to directly optimizing the original nonconvex loss.

We use geometric schedules (29) (Example 1) or (30) (Examples 2–3) with $\gamma \in [1.2, 2]$ in our experiments. For initialization, we set X to the sample mean (or median for additional robustness) and choose β_0 such that $f(\beta_0, \alpha^{(1)})$ is sufficiently close to 2 (e.g., $|f(\beta_0, \alpha^{(1)}) - 2| \le 10^{-2}$). Each inner iteration costs $\mathcal{O}(m)$ due to the closed-form updates (24) and (28). Hence the total cost is $\mathcal{O}(m\, T_{\max}\, K_{\max})$. The continuation parameter β controls the rate at which nonconvexity is injected. A faster schedule (larger γ in (29)–(30)) can accelerate convergence but may reduce robustness to poor initialization; conversely, a slower schedule yields more stable iterations with larger attraction basins at the expense of additional outer steps.

Algorithm 1 GNC-IRLS for solving (9) (Step-1 update).

Require: Measurements $(X_i)_{i=1}^m$; parameters $(\alpha^{(1)}, c^{(1)})$; mapping $f(\beta, \alpha^{(1)})$; schedule $\beta \leftarrow \mathcal{S}(\beta)$; tolerances $\varepsilon_X, \varepsilon_f$; iteration limits $T_{\max}$ (inner) and $K_{\max}$ (outer).

Ensure: $\widetilde{X}^{(1)}$.

$X \leftarrow \frac{1}{m}\sum_{i=1}^m X_i$ (or median); choose $\beta \leftarrow \beta_0$ such that $f(\beta_0, \alpha^{(1)}) \approx 2$.
for $k = 1, 2, \ldots, K_{\max}$ **do**
  $f \leftarrow f(\beta, \alpha^{(1)})$.
  for $t = 1, 2, \ldots, T_{\max}$ **do**
    $\epsilon_i \leftarrow X_i - X$, $w_i \leftarrow \frac{1}{\epsilon_i}\frac{\partial}{\partial \epsilon_i}\rho(\epsilon_i; f, c^{(1)})$, $w_i \leftarrow \min\{1, \max\{\delta, w_i\}\}$, $\forall i$
    $X_{\text{new}} \leftarrow \frac{\sum_{i=1}^m w_i X_i}{\sum_{i=1}^m w_i}$
    if $|X_{\text{new}} - X| \le \varepsilon_X$ **then**
      $X \leftarrow X_{\text{new}}$;
    end if
    $X \leftarrow X_{\text{new}}$.
  end for
  if $\left| f(\beta, \alpha^{(1)}) - \alpha^{(1)} \right| \le \varepsilon_f$ **then**
    break
  end if
  $\beta \leftarrow \mathcal{S}(\beta)$ *(increase nonconvexity).*
end for
return $\widetilde{X}^{(1)} \leftarrow X$.

G Technical Proofs

G.1 Proof of Lemma 3.1

Proof. We divide the analysis into two cases. **Case 1:** $\alpha < 2$**.** Set

$$y := 1 + \frac{(e/c)^2}{2-\alpha} \geq 1.$$

Then

$$\rho(e;\alpha,c) = \frac{2-\alpha}{\alpha}\big(y^{\alpha/2}-1\big).$$

A direct differentiation gives

$$\frac{\partial}{\partial\alpha}\rho(e;\alpha,c) = -\frac{2}{\alpha^2}\big(y^{\alpha/2}-1\big) + \frac{2-\alpha}{2\alpha}y^{\alpha/2}\log y + \frac{1}{2}y^{\alpha/2-1}(y-1).$$

Define

$$G(y) := -\frac{2}{\alpha^2}(y^{\alpha/2}-1) + \frac{2-\alpha}{2\alpha}y^{\alpha/2}\log y + \frac{1}{2}y^{\alpha/2-1}(y-1), \qquad y \geq 1.$$

Then

$$G'(y) = \frac{2-\alpha}{4}y^{\alpha/2-2}\big(y\log y - (y-1)\big).$$

Since $y\log y - (y-1) \geq 0$ for $y \geq 1$, we get $G'(y) \geq 0$. Also $G(1) = 0$. Hence $G(y) \geq 0$ for all $y \geq 1$, with strict inequality for $y > 1$. Because $y > 1$ is equivalent to $e \neq 0$, the claim follows.

Case 2: $\alpha > 2$**.** Set

$$y := 1 + \frac{(e/c)^2}{\alpha-2} \geq 1.$$

Now

$$\rho(e;\alpha,c) = \frac{\alpha-2}{\alpha}\big(y^{\alpha/2}-1\big),$$

and differentiation yields

$$\frac{\partial}{\partial\alpha}\rho(e;\alpha,c) = \frac{2}{\alpha^2}(y^{\alpha/2}-1) + \frac{\alpha-2}{2\alpha}y^{\alpha/2}\log y - \frac{1}{2}y^{\alpha/2-1}(y-1).$$

Define

$$F(y) := \frac{2}{\alpha^2}(y^{\alpha/2}-1) + \frac{\alpha-2}{2\alpha}y^{\alpha/2}\log y - \frac{1}{2}y^{\alpha/2-1}(y-1), \qquad y \geq 1.$$

Then

$$F'(y) = \frac{\alpha-2}{4}y^{\alpha/2-2}\big(1-y+y\log y\big).$$

Because $1 - y + y\log y \geq 0$ for $y \geq 1$, one gets $F'(y) \geq 0$ and again $F(1) = 0$. Hence $F(y) \geq 0$ for all $y \geq 1$, with strict inequality if $y > 1$. This proves the result. □

G.2 Basic properties of the score function

Fix $0 < \alpha \leq 1$ and set

$$\kappa_\alpha := 2 - \alpha \in [1, 2).$$

The adaptive loss and score are

$$\rho(e;\alpha,c) := \frac{\kappa_\alpha}{\alpha}\left[\Big(1 + \frac{(e/c)^2}{\kappa_\alpha}\Big)^{\alpha/2} - 1\right], \qquad \psi(e;\alpha,c) := \frac{e}{c^2}\Big(1 + \frac{(e/c)^2}{\kappa_\alpha}\Big)^{\alpha/2-1}.$$

For later use we introduce the explicit constants

$$L_\alpha := \sup_{z\geq 0} z\Big(1 + \frac{z^2}{\kappa_\alpha}\Big)^{\alpha/2-1} = \begin{cases} \big(\frac{1-\alpha}{2-\alpha}\big)^{(1-\alpha)/2}, & 0 < \alpha < 1, \\ 1, & \alpha = 1, \end{cases} \tag{32}$$

and

$$\lambda_\alpha := \frac{1}{\kappa_\alpha}\Big(1 + \frac{1}{\kappa_\alpha}\Big)^{\alpha/2-2} > 0. \tag{33}$$

The constant L_α is the sharp global bound for $c\,|\psi|$, and λ_α is an explicit lower curvature constant on $|e| \leq c$.

Lemma G.1 (Sharp score bounds)**.** *For every $0 < \alpha \le 1$, every $c > 0$, and every $e \in \mathbb{R}$,*

$$|\psi(e;\alpha,c)| \le \min\Big\{\frac{|e|}{c^2}, \frac{L_\alpha}{c}\Big\}, \tag{34}$$

$$|\psi'(e;\alpha,c)| \le \frac{1}{c^2}, \tag{35}$$

and if $|e| \le c$, then

$$\psi'(e;\alpha,c) \ge \frac{\lambda_\alpha}{c^2}. \tag{36}$$

Moreover,

$$c^2\psi(e;\alpha,c) \to e, \qquad c^2\psi'(e;\alpha,c) \to 1, \qquad c \to \infty, \tag{37}$$

for every fixed $e \in \mathbb{R}$.

Proof. Write $z := |e|/c$ and $u := z^2/\kappa_\alpha$. Then

$$|\psi(e;\alpha,c)| = \frac{z}{c}(1+u)^{-\kappa_\alpha/2}.$$

The bound $|\psi| \le |e|/c^2$ is immediate because $(1+u)^{-\kappa_\alpha/2} \le 1$. To obtain the sharp bounded-score constant, maximize

$$h_\alpha(z) := z\Big(1+\frac{z^2}{\kappa_\alpha}\Big)^{-\kappa_\alpha/2}, \qquad z \ge 0.$$

If $\alpha = 1$ then $\kappa_\alpha = 1$ and $h_1(z) = z(1+z^2)^{-1/2} \uparrow 1$, so $L_\alpha = 1$. If $0 < \alpha < 1$, then

$$h'_\alpha(z) = \Big(1+\frac{z^2}{\kappa_\alpha}\Big)^{-\kappa_\alpha/2-1}\Big(1-\frac{1-\alpha}{\kappa_\alpha}z^2\Big),$$

so the unique maximizer is $z_*^2 = \kappa_\alpha/(1-\alpha)$. Substituting gives exactly (32). This proves (34).

Next,

$$\psi'(e;\alpha,c) = \frac{1}{c^2}\Big(1+u\Big)^{\alpha/2-2}\Big(1+(\alpha-1)u\Big).$$

Since $|1+(\alpha-1)u| \le 1+u$ and $\alpha/2-2 \le -1$,

$$|\psi'(e;\alpha,c)| \le \frac{1}{c^2}(1+u)^{\alpha/2-1} \le \frac{1}{c^2},$$

which proves (35). If $|e| \le c$, then $u \le 1/\kappa_\alpha$, hence

$$1+(\alpha-1)u = 1-(1-\alpha)u \ge 1-\frac{1-\alpha}{\kappa_\alpha} = \frac{1}{\kappa_\alpha},$$

and because the exponent $\alpha/2-2$ is negative,

$$(1+u)^{\alpha/2-2} \ge \Big(1+\frac{1}{\kappa_\alpha}\Big)^{\alpha/2-2}.$$

Combining these yields (36) with λ_α from (33). Finally, (37) follows by dominated convergence from the explicit formulas, using $(1+u)^{-\kappa_\alpha/2} \le 1$ and $(1+u)^{\alpha/2-2}|1+(\alpha-1)u| \le 1$. □

G.3 Proof of Theorem 4.3

We split the proof into lemmas.

Lemma G.2 (Population slope and population root)**.** *For all sufficiently large m, there exists a unique θ_m solving $g_m(x) = 0$ in the interval $[\mu - r_m/4, \mu + r_m/4]$. Moreover,*

$$|\theta_m - \mu| \le \frac{2\sigma^2}{\lambda_\alpha c_m}. \tag{38}$$

In addition,

$$\sup_{|x-\mu|\le r_m/2} \big|c_m^2 g'_m(x) - 1\big| \to 0. \tag{39}$$

In particular, for all large m,

$$\inf_{|x-\mu|\le r_m/2} g'_m(x) \ge \frac{\lambda_\alpha}{2c_m^2}. \tag{40}$$

Proof. Because $\mathbb{E}(X_1 - \mu) = 0$,

$$g_m(\mu) = -\mathbb{E}\,\psi(X_1 - \mu; \alpha, c_m) = -\frac{1}{c_m^2}\mathbb{E}\Big[(X_1 - \mu)\Big\{\Big(1 + \frac{(X_1-\mu)^2}{\kappa_\alpha c_m^2}\Big)^{-\kappa_\alpha/2} - 1\Big\}\Big].$$

Using $|1 - (1+t)^{-\kappa_\alpha/2}| \le \min\{\frac{\kappa_\alpha}{2}t, 1\} \le \min\{t, 1\}$ and splitting at $|X_1 - \mu| \le c_m$, one gets

$$|g_m(\mu)| \le \frac{1}{c_m^2}\mathbb{E}\Big[|X_1 - \mu| \min\Big\{\frac{(X_1-\mu)^2}{\kappa_\alpha c_m^2}, 1\Big\}\Big] \le \frac{\sigma^2}{c_m^3}.$$

Next, by Lemma G.1,

$$g_m'(x) = \mathbb{E}\,\psi'(X_1 - x; \alpha, c_m).$$

For every fixed x, $c_m^2\psi'(X_1 - x; \alpha, c_m) \to 1$ almost surely and is bounded in absolute value by 1, so dominated convergence yields $c_m^2 g_m'(x) \to 1$. Since $r_m/c_m \to 0$ and ψ' is of the form $c_m^{-2}\phi'((X_1 - x)/c_m)$ with ϕ' continuously differentiable and bounded derivative, the convergence is uniform on $|x - \mu| \le r_m/2$. This proves (39) and hence (40) for large m.

Now apply the mean value theorem to g_m between μ and any root θ_m:

$$0 = g_m(\theta_m) = g_m(\mu) + (\theta_m - \mu)g_m'(\bar\theta_m)$$

for some $\bar\theta_m$ between μ and θ_m. Using (40) gives

$$|\theta_m - \mu| \le \frac{|g_m(\mu)|}{\inf_{|x-\mu|\le r_m/2} g_m'(x)} \le \frac{\sigma^2/c_m^3}{\lambda_\alpha/(2c_m^2)} = \frac{2\sigma^2}{\lambda_\alpha c_m}.$$

Since $c_m^{-1} = o(r_m)$, this shows $|\theta_m - \mu| \le r_m/4$ for all large m. Finally, strict monotonicity from (40) gives uniqueness of the population root in $[\mu - r_m/4, \mu + r_m/4]$. □

Lemma G.3 (Empirical monotonicity and existence of the central root)**.** *Under the assumptions of Theorem 4.3, there is an event $\mathscr{E}_m$ with $\mathbb{P}(\mathscr{E}_m) \to 1$ such that, on $\mathscr{E}_m$,*

$$\inf_{x \in I_m} \widehat{g}_m'(x) \ge \frac{\lambda_\alpha}{4c_m^2}, \tag{41}$$

and

$$|\widehat{g}_m(\theta_m)| \le \frac{\lambda_\alpha r_m}{16 c_m^2}. \tag{42}$$

Consequently, $\widehat{g}_m$ has a unique zero in I_m.

Proof. By (10) and (38), with probability tending to one the interval $[\theta_m - r_m/4, \theta_m + r_m/4]$ is contained in I_m. On the same event, $|x - \mu| \le r_m/2$ for all $x \in I_m$ once m is large.

For the slope, observe that

$$\widehat{g}_m'(x) - g_m'(x) = \frac{1}{m}\sum_{i=1}^m \Big\{\psi'(X_i - x; \alpha, c_m) - \mathbb{E}\psi'(X_1 - x; \alpha, c_m)\Big\}.$$

At any fixed x, the summands are bounded by c_m^{-2} in absolute value by (35), hence

$$\widehat{g}_m'(x) - g_m'(x) = O_p\Big(\frac{1}{c_m^2\sqrt{m}}\Big) = o_p(c_m^{-2}).$$

Because $r_m/c_m \to 0$ and $\psi''(e; \alpha, c) = c^{-3}\phi''(e/c)$ for a bounded ϕ'', the map $x \mapsto \widehat{g}_m'(x) - g_m'(x)$ is uniformly equicontinuous on I_m at scale $o(c_m^{-2})$. Combining this with (40) yields (41) on an event of probability tending to one.

Next,

$$\widehat{g}_m(\theta_m) = -\frac{1}{m}\sum_{i=1}^m \Big\{\psi(X_i - \theta_m; \alpha, c_m) - \mathbb{E}\psi(X_1 - \theta_m; \alpha, c_m)\Big\}.$$

Since $|\psi(e; \alpha, c_m)| \le |e|/c_m^2$ by (34),

$$\mathrm{Var}\big(\psi(X_1 - \theta_m; \alpha, c_m)\big) \le \frac{\mathbb{E}(X_1 - \theta_m)^2}{c_m^4} = O(c_m^{-4}),$$

therefore

$$\widehat{g}_m(\theta_m) = O_p\Big(\frac{1}{c_m^2\sqrt{m}}\Big).$$

Because $r_m\sqrt{m} \to \infty$, this implies (42) with probability tending to one.

Finally, on the intersection of these high-probability events,

$$\widehat{g}_m\Big(\theta_m + \frac{r_m}{4}\Big) \geq \widehat{g}_m(\theta_m) + \frac{r_m}{4} \inf_{x\in I_m} \widehat{g}'_m(x) \geq -\frac{\lambda_\alpha r_m}{16c_m^2} + \frac{\lambda_\alpha r_m}{16c_m^2} \geq 0,$$

and similarly $\widehat{g}_m(\theta_m - r_m/4) \leq 0$. Since $\widehat{g}_m$ is strictly increasing on I_m, it has exactly one zero there. □

Proof of Theorem 4.3. Lemma G.3 gives existence and uniqueness of the central root on an event $\mathscr{E}_m$ with probability tending to one. On $\mathscr{E}_m$, by the mean value theorem,

$$0 = \widehat{g}_m(\widetilde{X}_m) = \widehat{g}_m(\theta_m) + (\widetilde{X}_m - \theta_m)\widehat{g}'_m(\xi_m)$$

for some ξ_m between $\widetilde{X}_m$ and θ_m. Hence

$$\sqrt{m}(\widetilde{X}_m - \theta_m) = -\frac{c_m^2\widehat{g}_m(\theta_m)}{c_m^2\widehat{g}'_m(\xi_m)}\sqrt{m}. \tag{43}$$

Now

$$-\sqrt{m}\, c_m^2\widehat{g}_m(\theta_m) = \frac{1}{\sqrt{m}}\sum_{i=1}^m Z_{i,m}, \qquad Z_{i,m} := c_m^2\Big\{\psi(X_i-\theta_m;\alpha,c_m) - \mathbb{E}\psi(X_1-\theta_m;\alpha,c_m)\Big\}.$$

By (34),

$$|c_m^2\psi(X_i-\theta_m;\alpha,c_m)| \leq |X_i-\theta_m|,$$

so dominated convergence and (37) imply

$$Z_{i,m} \to X_i - \mu \quad \text{in } L^2, \qquad \mathrm{Var}(Z_{i,m}) \to \sigma^2.$$

The Lindeberg condition follows from the uniform square-integrability bound $|Z_{i,m}| \leq 2|X_i-\theta_m|$ and $\theta_m \to \mu$. Therefore the triangular-array CLT yields

$$\frac{1}{\sqrt{m}}\sum_{i=1}^m Z_{i,m} \Rightarrow N(0,\sigma^2).$$

On the other hand, Lemma G.2 and Lemma G.3 imply

$$c_m^2\widehat{g}'_m(\xi_m) \xrightarrow{p} 1.$$

Hence (43) and Slutsky give

$$\sqrt{m}(\widetilde{X}_m - \theta_m) \Rightarrow N(0,\sigma^2).$$

Finally, (38) gives

$$\sqrt{m}\,|\theta_m - \mu| \leq \frac{2\sigma^2\sqrt{m}}{\lambda_\alpha c_m} = \frac{2\sigma^2}{\lambda_\alpha}m^{-\gamma} \to 0,$$

so another application of Slutsky yields

$$\sqrt{m}(\widetilde{X}_m - \mu) \Rightarrow N(0,\sigma^2).$$

Consistency follows immediately. □

G.4 Proof of Theorem 4.4

Lemma G.4 (Population bias under $(1+\epsilon)$-moments)**.** *For all sufficiently large m, the population equation $g_m(x) = 0$ has a unique solution θ_m in $[\mu - r_m/4, \mu + r_m/4]$, and*

$$|\theta_m - \mu| \le \frac{2}{\lambda_\alpha} \frac{v^{1+\epsilon}}{c_m^\epsilon}. \tag{44}$$

Moreover,

$$\inf_{|x-\mu| \le r_m/2} g_m'(x) \ge \frac{\lambda_\alpha}{2c_m^2} \tag{45}$$

for all large m.

Proof. The proof is the heavy-tailed analogue of Lemma G.2. Using again

$$g_m(\mu) = -\frac{1}{c_m^2}\mathbb{E}\Big[(X_1-\mu)\Big\{\Big(1+\frac{(X_1-\mu)^2}{\kappa_\alpha c_m^2}\Big)^{-\kappa_\alpha/2} - 1\Big\}\Big],$$

and the elementary bound

$$\Big|1-(1+t)^{-\kappa_\alpha/2}\Big| \le \min\{t,1\}, \qquad t \ge 0,$$

we obtain

$$|g_m(\mu)| \le \frac{1}{c_m^2}\mathbb{E}\Big[|X_1-\mu| \min\Big\{\frac{(X_1-\mu)^2}{\kappa_\alpha c_m^2}, 1\Big\}\Big].$$

If $|X_1-\mu| \le c_m$, then $|X_1-\mu|^3/c_m^2 \le |X_1-\mu|^{1+\epsilon}/c_m^\epsilon$. If $|X_1-\mu| > c_m$, then $|X_1-\mu| \le |X_1-\mu|^{1+\epsilon}/c_m^\epsilon$. Hence

$$|g_m(\mu)| \le \frac{v^{1+\epsilon}}{c_m^{2+\epsilon}}.$$

As in the finite-variance case, dominated convergence plus (37) shows that $c_m^2 g_m'(x) \to 1$ uniformly on $|x-\mu| \le r_m/2$, because $r_m = o(c_m)$. Therefore (45) holds for large m, and the mean value theorem gives

$$|\theta_m - \mu| \le \frac{|g_m(\mu)|}{\inf_{|x-\mu|\le r_m/2} g_m'(x)} \le \frac{2}{\lambda_\alpha}\frac{v^{1+\epsilon}}{c_m^\epsilon}.$$

Since $c_m \to \infty$ and $r_m \to \infty$ with $r_m = o(c_m)$, the right-hand side is $o(r_m)$, so the unique population root lies in $[\mu - r_m/4, \mu + r_m/4]$ for all large m. □

Lemma G.5 (Variance and Bernstein control)**.** *Let θ_m be as in Lemma G.4. Then, for all sufficiently large m,*

$$\mathrm{Var}\big(\psi(X_1-\theta_m;\alpha,c_m)\big) \le \frac{2^{1+\epsilon}\max\{1,L_\alpha^2\}v^{1+\epsilon}}{c_m^{3+\epsilon}}. \tag{46}$$

Consequently,

$$\mathbb{P}\left(|\widehat{g}_m(\theta_m)| > \sqrt{\frac{2^{2+\epsilon}\max\{1,L_\alpha^2\}v^{1+\epsilon}t_\delta}{mc_m^{3+\epsilon}}} + \frac{4L_\alpha t_\delta}{3mc_m}\right) \le \delta/4. \tag{47}$$

Proof. By (34),

$$\psi(e;\alpha,c_m)^2 \le \min\Big\{\frac{e^2}{c_m^4}, \frac{L_\alpha^2}{c_m^2}\Big\}.$$

If $|e| \le c_m$, then $e^2/c_m^4 \le |e|^{1+\epsilon}/c_m^{3+\epsilon}$. If $|e| > c_m$, then $L_\alpha^2/c_m^2 \le L_\alpha^2|e|^{1+\epsilon}/c_m^{3+\epsilon}$. Hence

$$\psi(e;\alpha,c_m)^2 \le \frac{\max\{1,L_\alpha^2\}|e|^{1+\epsilon}}{c_m^{3+\epsilon}}.$$

Apply this with $e = X_1 - \theta_m$. Since

$$|X_1-\theta_m|^{1+\epsilon} \le 2^\epsilon\big(|X_1-\mu|^{1+\epsilon} + |\theta_m-\mu|^{1+\epsilon}\big),$$

and (44) implies $|\theta_m - \mu|^{1+\epsilon} \le v^{1+\epsilon}$ for all large m, we get

$$\mathbb{E}|X_1 - \theta_m|^{1+\epsilon} \le 2^{1+\epsilon} v^{1+\epsilon}$$

for all sufficiently large m. This proves (46).

Now set

$$Y_i := \psi(X_i - \theta_m; \alpha, c_m) - \mathbb{E}\psi(X_1 - \theta_m; \alpha, c_m).$$

Then $|Y_i| \le 2L_\alpha / c_m$ by (34), and $\mathrm{Var}(Y_i)$ is bounded by the right-hand side of (46). Bernstein's inequality therefore yields

$$\mathbb{P}\Big(\Big|\frac{1}{m}\sum_{i=1}^{m} Y_i\Big| > \sqrt{\frac{2\mathrm{Var}(Y_1)t_\delta}{m}} + \frac{2(2L_\alpha/c_m)t_\delta}{3m}\Big) \le 2e^{-t_\delta} \le \delta/4,$$

which is exactly (47) because $\widehat{g}_m(\theta_m) = -(1/m)\sum_i Y_i$. □

Proof of Theorem 4.4. By the same argument as in Lemma G.3, the pilot condition (11), the slope approximation (45), and the fact that $r_m/c_m \to 0$ imply that there is an event of probability at least $1 - \delta/4$ on which

$$\inf_{x \in I_m} \widehat{g}'_m(x) \ge \frac{\lambda_\alpha}{4c_m^2}$$

and $\widehat{g}_m$ has exactly one zero in I_m. On this event,

$$|\widetilde{X}_m - \theta_m| \le \frac{4c_m^2}{\lambda_\alpha} |\widehat{g}_m(\theta_m)|.$$

Combining this with (47) from Lemma G.5, we obtain with probability at least $1 - \delta/2$,

$$|\widetilde{X}_m - \theta_m| \le \frac{4c_m^2}{\lambda_\alpha}\left[\sqrt{\frac{2^{2+\epsilon}\max\{1, L_\alpha^2\} v^{1+\epsilon} t_\delta}{m c_m^{3+\epsilon}}} + \frac{4L_\alpha t_\delta}{3mc_m}\right].$$

Adding the bias bound (44) gives, with probability at least $1 - \delta$,

$$|\widetilde{X}_m - \mu| \le \frac{2}{\lambda_\alpha}\frac{v^{1+\epsilon}}{c_m^\epsilon} + \frac{2^{2+\epsilon/2}\max\{1, L_\alpha\}}{\lambda_\alpha}\sqrt{\frac{v^{1+\epsilon}t_\delta}{m} c_m^{1-\epsilon}} + \frac{8L_\alpha}{3\lambda_\alpha}\frac{c_m t_\delta}{m}.$$

Now insert the tuning

$$c_m = \tau\Big(\frac{v^{1+\epsilon} m}{t_\delta}\Big)^{1/(1+\epsilon)}.$$

Each term becomes a constant multiple of $(v^{1+\epsilon} t_\delta / m)^{\epsilon/(1+\epsilon)}$, namely

$$\frac{2}{\lambda_\alpha}\frac{v^{1+\epsilon}}{c_m^\epsilon} = \frac{2}{\lambda_\alpha}\tau^{-\epsilon}\Big(\frac{v^{1+\epsilon}t_\delta}{m}\Big)^{\epsilon/(1+\epsilon)},$$

$$\frac{2^{2+\epsilon/2}\max\{1, L_\alpha\}}{\lambda_\alpha}\sqrt{\frac{v^{1+\epsilon}t_\delta}{m} c_m^{1-\epsilon}} = \frac{2^{2+\epsilon/2}\max\{1, L_\alpha\}}{\lambda_\alpha}\tau^{(1-\epsilon)/2}\Big(\frac{v^{1+\epsilon}t_\delta}{m}\Big)^{\epsilon/(1+\epsilon)},$$

$$\frac{8L_\alpha}{3\lambda_\alpha}\frac{c_m t_\delta}{m} = \frac{8L_\alpha}{3\lambda_\alpha}\tau\Big(\frac{v^{1+\epsilon}t_\delta}{m}\Big)^{\epsilon/(1+\epsilon)}.$$

Collecting the coefficients gives exactly (12). □

H Proofs for the ARE results

H.1 Proof of Theorem 4.5

Proof. By assumption (iii),

$$\sqrt{k}\,(\widetilde{\mu} - q_m) \Rightarrow N\left(0, \frac{1}{4f_m(q_m)^2}\right).$$

Using assumption (ii),

$$\frac{1}{4f_m(q_m)^2} = \frac{\pi\sigma^2}{2m}\,(1+o(1)).$$

Therefore

$$\sqrt{n}\,(\widetilde{\mu}-q_m) = \sqrt{mk}\,(\widetilde{\mu}-q_m) \Rightarrow N\Big(0, \frac{\pi}{2}\sigma^2\Big),$$

because $n = mk + o(n)$. Now assumption (i) gives $\sqrt{n}\,(q_m - \mu) \to 0$. Slutsky's theorem yields

$$\sqrt{n}\,(\widetilde{\mu}-\mu) = \sqrt{n}\,(\widetilde{\mu}-q_m) + \sqrt{n}\,(q_m-\mu) \Rightarrow N\Big(0, \frac{\pi}{2}\sigma^2\Big).$$

The asymptotic relative efficiency is then

$$\mathrm{ARE}(\widetilde{\mu}, \bar{X}_n) = \frac{\sigma^2}{(\pi/2)\sigma^2} = \frac{2}{\pi}.$$

□

Remark H.1. A convenient sufficient condition for assumption (i) is

$$q_m - \mu = O(m^{-1/2-\gamma}) \qquad \text{and} \qquad k = o(m^{2\gamma}).$$

Indeed, then $\sqrt{n}\,|q_m-\mu| \lesssim \sqrt{k}m^{-\gamma} \to 0$. This is the condition naturally suggested by the finite-variance tuning $c_m = m^{1/2+\gamma}$.

H.2 Proof of Theorem 4.6

Proof. Apply Theorem 4.4 to each block with confidence level $\delta_0 = 1/4$. Since

$$t_{\delta_0} = \log\frac{4}{\delta_0} = \log 16,$$

we get, for all sufficiently large m,

$$\mathbb{P}\Big(|\widetilde{X}_{j,m}-\mu| \le r_m\Big) \ge \frac{3}{4}, \qquad r_m := C_{\alpha,\epsilon}(\tau)\Big(\frac{v^{1+\epsilon}\log 16}{m}\Big)^{\epsilon/(1+\epsilon)}.$$

The block estimators are independent because they are computed from disjoint blocks. Therefore the median-boosting lemma gives

$$\mathbb{P}\big(|\widetilde{\mu}-\mu| > r_m\big) \le e^{-k/8} \le \frac{\delta}{2} < \delta.$$

It remains to rewrite r_m in terms of n and δ. Since $k = \lceil 8\log(2/\delta)\rceil \le 16\log(2/\delta)$ and $m = \lfloor n/k\rfloor \ge n/(2k)$ for all sufficiently large n,

$$\frac{1}{m} \le \frac{2k}{n} \le \frac{32\log(2/\delta)}{n}.$$

Hence

$$r_m \le C_{\alpha,\epsilon}(\tau)\Big(\frac{32\,v^{1+\epsilon}\log 16\,\log(2/\delta)}{n}\Big)^{\epsilon/(1+\epsilon)} = C^{\mathrm{ARE}}_{\alpha,\epsilon}(\tau)\Big(\frac{v^{1+\epsilon}\log(2/\delta)}{n}\Big)^{\epsilon/(1+\epsilon)},$$

with $C^{\mathrm{ARE}}_{\alpha,\epsilon}(\tau)$ defined in (13). This proves the claim. □

I Some additional experimental details

I.1 Single-Agent on mathematical reasoning

We fine-tune Qwen3-4B-Base using reinforcement learning within the verl framework, adopting Dr.GRPO and REINFORCE++ as the advantage estimators and a reward model for supervision. Training is conducted on the MATH training set. We set the training batch size to 840, the maximum prompt and response lengths to 512 and 1024, respectively, and the actor learning rate to 1e-6. For each prompt, 8 responses are sampled, and rewards are aggregated using a robust mean estimator with $\tau = 3$. A KL regularization term with coefficient 0.001 is applied to stabilize training. The reward model is InternLM2-1.8B-Reward. All experiments are conducted on a single node with 6 A800 GPUs for 10 epochs.

Regarding the influence of some hyperparameters in the algorithm, such as the number of rolls and the value of tau, the experimental results are shown in the Figure 10.

Table 11: Training hyperparameters for dual-agent ILR training.

Hyperparameter	Value	Description
Learning rate (Agent 1)	1×10^{-6}	Adam optimizer
Learning rate (Agent 2)	1×10^{-6}	Adam optimizer
Sampling temperature	0.5	—
Training batch size	32	—
Number of episodes	10	—
Max prompt length	6144	Tokens
Max generation length	1536	Tokens
Top-p sampling	0.95	Nucleus sampling
Discount factor γ	1.0	—
KL estimator	K2	—
Advantage estimator	ILR	—
Precision	BF16	Mixed precision

Table 12: Reward noise hyperparameters (Sparse Cauchy Outlier Noise).

Parameter	Symbol	Value	Description
Noise type	—	Outlier	Sparse one-sided heavy-tail
Group contamination prob.	p	0.2	Probability of corrupting a group
Cauchy scale	γ	1.0	Controls distribution width
Spike scale	α	10.0	Noise magnitude scaling factor
Spike clip	$\Delta_{\max}$	10.0	Maximum absolute noise value
Noise target	—	Argmax	Inject noise to group maximum

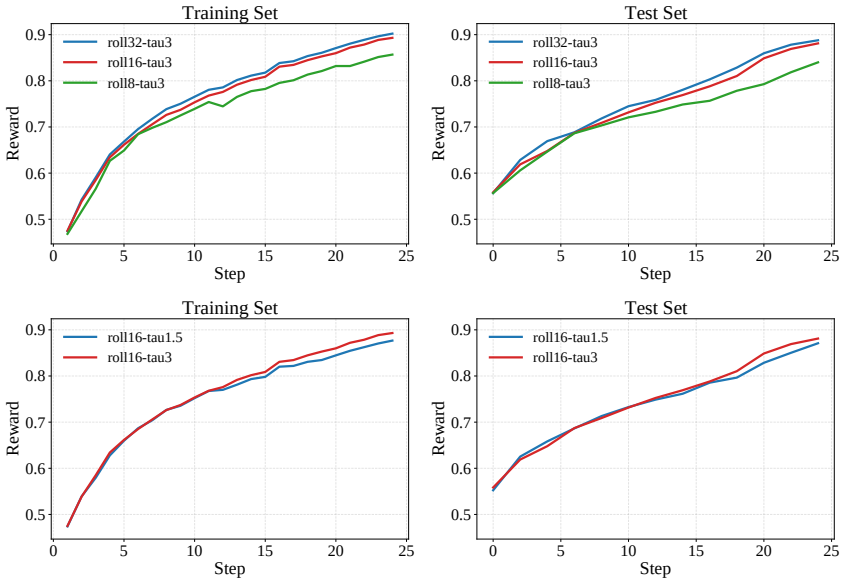


Figure 10: Training the MATH dataset using the Dr.GRPO algorithm, and the influence of different parameters on the training outcome.

I.2 Multi-Agent DACR

Regarding the interaction mode of the DACR we have designed, experiments are conducted on a single node with 8 A800 GPUs. We use Ray for distributed training orchestration and vLLM for efficient inference with 4 inference engines and tensor parallelism of 2. Gradient checkpointing and Adam parameter offloading are enabled to optimize GPU memory usage. The specific parameters are shown in Table 11.

To simulate reward uncertainty in realistic scenarios, we introduce a sparse one-sided heavy-tail spike noise based on the Cauchy distribution. The specific parameters are shown in Table 12.

I.3 Multi-Agent ReMA

The two figures on the left are related to our ReMA interaction model, namely the "Thinker-Reasoner" interaction model. The hyperparameters we used in training the two Agents are shown in Table 13.

Table 13: Hyperparameters for REINFORCE++ with ReMA.

Hyperparameter	**Value**
Model	
Policy model	Qwen3-1.7B-Base
Process reward model	Skywork-o1-Open-PRM-Qwen-2.5-1.5B
RL Algorithm	
Advantage estimator	REINFORCE++
Mean baseline estimator	GNC-Adapt-Mom
GNC max iterations	50
GNC τ_{scale}	3.0
GNC convergence threshold	10^{-6}
Clip mode	Turn
Aggregation mode	Trajectory
Training	
Learning rate	1×10^{-6}
Training batch size	6
PPO mini-batch size	6
Micro-batch size per GPU	1
Rollout samples per prompt (n)	8
Max prompt length	4096
Max response length	2048
KL loss	Disabled
Entropy coefficient	0
Total epochs	10
Validation frequency	Every 10 steps
Reward	
Training reward	Skywork PRM (last aggregation)
Validation reward	ReMA
Mask unfinished reward	True

I.4 Additional mathematical reasoning experiments

The experiments below supplement the three-seed results in Tables 1 and 3. The rollout, entropy, and centering comparisons use the Qwen3-4B-Base Dr.GRPO setting and report point estimates for each configuration. Unless varied below, the rollout count is eight responses per prompt in both single-agent and multi-agent experiments.

Rollout count. Table 14 compares 8, 16, and 32 responses per prompt. ARE has higher accuracy on four of the five benchmarks at both 16 and 32 rollouts. AIME2024 is the exception: the baseline reaches 14.17% and 15.21%, compared with 10.00% and 10.83% for ARE. On MinervaMath, ARE obtains 21.69% at all three rollout counts, compared with 12.87%, 13.97%, and 11.40% for the baseline.

Table 14: Accuracy (%) as the number of responses per prompt varies. Base uses empirical-mean centering.

Rollouts	MATH500		AIME2024		AIME2025		MinervaMath		Gaokao2023	
	Base	ARE	Base	ARE	Base	ARE	Base	ARE	Base	ARE
8	73.04	73.24	11.67	12.08	9.259	10.42	12.87	21.69	6.234	14.29
16	72.03	72.40	14.17	10.00	8.333	8.958	13.97	21.69	6.494	15.07
32	70.42	73.20	15.21	10.83	9.375	9.491	11.40	21.69	4.675	15.07

Entropy regularization. Table 15 varies the entropy coefficient from 0 to 0.01. At 0.001, AIME2024 accuracy is 15.21% with ARE and 13.54% with the baseline. At 0.01, ARE has lower AIME2025 accuracy (5.324% versus 7.176%). Increasing the coefficient from 0 to 0.01 reduces MATH500 accuracy for both methods, from 73.04% to 66.60% for the baseline and from 73.24% to 66.80% for ARE.

Table 15: Accuracy (%) as the entropy regularization coefficient varies.

Coefficient	MATH500		AIME2024		AIME2025		MinervaMath		Gaokao2023	
	Base	ARE	Base	ARE	Base	ARE	Base	ARE	Base	ARE
0	73.04	73.24	11.67	12.08	9.259	10.42	12.87	21.69	6.234	14.29
0.001	72.03	72.44	13.54	15.21	7.870	8.102	12.50	12.87	4.935	5.195
0.01	66.60	66.80	11.04	13.13	7.176	5.324	8.456	9.191	2.857	3.117

Reward distribution at larger rollout counts. Table 16 reports reward distribution statistics at each rollout count. Skewness changes from -0.482 at 8 rollouts to -0.568 at 32 rollouts. Excess kurtosis changes from -0.852 to -0.742, while Moors kurtosis changes from 1.057 to 1.086. The reward samples remain asymmetric at all three rollout counts.

Table 16: Reward distribution statistics at different rollout counts.

Rollouts	Skewness	Excess kurtosis	Moors kurtosis
8	-0.482	-0.852	1.057
16	-0.518	-0.805	1.075
32	-0.568	-0.742	1.086

Median-centering baseline. Table 17 compares empirical-mean centering, sample-median centering, and ARE. ARE has the highest accuracy on four of the five benchmarks. Median centering has the highest AIME2024 accuracy, at 13.33%, compared with 12.08% for ARE. On MinervaMath, accuracy is 12.87% with the mean, 18.01% with the median, and 21.69% with ARE.

Table 17: Accuracy (%) with empirical-mean, sample-median, and ARE reward centering.

Center	MATH500	AIME2024	AIME2025	MinervaMath	Gaokao2023
Mean	73.04	11.67	9.259	12.87	6.234
Median	72.03	13.33	9.259	18.01	12.21
ARE	73.24	12.08	10.42	21.69	14.29

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's motivation, method, theoretical claims, and empirical scope, and these claims are supported by the sections on Theoretical Properties and the single-agent and multi-agent experiments.

   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: The paper discusses limitations in the conclusion and Limitations section, including computational overhead, adaptive parameterization, and the fact that the embodied VLN study validates the estimator component rather than the full multi-agent interaction protocol.

   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: Assumptions and theorem statements are provided in the Theoretical Properties section, and complete proofs are included in the appendix sections Technical Proofs and Proofs for the ARE results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper describes datasets, models, reward settings, collaboration protocols, and additional implementation details in the experimental sections, DACR Interaction Framework, and Some additional experimental details.

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: [No]

   Justification: This submission does not yet provide anonymized code or a packaged data release; instead, it documents the setup and datasets used in the paper and appendix.

   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: The experimental sections and appendix specify datasets, evaluation benchmarks, batch sizes, learning rates, sequence lengths, rollout settings, and multi-agent hyperparameter tables needed to understand the reported results.

   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: Table 1 reports mean accuracy and standard deviation over three random seeds for Qwen3-4B-Base and OLMo3-7B-Instruct; Table 3 reports the corresponding ReMA results. Standard deviations measure variation across training runs; they are not confidence intervals or significance tests. The VLN and DACR benchmark tables retain point estimates.

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: [No]

   Justification: The paper reports GPU types and several hardware settings, but it does not yet provide runtime or full compute-cost accounting for every experiment.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The study focuses on algorithmic training and evaluation using existing models and benchmarks, and we are not aware of any aspect that conflicts with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The Broader Impacts section discusses both positive uses, such as more reliable robust post-training, and negative risks, such as misuse in autonomous or content-generation systems and increased compute costs.

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: This submission does not release a new high-risk model or scraped dataset, so no dedicated safeguard mechanism is part of the present paper.

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: [No]

Justification: The paper cites the original datasets and models it uses, but it does not yet systematically enumerate the licenses and terms of use for each external asset.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not introduce or release a new dataset, codebase, or model checkpoint as an asset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [N/A]

Justification: The reported experiments do not involve crowdsourcing or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing or research with human subjects, so IRB approval is not applicable.

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: [Yes]

Justification: LLMs and VLMs are core experimental components of the research, and the paper explicitly describes the use of Qwen3-4B-Base, Qwen2-VL-2B-Instruct, reward models, and multi-agent collaborative training protocols in the methodology and experiments.

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.