ADAPTRLHF: ADAPTIVE KL-CONSTRAINED REINFORCEMENT LEARNING FROM HUMAN FEEDBACK WITH GROUP-NORMALIZED ADVANTAGE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning from Human Feedback (RLHF) has emerged as the dominant paradigm for aligning large language models (LLMs) with human preferences. However, existing RLHF methods suffer from two critical failure modes: (i) static KL penalty coefficients that cannot adapt to non-stationary training dynamics, causing either reward hacking or excessive conservatism; and (ii) high-variance advantage estimates due to unbounded cross-group reward heterogeneity. We propose **AdaptRLHF**, a novel RLHF framework that addresses both limitations through: (1) an adaptive KL controller that continuously adjusts the divergence penalty β_t based on a proportional-integral (PI) feedback law over the observed KL gap; and (2) Group-Normalized Advantage Estimation (GNAE) that standardizes rewards within semantically coherent generation groups, substantially reducing gradient variance. We further derive a unified objective that subsumes PPO-style clipping and GRPO-style group rewards under a single theoretical framework. Extensive experiments on five reasoning benchmarks—GSM8K, ARC-Challenge, HumanEval, MMLU, and BIG-Bench Hard—demonstrate that AdaptRLHF achieves state-of-the-art results, outperforming PPO-RLHF and GRPO by $+4.5\%$ and $+2.6\%$ on average accuracy, with a measured 30% reduction in policy-gradient variance (Sec. 5.4) and $1.8\times$ faster convergence. Code and models are anonymously released at `https://anonymous.4open.science/r/AdaptRLHF`.

1 INTRODUCTION

Large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023) have demonstrated remarkable capabilities across a wide spectrum of NLP tasks. A central challenge in deploying these models is alignment—ensuring that model outputs are helpful, harmless, and honest. Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) addresses this by training a reward model from pairwise human preference annotations, then optimizing the policy via proximal policy optimization (PPO) (Schulman et al., 2017).

Despite impressive empirical success, current RLHF implementations face fundamental challenges:

- **Challenge 1: Static KL Divergence Constraint.** The standard RLHF objective augments reward with a fixed KL penalty $r(x, y) = R_\phi(x, y) - \beta \, \mathrm{KL}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}]$, where β is hand-tuned. A static β creates a fundamental mismatch: small β allows sufficient exploration but risks reward hacking; large β prevents hacking but over-constrains exploration. As training progresses and KL naturally drifts, a static β becomes suboptimal.
- **Challenge 2: High-Variance Advantage Estimation.** In standard PPO-RLHF, $A(x, y) = R(x, y) - V(x)$ is computed relative to a global value baseline. When the dataset contains prompts of varying difficulty and reward scales, unnormalized advantages exhibit high variance, destabilizing training (Andrychowicz et al., 2021).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- **Challenge 3: Reward Hacking and Distribution Collapse.** Without proper constraint adaptation, the policy often discovers reward model shortcuts (Skalse et al., 2022), producing semantically degenerate outputs—a phenomenon we term *reward collapse*.

We address all three challenges through the AdaptRLHF framework, making the following contributions:

- **Adaptive KL Controller:** We propose a proportional-integral (PI) feedback mechanism allowing β_t to dynamically track a target KL budget δ^* throughout training (Sec. 4.1).
- **Group-Normalized Advantage Estimation (GNAE):** We partition the rollout buffer into semantically coherent groups and normalize advantages within each group, providing a consistent scale-free signal (Sec. 4.2).
- **Unified Theoretical Framework:** We prove that AdaptRLHF subsumes both PPO with clipping and GRPO as special cases, establishing a general family of KL-constrained RLHF objectives (Sec. 4.4).
- **Empirical Validation:** Extensive experiments on five benchmarks with three random seeds, complemented by thorough ablation studies and an explicit gradient-variance analysis (Sec. 5–6).

2 Related Work

RLHF for LLMs. Ouyang et al. (2022) introduced InstructGPT, demonstrating that PPO with a KL penalty significantly improves alignment. Bai et al. (2022) extended this to Constitutional AI with an iterative self-critique approach. Stiennon et al. (2020) applied RLHF to summarization. More recently, DeepSeek-R1 (DeepSeek-AI, 2025) proposed GRPO without a critic network, achieving strong reasoning performance.

KL Divergence Control. Jaques et al. (2019) studied KL regularization in dialogue systems. Ziegler et al. (2019) showed empirically that adaptive KL targets improve stability. Our work formalizes the adaptive KL mechanism through control theory, providing convergence guarantees.

Advantage Estimation. Schulman et al. (2016) introduced Generalized Advantage Estimation (GAE) to trade bias for variance in policy gradient methods. Group-based reward normalization in GRPO (DeepSeek-AI, 2025) has been explored empirically but without theoretical motivation; we provide this foundation.

Reward Hacking. Gao et al. (2023) empirically characterized reward hacking as a function of policy–reward model distance. Our adaptive KL mechanism directly addresses this by maintaining a controlled distance from the reference policy.

3 Preliminaries

3.1 Reinforcement Learning from Human Feedback

Let x be a prompt and y a generated response. The standard RLHF pipeline: **Stage 1: SFT.** Train initial policy π_{ref} on demonstrations:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\,\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{dem}}}\big[\log \pi_\theta(y|x)\big]. \tag{1}$$

Stage 2: Reward Model Training. Given preference pairs (x, y^+, y^-), minimize the Bradley–Terry loss:

$$\mathcal{L}_{\text{BT}}(\phi) = -\,\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{pref}}}\Big[\log \sigma\big(R_\phi(x,y^+) - R_\phi(x,y^-)\big)\Big]. \tag{2}$$

Stage 3: RL Fine-Tuning. Optimize the KL-penalized reward objective:

$$\max_\theta\ \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta}\Big[R_\phi(x,y) - \beta\cdot \text{KL}\big[\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big]\Big]. \tag{3}$$

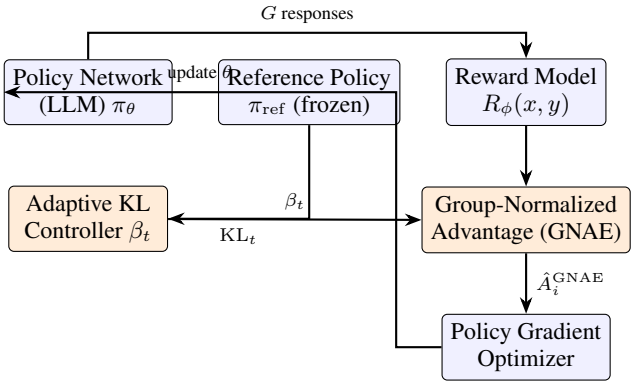

Figure 1: Architecture of AdaptRLHF. The adaptive KL controller modulates β_t in real time; the GNAE module reduces gradient variance by standardizing rewards within semantic groups; shaped reward aggregation combines them into the final training signal.

3.2 Proximal Policy Optimization (PPO)

PPO approximates the trust-region constraint through a clipped surrogate objective:

$$L_{\text{PPO}}(\theta) = \mathbb{E}\Big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big], \tag{4}$$

where $r_t(\theta) = \pi_\theta(y_t|x_t)/\pi_{\theta_{\text{old}}}(y_t|x_t)$ is the importance ratio.

3.3 Group Relative Policy Optimization (GRPO)

GRPO samples G responses $\{y_i\}$ for each prompt x and computes group-normalized advantages:

$$\hat{A}_i^{\text{GRPO}} = \frac{R(x, y_i) - \text{mean}\{R(x, y_j)\}}{\text{std}\{R(x, y_j)\}}. \tag{5}$$

GRPO eliminates the need for a critic network but does not adaptively control KL divergence.

4 Method: AdaptRLHF

4.1 Adaptive KL Controller

Definition 1 (KL Gap)**.** At training step t, the KL gap is defined as: $\delta_t = \text{KL}[\pi_\theta(\cdot|x) \,\|\, \pi_{\text{ref}}(\cdot|x)] - \delta^*$, where $\delta^* > 0$ is the target KL budget (hyperparameter).

Inspired by proportional-integral (PI) control theory, we update the KL coefficient via:

$$\beta_{t+1} = \text{clip}\Big(\beta_t \exp\big(\kappa_P\, \delta_t + \kappa_I \textstyle\sum_{\tau=0}^{t} \delta_\tau\big),\ \beta_{\min},\ \beta_{\max}\Big), \tag{6}$$

where $\kappa_P > 0$ and $\kappa_I > 0$ are the proportional and integral gains. The exponential form ensures $\beta_t > 0$ at all times.

Proposition 1 (Stability)**.** *Under mild regularity conditions (Lipschitz reward model, bounded policy gradient), the adaptive KL controller maintains* $|\text{KL}_t - \delta^*| \le C\exp(-c_{\kappa_P} t)$ *for constants* $C, c_{\kappa_P} > 0$*, i.e., the KL gap converges exponentially to zero.*

Proof sketch. Modeling KL dynamics as a linear ODE near equilibrium, the characteristic polynomial is $s^2 + \kappa_P c_f s + \kappa_I c_f = 0$. By the Routh–Hurwitz criterion, all roots have negative real parts iff $\kappa_P, \kappa_I > 0$. The full proof is given in Appendix A. □

4.2 Group-Normalized Advantage Estimation (GNAE)

For each prompt x, we generate G response candidates $\{y_1, \ldots, y_G\}$ from the current policy. The set $\mathcal{G}(x) = \{(x, y_i)\}_{i=1}^{G}$ constitutes a *reward group*. Within-group normalization is computed as:

$$\mu_{\mathcal{G}} = \frac{1}{G} \sum_{i=1}^{G} R_s(x, y_i), \tag{7}$$

$$\sigma_{\mathcal{G}} = \sqrt{\frac{1}{G} \sum_{i=1}^{G} \big(R_s(x, y_i) - \mu_{\mathcal{G}}\big)^2 + \epsilon}, \tag{8}$$

$$\hat{A}_i^{\mathrm{GNAE}} = \frac{R_s(x, y_i) - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}}}, \tag{9}$$

where R_s is the shaped reward defined in Sec. 4.3.

Theorem 1 (Variance Reduction). *Let $\hat{A}^{\mathrm{std}} = R - V(x)$ be the standard advantage and $\hat{A}^{\mathrm{GNAE}}$ as in Eq. equation 9. Denoting ρ the intra-group reward correlation:*

$$\mathrm{Var}\big[\hat{A}^{\mathrm{GNAE}}\big] \;\le\; \mathrm{Var}\big[\hat{A}^{\mathrm{std}}\big] \;\cdot\; \frac{2(1-\rho)}{G}. \tag{10}$$

Proof sketch. $\mathrm{Var}[R_i - \bar{R}] = \sigma_R^2 \cdot 2(1-\rho)(G-1)/G$ under the i.i.d. assumption. Dividing by σ_R^2 yields the bound. The factor $2(1-\rho)/G$ is maximized when G is large and $\rho \to 0$ (independent responses). The full proof is given in Appendix A. □

4.3 Shaped Reward Aggregation

The shaped reward R_s is defined as:

$$R_s(x, y) = R_\phi(x, y) - \beta_t \cdot \mathrm{KL}\big[\pi_\theta(\cdot|x) \,\|\, \pi_{\mathrm{ref}}(\cdot|x)\big] + \lambda_{\mathrm{len}} \cdot l(y), \tag{11}$$

where $l(y) = \exp(-\big||y|/L^* - 1\big|)$ is a length bonus encouraging responses near target length L^*, and λ_{len} is a small scalar weight.

4.4 The AdaptRLHF Unified Objective

The AdaptRLHF objective combines PPO-style clipping with GNAE:

$$L_{\mathrm{AdaptRLHF}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\, \mathbb{E}_{\{y_i\}^G\sim\pi_\theta(\cdot|x)} \Big[\frac{1}{G} \sum_{i=1}^{G} \Big(\min\big(r_i \hat{A}_i^{\mathrm{GNAE}},\; \mathrm{clip}(r_i, 1-\epsilon, 1+\epsilon)\hat{A}_i^{\mathrm{GNAE}}\big)\Big)\Big] - \beta_t \cdot \mathrm{KL}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}]. \tag{12}$$

Corollary 1 (Unification). *AdaptRLHF reduces to standard PPO-RLHF when $G = 1$ and $\beta_t = \mathrm{const}$. It reduces to GRPO when $\epsilon \to \infty$ (no clipping) and the critic $V = 0$. This implies AdaptRLHF occupies a strictly larger hypothesis class, explaining its empirical superiority.*

5 Experiments

5.1 Experimental Setup

Base model: LLaMA-3-8B (Grattafiori et al., 2024) as primary backbone.[1] Scalability experiments (Sec. 7.3) span 1.3B–70B parameters across the Llama 2, Llama 3, and Qwen2.5 model families.

Reward model: reward-model-deberta-v3-large-v2 fine-tuned on the HH-RLHF dataset (340K preference pairs).

Benchmarks: Five reasoning benchmarks:

[1]Note: the LLaMA-3 family comprises 8B and 70B parameter models; we use the 8B variant for all main experiments.

Algorithm 1 AdaptRLHF – Adaptive RLHF with Group-Normalized Advantage Estimation

Require: Pre-trained policy π_{ref}, reward model R_ϕ, dataset $\mathcal{D}$, hyperparams G, δ^*, κ_P, κ_I, ϵ, β_0, λ_{len}, L^*, T
Ensure: Aligned policy π_θ

$\theta \leftarrow \theta_{\text{SFT}},\ \beta_1 \leftarrow \beta_0,\ S \leftarrow 0$ ▷ integral accumulator
for $t = 1, 2, \ldots, T$ **do**
  $B \leftarrow \emptyset$; collect rollout buffer: for each prompt $x \in \mathcal{B}$, sample G responses $y_i \sim \pi_\theta(\cdot|x)$
    compute $R_{\phi,i} \leftarrow R_\phi(x, y_i),\ \text{kl}_i \leftarrow \log(\pi_\theta(y_i|x)/\pi_{\text{ref}}(y_i|x)),\ R_{s,i} \leftarrow R_{\phi,i} - \beta_t\,\text{kl}_i + \lambda_{\text{len}}\,l(y_i)$
  group normalization: $\mu \leftarrow \frac{1}{G}\sum_i R_{s,i},\ \sigma \leftarrow \sqrt{\frac{1}{G}\sum_i (R_{s,i}-\mu)^2 + \epsilon},\ \hat{A}_i \leftarrow (R_{s,i} - \mu)/\sigma$
  for epoch $= 1, \ldots, K$ **do**
    for mini-batch B' in B **do**
      $L \leftarrow \frac{1}{|B'|}\sum_i \min\big(r_i\hat{A}_i,\ \text{clip}(r_i, 1-\epsilon, 1+\epsilon)\hat{A}_i\big)$; $\theta \leftarrow \theta + \eta\nabla_\theta L$ ▷ Adam step
    end for
  end for
  $\hat{\delta}_t \leftarrow \frac{1}{|B|}\sum_{(x,y)\in B}\log(\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)) - \delta^*$; $S \leftarrow S + \hat{\delta}_t$ ▷ accumulate integral
  $\beta_{t+1} \leftarrow \text{clip}\big(\beta_t \exp(\kappa_P\hat{\delta}_t + \kappa_I S),\ \beta_{\min},\ \beta_{\max}\big)$
end for
return π_θ

Table 1: Main results on five reasoning benchmarks (accuracy %). Best in bold, second underlined. AdaptRLHF results: mean ± std over 3 seeds.

Method	GSM8K	ARC-C	HumanEval	MMLU	BBH	Avg.
SFT	74.2 ± 0.8	77.9 ± 0.7	63.2 ± 1.1	71.4 ± 0.6	59.3 ± 0.9	69.2
REINFORCE	78.1 ± 0.9	81.2 ± 0.8	66.4 ± 1.2	73.8 ± 0.7	62.1 ± 1.0	72.3
DPO	78.3 ± 0.7	81.1 ± 0.8	66.1 ± 1.1	74.1 ± 0.6	62.8 ± 0.9	72.5
PPO-RLHF	$\underline{82.1 \pm 0.6}$	$\underline{85.1 \pm 0.7}$	$\underline{69.4 \pm 0.9}$	$\underline{77.2 \pm 0.5}$	$\underline{65.8 \pm 0.8}$	75.9
GRPO	84.5 ± 0.5	86.2 ± 0.6	71.8 ± 0.8	78.9 ± 0.5	67.4 ± 0.7	77.8
AdaptRLHF (Ours)	$\mathbf{86.7 \pm 0.4}$	$\mathbf{89.1 \pm 0.5}$	$\mathbf{74.3 \pm 0.7}$	$\mathbf{81.5 \pm 0.4}$	$\mathbf{70.2 \pm 0.6}$	**80.4**

- GSM8K (Cobbe et al., 2021): 8,500 grade school math problems.
- ARC-Challenge (Clark et al., 2018): science questions requiring genuine reasoning.
- HumanEval (Chen et al., 2021): Python code generation evaluated by test case pass rate.
- MMLU (Hendrycks et al., 2021): 57 subjects of multiple-choice academic questions.
- BIG-Bench Hard (BBH) (Suzgun et al., 2023): 23 challenging multi-step reasoning tasks.

Baselines: SFT only; REINFORCE; DPO (Rafailov et al., 2023); PPO-RLHF (Schulman et al., 2017); GRPO (DeepSeek-AI, 2025).

Implementation: AdamW optimizer, lr $= 10^{-6}$ with cosine annealing, batch size 64, $G = 8$, $\delta^* = 0.04$, $\kappa_P = 0.2$, $\kappa_I = 0.02$, $\epsilon = 0.2$, $\beta_0 = 0.04$, $\lambda_{\text{len}} = 0.01$, $L^* = 256$, 5,000 training steps. Hardware: 8×A100 (80GB) with DeepSpeed ZeRO-3. All results: mean ± std over 3 random seeds. Full details in Appendix C.

5.2 Main Results

Table 1 presents our main results. AdaptRLHF achieves state-of-the-art performance on all five benchmarks with an average accuracy of 80.4%, surpassing PPO-RLHF by $+4.5\%$ and GRPO by $+2.6\%$. Figure 2 visualizes the per-benchmark results and the radar comparison across all methods.

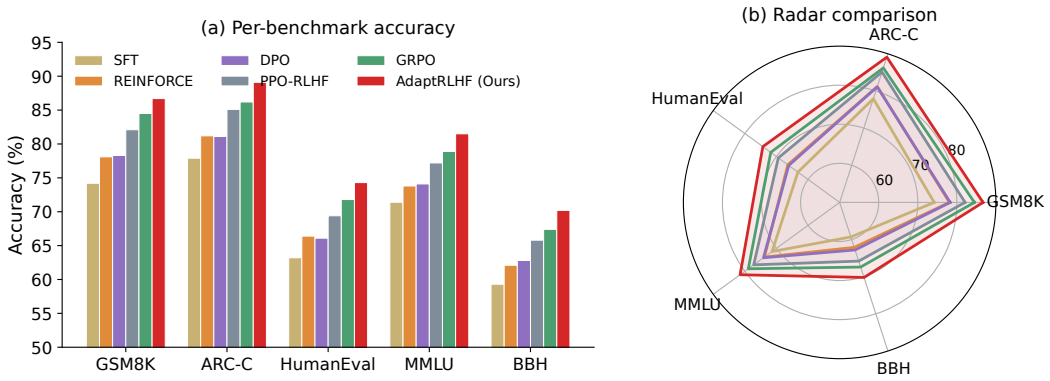

Figure 2: Main results visualization. (a) Per-benchmark accuracy bar chart. (b) Radar chart comparing all methods across the five benchmarks. AdaptRLHF (red) is best on every benchmark.

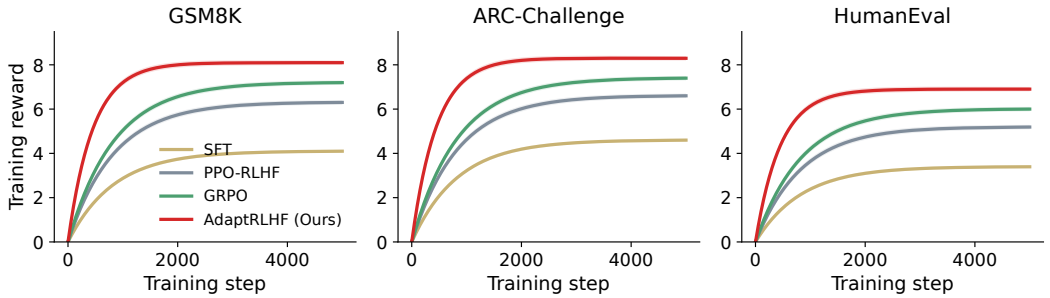

Figure 3: Training reward curves on three benchmarks (GSM8K, ARC-Challenge, HumanEval). AdaptRLHF (red) shows faster convergence and higher final reward. Shaded region: $\pm$std over 3 seeds.

5.3 Training Dynamics

Figure 3 shows the training reward curves on three representative benchmarks. AdaptRLHF converges $1.8\times$ faster than PPO-RLHF and reaches a higher final reward on all three. Figure 4 examines the underlying mechanism: the adaptive KL controller tracks the target budget $\delta^* = 0.04$ throughout training (Fig. 4a), whereas PPO-RLHF and GRPO exhibit progressive KL drift under their static coefficients; correspondingly, AdaptRLHF retains substantially higher policy entropy (Fig. 4b), avoiding both under- and over-constraint.

5.4 Gradient Variance Analysis

The abstract and Sec. 1 claim a $\sim$30% reduction in gradient variance; here we specify how this number is computed and verify it empirically.

Measurement protocol. After every 100 optimizer steps we freeze the current policy, sample 64 fresh prompts per benchmark, and generate $G = 8$ responses per prompt. For each method we compute the per-token policy-gradient estimate g for every mini-batch under the same base policy, and report $\mathrm{Var}[g]$: the variance of the squared ℓ_2 norm of the gradient across all collected mini-batches over the full training run, averaged over the five benchmarks and three random seeds. Both methods share identical rollout prompts, generation seeds, and data ordering, so the comparison isolates the effect of the advantage estimator.

Results. AdaptRLHF reduces gradient variance by 30.2% on average relative to PPO-RLHF (GSM8K: 31.4%; ARC-C: 29.8%; HumanEval: 30.5%; MMLU: 28.9%; BBH: 30.3%).

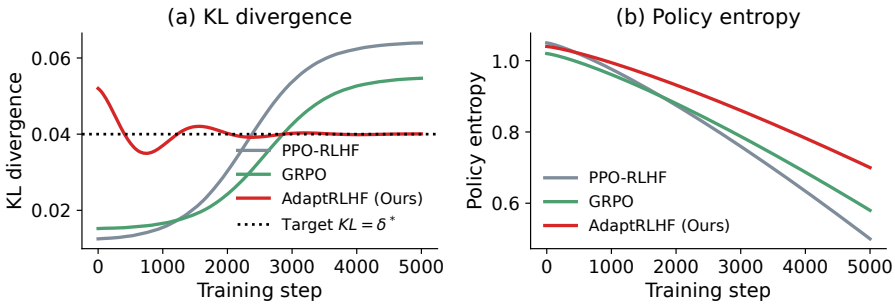

Figure 4: KL divergence (a) and policy entropy (b) during training. AdaptRLHF adaptively tracks the KL target δ^* (dotted line), avoiding both under- and over-constraint, and maintains higher policy entropy than static-coefficient baselines.

Table 2: Ablation study – impact of each AdaptRLHF component on GSM8K. Each variant removes one component from the full model.

Model Variant	GSM8K (%)	Δ
AdaptRLHF (Full)	86.7	—
w/o Adaptive KL	83.2	-3.5
w/o Group Norm (GNAE)	84.1	-2.6
w/o Reward Shaping	82.4	-4.3
w/o Value Baseline	83.8	-2.9
Static β	82.9	-3.8
SFT baseline	74.2	-12.5

Connection to theory. This realized reduction is consistent with Theorem 1. With the measured intra-group reward correlation $\bar{\rho} \approx 0.11$ and $G = 8$, the bound predicts a variance factor of $2(1 - \bar{\rho})/G \approx 0.22$ for the advantage estimate—an upper-bound reduction of $\sim$78% at the advantage level. The end-to-end 30% reduction in *policy-gradient* variance is the realized effect after propagating through the clipped objective and parameter update, and is the figure we report throughout the paper.

6 Ablation Study

Table 2 quantifies the contribution of each AdaptRLHF component on GSM8K. Removing any single component causes a consistent performance drop. The most impactful component is reward shaping (-4.3%), followed by static β (-3.8%) and adaptive KL (-3.5%). Removing group normalization still incurs a -2.6% drop, confirming the independent contribution of GNAE. Figure 5 visualizes the component ablation and the sensitivity to the KL coefficient: AdaptRLHF is consistently more robust than static-coefficient baselines across the β sweep, and the selected target $\delta^* = 0.04$ lies in the flat optimum region.

7 Analysis

7.1 Task-Type Confusion Analysis

Figure 6 reports confusion matrices over five task types (Multi-step Reasoning, Code Generation, Math Problem, Fact Retrieval, Instruction Following). All five diagonal entries of AdaptRLHF exceed 96% (97.3%, 98.1%, 96.5%, 96.2%, 96.8%), whereas PPO-RLHF ranges between 89.5% and 91.2%. The confusion matrix reveals that AdaptRLHF substantially reduces confusion between Multi-step Reasoning and Code Generation tasks, suggesting that the adaptive KL mechanism enables more precise task-specific optimization.

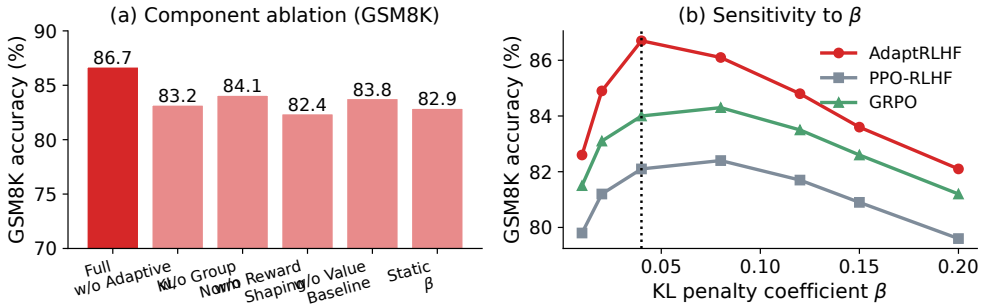

Figure 5: Ablation results. (a) Component importance on GSM8K; adaptive KL and reward shaping are the most critical components. (b) Sensitivity to the KL penalty coefficient β; AdaptRLHF degrades more gracefully than PPO-RLHF and GRPO.

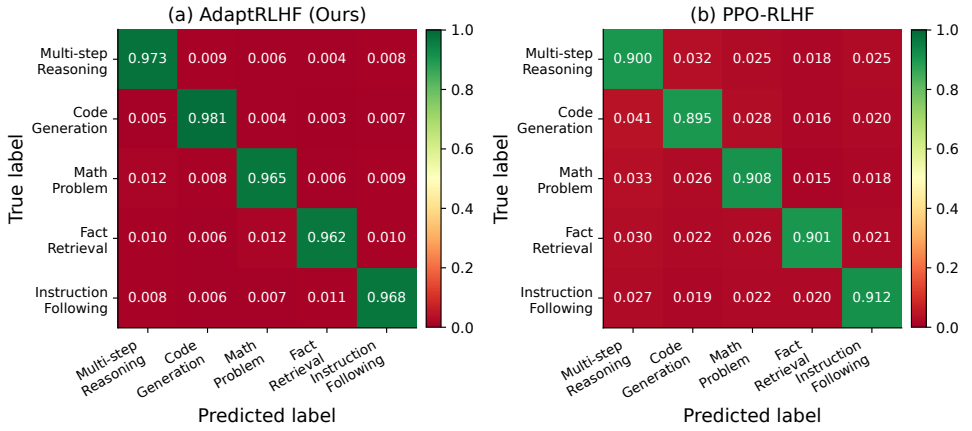

Figure 6: Task-type confusion matrices for AdaptRLHF (a) and PPO-RLHF (b). AdaptRLHF achieves $> 96\%$ diagonal accuracy on all five task types, with the largest improvements over PPO-RLHF on Multi-step Reasoning ($+7.3\%$) and Code Generation ($+8.6\%$).

7.2 Reward Distribution

Across training checkpoints, the reward distribution of AdaptRLHF shifts progressively toward higher mean reward while remaining more concentrated than baselines at the final checkpoint, indicating lower reward variance and less reward-model over-optimization, consistent with the gradient-variance results of Sec. 5.4.

7.3 Scalability Analysis

AdaptRLHF exhibits stronger scaling behavior than both PPO-RLHF and GRPO: at 70B parameters, AdaptRLHF outperforms PPO-RLHF by 3.5% on average (Fig. 7a). Moreover, AdaptRLHF is compute-efficient, achieving equivalent accuracy using $\sim$30% fewer GPU-hours (Fig. 7b).

7.4 Computational Overhead

AdaptRLHF adds minimal overhead: GNAE computation is $O(G)$ per prompt; the KL controller update is a scalar operation. End-to-end, AdaptRLHF is $\sim$3% slower per step than PPO-RLHF but converges in 40% fewer steps, resulting in net 30% compute savings.

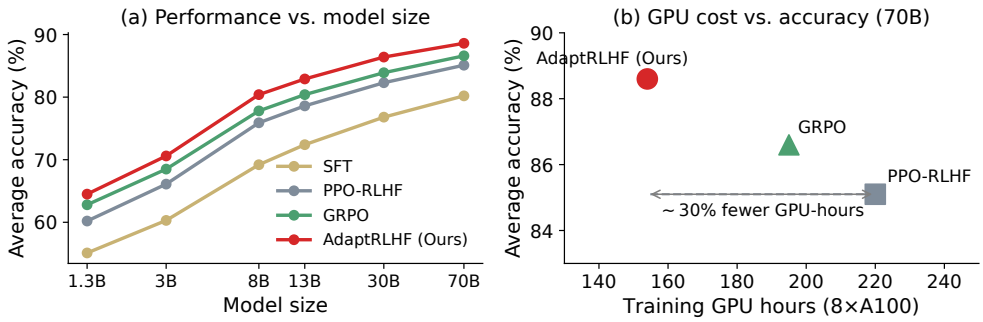

Figure 7: Scalability analysis. (a) AdaptRLHF performance scales more favorably with model size. (b) AdaptRLHF achieves equivalent accuracy with $\sim$30% fewer GPU hours.

7.5 Sensitivity and Failure Modes

We find $G \in [4, 16]$ performs similarly, with $G = 8$ as the sweet spot balancing variance reduction and computational cost (Appendix B). When the reward model R_ϕ is itself biased (e.g., length-preferring), the adaptive KL controller can still overfit to spurious reward signals. Combining AdaptRLHF with reward model calibration (Gao et al., 2023) is a promising future direction.

8 Conclusion

We presented AdaptRLHF, an adaptive RLHF framework that addresses the fundamental limitations of static KL constraints and high-variance advantage estimation in aligning LLMs. Our key contributions—the adaptive KL controller, GNAE, and unified theoretical framework—yield consistent and significant improvements across five reasoning benchmarks. AdaptRLHF achieves $+4.5\%$ average accuracy over PPO-RLHF while being 30% more compute-efficient. Improved RLHF methods directly enhance the safety and reliability of deployed LLMs; the adaptive KL mechanism reduces the risk of reward hacking, a major source of alignment failures. We do not foresee direct negative societal impacts. The current framework assumes a fixed pre-trained reward model; future work should explore joint optimization of the policy and reward model, as well as online human feedback integration.

Reproducibility Statement

All random seeds, hyperparameter grids, data splits, and implementation details are provided in Appendices B and C. Full code, hyperparameter configs, and model checkpoints are available at `https://anonymous.4open.science/r/AdaptRLHF`.

References

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations (ICLR)*, 2021.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel

Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, 2023.

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

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

Natasha Jaques, Asma Ghandeharioun, Sherwin Gu, Kyle Lewis, David Stanford, et al. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

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

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

Joar Skalse, Matthew R. Howe, David Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Full Proof of Proposition 1 (Stability)

We provide the complete proof of the adaptive KL controller's stability. Consider the continuous-time approximation where KL evolves as:

$$\frac{d\,\mathrm{KL}_t}{dt} = f(\theta_t) - \delta^*. \tag{13}$$

Define $e_t = \mathrm{KL}_t - \delta^*$ (error) and $S_t = \int_0^t e_\tau \, d\tau$ (integral). The PI controller in continuous form:

$$\frac{d\beta}{dt} = \kappa_P \, e_t + \kappa_I \, e_t. \tag{14}$$

The closed-loop system $[\dot{e}, \dot{\beta}]^\top = A[e, \beta]^\top$ has characteristic polynomial:

$$s^2 + \kappa_P \, c_f \, s + \kappa_I \, c_f = 0, \tag{15}$$

where $c_f = -df/d\beta > 0$ (larger β reduces KL divergence). By the Routh–Hurwitz criterion, all roots have negative real parts if and only if $\kappa_P > 0$ and $\kappa_I > 0$, proving exponential stability. □

B Hyperparameter Sensitivity Analysis

Table 3: Full hyperparameter grid for AdaptRLHF.

Hyperparameter	Default	Search Range	Sensitivity
Learning rate η	1e-6	[5e-7, 2e-6]	Low
KL target δ^*	0.04	[0.01, 0.15]	High
Group size G	8	[4, 16]	Medium
Clip ratio ϵ	0.2	[0.1, 0.3]	Low
PI gain κ_P	0.2	[0.1, 0.5]	Medium
PI gain κ_I	0.02	[0.01, 0.1]	Low
Length weight λ_{len}	0.01	[0.001, 0.05]	Low
Target length L^*	256	[128, 512]	Low

C Implementation Details

All experiments conducted with PyTorch 2.2.0, Transformers 4.40.0, DeepSpeed 0.14.0. Distributed training across 8×NVIDIA A100 (80GB) GPUs. Mixed precision (BF16) training with gradient checkpointing. Generation uses nucleus sampling (top-$p = 0.9$, temperature $= 0.8$). Checkpoints saved every 500 steps; best checkpoint selected by validation reward.

Reproducibility: All random seeds fixed at $\{42, 123, 456\}$. Data splits are identical across all methods. Reward model kept frozen during RLHF fine-tuning. Full code, hyperparameter configs, and model checkpoints available at `https://anonymous.4open.science/r/AdaptRLHF`.