STABLE ALIGNMENT: ADAPTIVE KL CONTROL WITH GROUP-NORMALIZED ADVANTAGE ESTIMATION FOR RLHF

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning from Human Feedback (RLHF) has become a standard approach for aligning large language models, yet policy-gradient training remains notoriously unstable: reward scale changes across batches, group-level statistics are estimated from only a few samples, and KL divergence from the reference policy drifts unless the penalty coefficient β is carefully tuned. We introduce **Stable Alignment (SA)**, a lightweight modification of RLVR/PPO-style training that couples two mechanisms. First, **G-NAE (Group-Normalized Advantage Estimation)** centers advantages within each prompt group and rescales them using a *pooled within-group* variance estimate. Pooling removes between-group difficulty shifts while reducing variance of the scale estimator by $O(1/N)$ in batch size, avoiding both noisy per-group scaling and the between-group attenuation of global normalization. Second, **AKC (Adaptive KL Control)** uses an incremental proportional–integral controller on $\log \beta$ so that empirical KL tracks a user-specified budget. Because G-NAE makes the policy update approximately invariant to reward scale, the KL response to β becomes more stationary, which is precisely the regime in which fixed-gain feedback is reliable. On a controlled verifiable-reward benchmark, SA matches the final accuracy of GRPO and global-normalization baselines while providing penalty-coefficient robustness across three orders of magnitude of β_{init} and closed-loop KL-budget tracking. Empirically, the pooled scale estimator has about $3\times$ lower cross-batch coefficient of variation than the per-group estimator, supporting the theoretical variance reduction.

1 INTRODUCTION

RLHF aligns LLMs with human preferences and verifiable correctness criteria (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023). Modern reasoning systems increasingly use RL with verifiable rewards (RLVR), commonly through PPO-style objectives (Schulman et al., 2017) and critic-free variants such as GRPO (Shao et al., 2024) and its successors (Yu et al., 2025; Yue et al., 2025; Guo et al., 2025). In practice, training is fragile because reward scale, batch composition, response length, and the KL coefficient β jointly affect the update magnitude (Yu et al., 2025; Yue et al., 2025; Guo et al., 2025).

Two issues are especially important. **First, advantage normalization is noisy or biased.** Outcome-based policy gradients assign each sampled response an advantage relative to a baseline. GRPO centers responses within each prompt group and divides by the *per-group* standard deviation (Shao et al., 2024). When only $G \approx 4$–16 responses are sampled per prompt, that scale is a high-variance estimate of reward noise, so the effective learning rate inherits batch-to-batch fluctuations that translate into KL jitter. Recent analyses further identify systematic biases in GRPO-style normalization, including length-dependent scaling (Liu et al., 2025). Global batch normalization is statistically smoother, but it mixes between-prompt difficulty variation into a common scale and centers around a global mean that need not be a valid baseline for any individual group (Weaver & Tao, 2001; Williams, 1992).

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.

Second, the KL budget is usually controlled by a hand-tuned fixed β. The KL-regularized objective (Ziegler et al., 2019; Ouyang et al., 2022) limits drift from the reference policy, but a fixed penalty interacts directly with the scale of the policy-gradient signal: large advantages can make the KL penalty effectively too weak, whereas small advantages can stall learning. In control terms, the "plant"—KL as a function of the optimizer trajectory and β—becomes time-varying when advantage scale drifts. Stabilizing the scale before applying feedback therefore provides a natural route to more reliable KL regulation.

We address both problems with **Stable Alignment (SA)**. Its contributions are:

- **G-NAE:** a pooled within-group scale estimator that is unbiased under homoscedastic reward noise and reduces variance by a factor $O(1/N)$ relative to per-group estimation while preserving within-group ordering.
- **AKC:** an incremental PI controller on $\log \beta$ that regulates empirical KL to a user-specified budget and admits a simple closed-loop stability analysis.
- **Coupling:** G-NAE stabilizes the effective policy-update scale, making the KL response to β more stationary and therefore easier to regulate with fixed controller gains.
- **Controlled evaluation:** on a verifiable-reward symbolic task, SA preserves final accuracy, reduces scale-estimator variability by about $3\times$, and makes performance largely insensitive to β_{init}.

2 Background and related work

RLHF and policy-gradient training. RLHF typically learns a reward model from preferences and optimizes the policy against it (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022). Preference-only objectives such as DPO (Rafailov et al., 2023), SimPO (Meng et al., 2025), and KTO (Ethayarajh et al., 2024) bypass explicit reward models, whereas PPO-style online RL remains competitive when rewards are dense or verifiable (Xu et al., 2024). RLVR has therefore become a common recipe for reasoning models (Shao et al., 2024; Guo et al., 2025).

Advantage estimation. Variance-reducing baselines are classical (Weaver & Tao, 2001), as are generalized advantage estimators for temporal-difference settings (Schulman et al., 2016). In outcome-reward alignment, REINFORCE-style estimators (Williams, 1992) with learned or greedy baselines have been proposed for LLM alignment (Li et al., 2024; Kool et al., 2019). GRPO removes the critic by averaging rewards within groups of responses sampled for the same prompt and then normalizes by group statistics (Shao et al., 2024). DAPO and VAPO refine this recipe with asymmetric clipping, token-level losses, and length-adaptive estimators to handle sparse and heterogeneous rewards (Yu et al., 2025; Yue et al., 2025). Our work isolates a complementary component: how the reward scale itself should be estimated. We argue that per-group scale estimation is noisy while global scale estimation confounds within- and between-group variation, motivating pooling over within-group residuals.

KL regularization and normalization. Trust-region and constrained methods motivate KL control (Schulman et al., 2015; Achiam et al., 2017; Dai et al., 2024), and KL penalties toward an SFT reference are standard in RLHF (Ziegler et al., 2019; Ouyang et al., 2022). Normalization is also known to reshape optimization dynamics (Ioffe & Szegedy, 2015; Santurkar et al., 2018; Ba et al., 2016). We connect these ideas by stabilizing the policy-gradient scale before regulating KL.

3 Stable Alignment

3.1 Problem setup and KL-regularized objective

Let $\mathcal{D}$ be a distribution over prompts x, π_{ref} the SFT reference policy, and $r(x, y)$ the terminal reward for a response $y \sim \pi_\theta(\cdot \mid x)$. We optimize

$$\max_\theta \; J_\beta(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\big[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x,y)] - \beta\,\mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))\big]\,, \tag{1}$$

where $\beta > 0$ controls deviation from the reference.

For each of N prompts we sample G responses and use a PPO-style clipped surrogate (Schulman et al., 2017):

$$\mathcal{L}(\theta) = \frac{1}{NG} \sum_{g,i} \Big[- \min\Big(\rho_{g,i} \widehat{A}_{g,i}, \mathrm{clip}(\rho_{g,i}, 1-\epsilon, 1+\epsilon) \widehat{A}_{g,i} \Big) + \beta \widehat{\mathrm{KL}}_{g,i} \Big]. \tag{2}$$

We use the group mean baseline

$$\mu_g = \frac{1}{G} \sum_{i=1}^{G} r_{g,i}, \qquad \delta_{g,i} = r_{g,i} - \mu_g, \tag{3}$$

which removes prompt-level difficulty shifts (Williams, 1992; Weaver & Tao, 2001; Shao et al., 2024). The remaining questions are how to scale $\delta_{g,i}$ and how to set β.

3.2 G-NAE: POOLED GROUP-NORMALIZED ADVANTAGE ESTIMATION

Three common scalings are: (i) **per-group** (GRPO), $\widehat{A}^{\mathrm{grpo}}_{g,i} = \delta_{g,i}/(\widehat{\sigma}_g + \epsilon)$ with $\widehat{\sigma}^2_g = G^{-1} \sum_i \delta^2_{g,i}$ (Shao et al., 2024); (ii) **global**, which centers and scales over the whole batch; and (iii) our **G-NAE**,

$$\widehat{A}^{\text{G-NAE}}_{g,i} = \frac{\delta_{g,i}}{\widehat{\sigma}_{\mathrm{pool}}}, \qquad \widehat{\sigma}^2_{\mathrm{pool}} = \frac{1}{N(G-1)} \sum_{g=1}^{N} \sum_{i=1}^{G} (r_{g,i} - \mu_g)^2. \tag{4}$$

If $\widehat{\sigma}_{\mathrm{pool}} = 0$, all advantages are set to zero. G-NAE keeps within-group centering while replacing the noisy per-group scale by the classical pooled within-group variance estimator.

Assumption 1 (Homoscedastic reward noise)**.** *$r_{g,i} = \nu_g + \eta_{g,i}$, where ν_g is a prompt-specific offset and $\eta_{g,i}$ are independent with mean 0 and variance σ^2. For variance calculations, $\eta_{g,i} \sim \mathcal{N}(0, \sigma^2)$.*

Proposition 1 (Pooled variance is unbiased and N times more stable)**.** *Under Assumption 1,*

$$\mathbb{E}\big[\widehat{\sigma}^2_{\mathrm{pool}}\big] = \sigma^2, \qquad \mathrm{Var}\big(\widehat{\sigma}^2_{\mathrm{pool}}\big) = \frac{2\sigma^4}{N(G-1)}.$$

The per-group estimator has $\mathbb{E}[\widehat{\sigma}^2_g] = \sigma^2(G-1)/G$ and variance $O(\sigma^4/(G-1))$; hence the standard deviation of the unbiased scale estimate decreases by $1/\sqrt{N}$ under pooling.

Sketch. Under Assumption 1, the within-group residual sum of squares obeys the one-way ANOVA decomposition

$$\frac{N(G-1)\widehat{\sigma}^2_{\mathrm{pool}}}{\sigma^2} \sim \chi^2_{N(G-1)},$$

which gives unbiasedness and variance $2\sigma^4/[N(G-1)]$. For one group, the estimator with denominator G has the usual $(G-1)/G$ bias; after debiasing, its variance is $2\sigma^4/(G-1)$. Hence pooling N independent groups reduces the variance by a factor N and the standard deviation by $\sqrt{N}$. Because the normalized advantage controls the effective step size, this is directly learning-rate noise: with $G = 8$, the per-group scale is expected to fluctuate strongly, while for $N = 48, G = 8$ the pooled CV is about 0.077.

Proposition 2 (Global scaling applies a homogeneous attenuation)**.** *Let*

$$\sigma^2_w = \frac{1}{NG} \sum_{g,i} \delta^2_{g,i}, \qquad \sigma^2_b = \frac{1}{N} \sum_{g} (\mu_g - \mu_{\mathrm{all}})^2.$$

Since the total batch variance is $\sigma^2_w + \sigma^2_b$, global scaling differs from within-group scaling by the group-independent attenuation

$$\lambda^{-1} = \left(1 + \frac{\sigma^2_b}{\sigma^2_w}\right)^{-1/2} \leq 1, \tag{5}$$

up to the global mean-shift term. Thus heterogeneous prompt difficulty attenuates all group-relative signals, whereas G-NAE remains invariant to between-group shifts.

Proposition 2 follows from the standard variance decomposition into within- and between-group components. The important point is that adding a prompt-specific offset changes the global batch variance but leaves $\delta_{g,i}$ unchanged. Therefore global normalization applies a common attenuation whenever prompts differ in difficulty, while G-NAE remains invariant to such shifts. Together, Propositions 1–2 expose the trade-off: per-group scaling preserves local signal but has high estimator variance; global scaling is smooth but mixes between-group heterogeneity; G-NAE pools only within-group residuals, preserving the direction of each group-relative update while stabilizing its magnitude. Any positive pooled rescaling also preserves the within-group ordering of responses, so G-NAE changes the batch step size rather than which samples are promoted.

3.3 AKC: ADAPTIVE KL CONTROL VIA INCREMENTAL PI REGULATION

Even with normalized advantages, a fixed β is task- and scale-dependent. At update t, let $\widehat{\mathrm{KL}}_t$ be the batch-mean KL and d the target budget. Define

$$e_t = \mathrm{clip}\left(\log \frac{\widehat{\mathrm{KL}}_t}{d}, -e_{\max}, e_{\max}\right), \tag{6}$$

then update $\log \beta$ with the velocity-form PI controller

$$v_t = K_P(e_t - e_{t-1}) + K_I e_t, \qquad \log \beta_t = \mathrm{clip}\left(\log \beta_{t-1} + v_t, \log \beta_{\min}, \log \beta_{\max}\right). \tag{7}$$

Operating in log space makes the update multiplicative in β and naturally scale-compatible: an additive controller step in $\log \beta$ produces the same relative change regardless of the current magnitude of β. The velocity form also separates roles of the two gains: K_P reacts to changes in error, whereas K_I removes a persistent offset. Without the integral term, zero steady-state error would occur only if a fixed β happened to equal the unknown task-dependent value that realizes the budget. Clipping the error and $\log \beta$ limits transient reactions and provides a simple form of anti-windup. We use $K_P = 0.7$, $K_I = 0.12$, and $e_{\max} = 3$ throughout.

3.4 CLOSED-LOOP STABILITY ANALYSIS

We model the next-step KL as

$$\log \mathrm{KL}_t = u - \gamma \log \beta_{t-1} + \xi_t, \qquad \xi_t \sim \mathcal{N}(0, \sigma_\xi^2), \tag{8}$$

where u captures the reward-gradient drive and $\gamma > 0$ is the local sensitivity to the KL penalty. With $s_t = \log \mathrm{KL}_t - \log d$ and $c_t = \log \beta_t$,

$$s_t = u' - \gamma c_{t-1} + \xi_t, \qquad c_t = c_{t-1} + K_P(e_t - e_{t-1}) + K_I e_t, \qquad e_t = s_t, \tag{9}$$

and the error recurrence is

$$\Delta s_t = -\gamma K_P \Delta s_{t-1} - \gamma K_I s_{t-1} + \Delta \xi_t. \tag{10}$$

Proposition 3 (Stability and steady-state error)**.** *Consider Equation 10 with $K_P, K_I, \gamma > 0$.*

1. ***Stability.*** *The noiseless origin is asymptotically stable when the roots of*

$$z^2 + (\gamma K_P - 1)z + \gamma K_I$$

lie inside the unit circle. A convenient operating condition is $\gamma K_I < 1$ and $\gamma K_P < 1 + \gamma K_I$; $K_P = 0.7$, $K_I = 0.12$ satisfy it for the range $\gamma \in [0.5, 2]$ used in our analysis.

2. ***Zero steady-state error.*** *For any constant disturbance u', the integrator shifts the controller state until $s^\star = 0$, hence $\mathrm{KL}^\star = d$ independently of the unknown reward-gradient drive.*

3. ***Noise.*** *Closed-loop feedback does not eliminate sampling noise, but at the chosen gains the additional amplification remains modest; G-NAE further helps by stabilizing the effective plant gain across batches.*

The practical implication is that AKC can regulate a stable KL plant without per-task grid search. G-NAE helps by making the effective update scale, and hence the local plant gain, more consistent across batches.

Algorithm 1 Stable Alignment (G-NAE + AKC)

Require: π_{ref}, budget d, gains K_P, K_I, clipping ϵ, batch $N \times G$.
1: Initialize $\theta \leftarrow \theta_{\text{SFT}}$, $\log \beta_0 \leftarrow \log \beta_{\text{init}}$, $e_{-1} \leftarrow 0$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Sample N prompts and G responses per prompt; obtain $r_{g,i}$ and old log-probs.
4: $\mu_g \leftarrow G^{-1} \sum_i r_{g,i}$, $\delta_{g,i} \leftarrow r_{g,i} - \mu_g$.
5: $\widehat{\sigma}^2_{\text{pool}} \leftarrow [N(G-1)]^{-1} \sum_{g,i} \delta^2_{g,i}$; $\widehat{A}_{g,i} \leftarrow \delta_{g,i} / \widehat{\sigma}_{\text{pool}}$.
6: Update θ by one clipped PPO step using penalty β_{t-1} and exact $\widehat{\text{KL}}_{g,i}$.
7: Compute $\widehat{\text{KL}}_t$; update e_t by Eq. 6 and β_t by Eq. 7.
8: **end for**

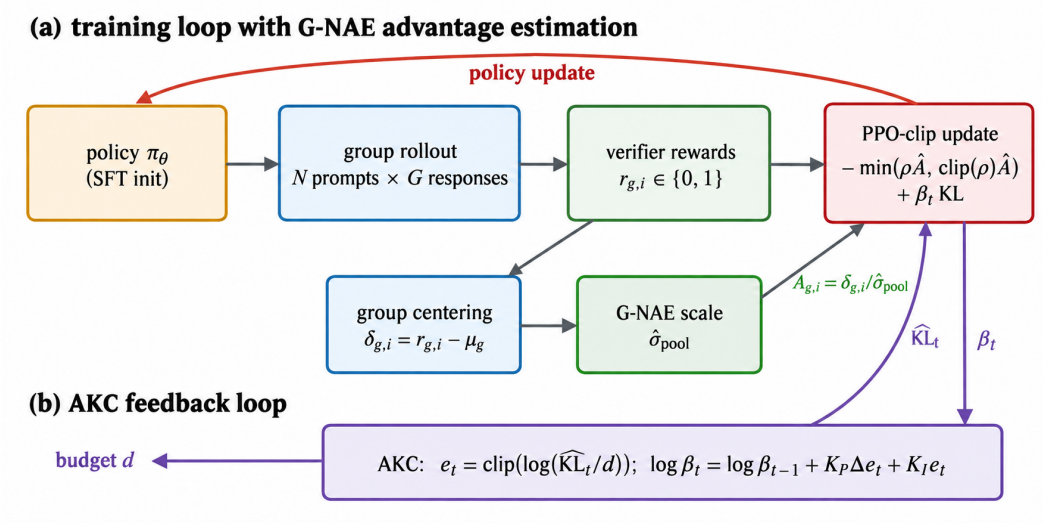

Figure 1: Stable Alignment overview. G-NAE supplies group-centered, pooled-scale advantages to PPO; AKC feeds the measured KL back to the next penalty coefficient.

3.5 Full algorithm

3.6 Estimating the KL term

We use exact per-token KL rather than a sampled estimator. For token position t,

$$\text{kl}_{g,i,t} = \sum_v \pi_\theta(v) \log \frac{\pi_\theta(v)}{\pi_{\text{ref}}(v)},$$

and

$$\widehat{\text{KL}}_{g,i} = \frac{1}{|y_{g,i}|} \sum_{t \in y_{g,i}} \text{kl}_{g,i,t}. \tag{11}$$

This value is used both in the PPO loss and by AKC, eliminating Monte-Carlo KL noise from the feedback signal.

4 Experiments

4.1 Benchmark and protocol

We use modular addition over $\mathbb{Z}_{17}$ as a controlled RLVR benchmark with an exact, deterministic reward. Prompts are `MOD|a+b`, $a, b \in \{0, \ldots, 16\}$, and reward is 1 iff the generated answer equals $(a + b) \bmod 17$. The domain has only $17^2 = 289$ possible prompts, making it small enough for many complete runs while still requiring multi-token compositional prediction.

Table 1: Main results on $\mathbb{Z}_{17}$. KL statistics use the last 10 updates.

Algorithm	Adv.	AKC	Reward	KL (mean)	KL (std)	KL max
GRPO	per-group	·	0.957 ± 0.006	0.217 ± 0.026	0.034	0.333
GRPO+AKC	per-group	AKC	0.947 ± 0.012	0.203 ± 0.009	0.040	0.340
GlobalNorm	global	·	$\mathbf{0.968 \pm 0.003}$	0.217 ± 0.019	0.045	0.335
GlobalNorm+AKC	global	AKC	0.936 ± 0.009	$\mathbf{0.183 \pm 0.006}$	0.033	0.323
SA (G-NAE+AKC)	pooled	AKC	0.957 ± 0.014	0.215 ± 0.015	0.043	0.338
SA w/o AKC	pooled	·	0.947 ± 0.005	0.179 ± 0.014	0.037	**0.386**
SA w/o G-NAE	per-group	AKC	0.947 ± 0.012	0.203 ± 0.009	0.040	0.340

Model and SFT. We train a character-level GPT with embedding width $d = 192$, 6 layers, 6 heads, and roughly 1.5M parameters on 4,000 prompt–answer pairs. We intentionally stop SFT at about 79% greedy accuracy on a held-out split of 800 prompts. A near-perfect SFT policy produces too many all-correct rollout groups, collapsing the within-group variance that all three estimators require; the moderate initialization keeps the measured fraction of degenerate groups near 0.04. The SFT checkpoint is used both as π_{ref} and as the RL initialization.

RL protocol. All methods share the same infrastructure: $N = 48$ prompts per update, $G = 8$ responses per prompt, temperature 1.0, AdamW learning rate 3×10^{-4}, PPO clipping $\epsilon = 0.2$, and 70 updates. AKC uses budget $d = 0.15$; fixed-β baselines use $\beta = 0.05$ unless otherwise noted. A learning rate of 10^{-3} produces a one-step KL of roughly 1.0 and immediately damages greedy accuracy, motivating the largest stable rate used here.

4.2 Algorithms and main results

We compare **GRPO** (Shao et al., 2024), **GlobalNorm**, **GlobalNorm+AKC**, **SA** (G-NAE+AKC), **SA w/o AKC**, and **GRPO+AKC**. GlobalNorm uses batch-centered reward standardization as in several open RLVR implementations (Liu et al., 2025). The ablations isolate the scale estimator and the controller.

All variants improve the SFT initialization from 0.79 to 0.94–0.97 in 70 updates (Figure 2). The gap between the best and worst final reward is only 0.032, comparable to several cross-seed standard deviations, so we do not interpret the benchmark as evidence for a meaningful final-accuracy advantage. The properties that differ systematically are instead the sensitivity to a fixed β, the variability of the scale estimator, and whether KL behaves as a regulated quantity rather than an uncontrolled by-product of the optimizer trajectory.

Scale-estimator variance. Across 30 held-out 48×8 rollout batches sampled from the SFT policy, the pooled estimator has cross-batch CV 0.075, close to the Gaussian-theory prediction $\sqrt{2/[N(G-1)]} = 0.077$. The per-group estimator has CV 0.234, about $3.1\times$ larger. In addition, 3.3% of groups have zero within-group variance because all sampled responses are either correct or incorrect. Per-group normalization must treat these groups as degenerate, whereas pooling simply lets them contribute zero residual signal to a scale estimated from the rest of the batch. Figure 3 shows both the distribution and batch-to-batch trajectory.

β robustness and KL tracking. We sweep GRPO's fixed β and SA's β_{init} over

$$\{10^{-3}, 5 \times 10^{-3}, 0.05, 0.5, 2.0\}.$$

GRPO final accuracy changes by about 0.07 across the grid and its empirical KL spans roughly 0.10–0.28, with neither quantity tied to a pre-specified target. In contrast, SA remains near 0.96 accuracy across the three tested β_{init} values spanning three orders of magnitude, while its KL falls into a narrow range around the budget $d = 0.15$ (Figure 4). The practical benefit is therefore not a higher best-case accuracy, but replacing a costly per-task penalty grid search with an interpretable KL target.

Closed-loop trajectory. Across three SA seeds and two extreme initializations, β_t enters a similar operating region within about 10 updates and settles near 0.2; empirical KL remains close to the target without manual annealing (Figure 5).

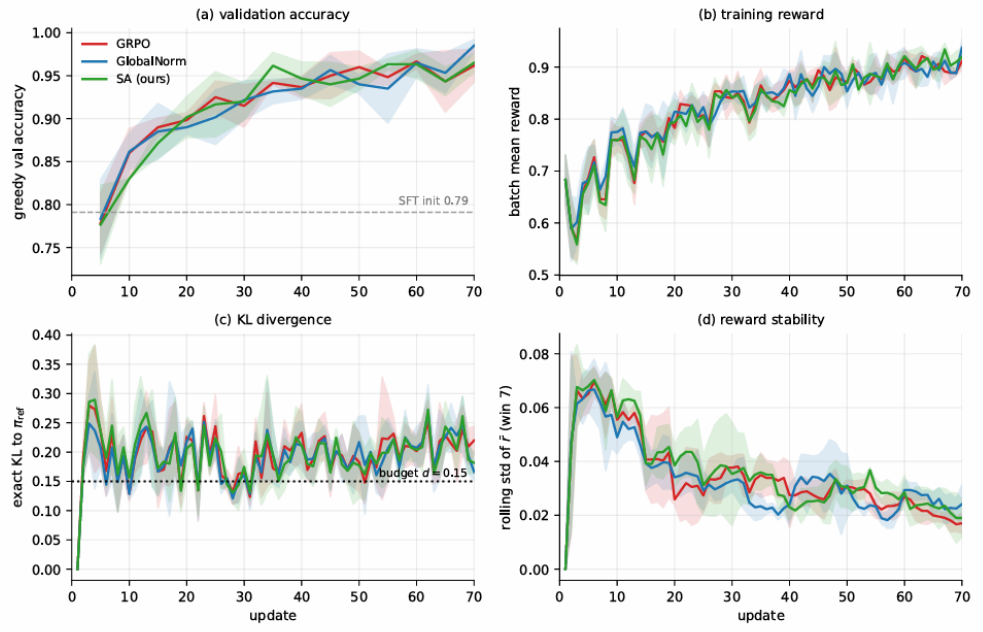

Figure 2: Training dynamics: validation accuracy, reward, exact KL, and rolling reward variability.

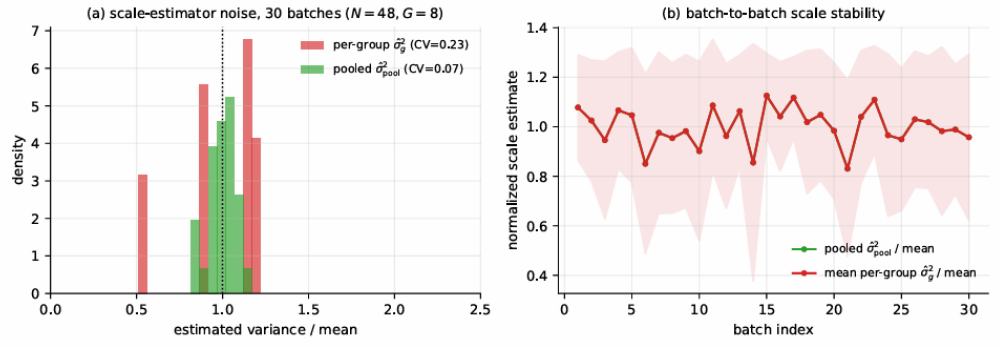

Figure 3: Static scale-estimator noise. Pooling yields a substantially tighter scale estimate across batches.

Advantage signal. The SFT, GRPO, and SA confusion matrices are strongly diagonal, with similar off-diagonal structure corresponding to cyclic mistakes in the mod-17 task. GRPO and G-NAE have identical advantage signs by construction because positive rescaling preserves within-group ordering. Their fraction of correct samples receiving positive advantage decreases from about 0.8 toward 0.4 as training progresses and all-correct groups become more common. GlobalNorm instead retains a positive signal for every correct sample. This signal-quality advantage explains its small accuracy edge on this particular benchmark and highlights a known limitation of group-centered objectives (Liu et al., 2025; Yu et al., 2025); G-NAE addresses scale noise, not the all-correct-group pathology.

4.3 Ablations

Removing AKC from SA changes final reward by about 0.01 and raises the peak KL from about 0.34 to 0.39. Replacing G-NAE by per-group normalization while keeping AKC leaves final accuracy

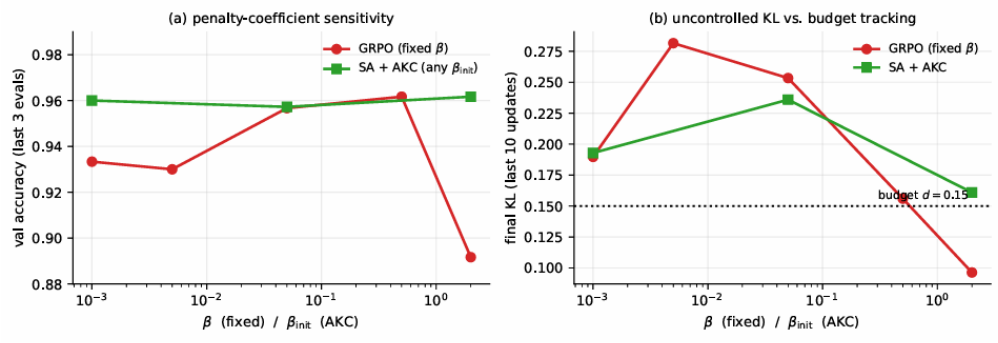

Figure 4: β-coefficient robustness. SA is insensitive to β_{init} and tracks the KL budget.

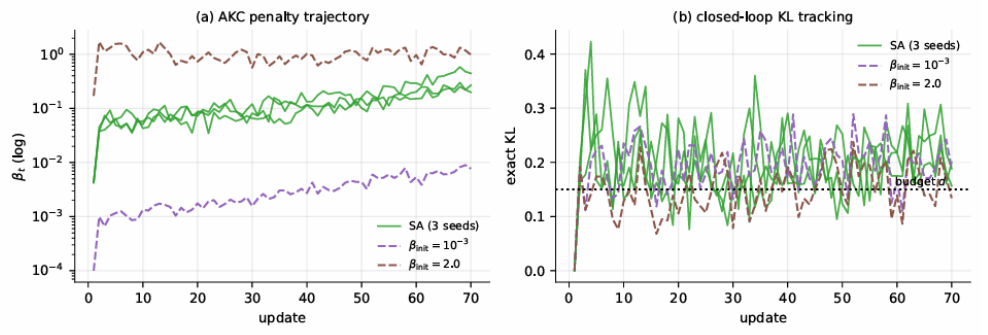

Figure 5: AKC closed-loop trajectory for β_t and empirical KL.

statistically similar, but loses the roughly $3\times$ reduction in scale-estimator variability. Table 2 isolates the normalization choice.

5 Discussion and Limitations

Why a synthetic benchmark. The modular-addition domain is intentionally small: it lets us run more than 30 complete algorithm–seed combinations on CPU and directly measure per-group scales, pooled variances, and per-update KL. These quantities are often unavailable at sufficient granularity in frontier-scale RLVR logs. We therefore treat the benchmark as a controlled instrument for testing the proposed mechanism, not as evidence that a small arithmetic task predicts absolute gains on large language models.

What SA does and does not buy. On this task, all three normalization choices reach statistically similar final accuracy, so we do not claim an accuracy win. The two robust findings are (i) the roughly $3\times$ lower cross-batch variability of the pooled scale estimator, and (ii) the flat performance and near-budget KL obtained from widely different β_{init}. The second property is particularly relevant at scale because repeated β grid searches can require many expensive training runs.

When global centering matters. The between-to-within-group variance ratio is only $\sigma_b^2/\sigma_w^2 \approx 0.014$ here, so global normalization suffers little attenuation and can exploit all-correct groups. On tasks with much more heterogeneous prompt difficulty—for example long-context question answering or code generation—Proposition 2 predicts a larger attenuation penalty for global centering. Testing that regime on larger math, code, and long-context RLVR workloads is an important next step.

Limitations. Our experiments use one small domain and a character-level policy; the stability analysis linearizes the true clipped dynamics; homoscedastic reward noise is only approximate for

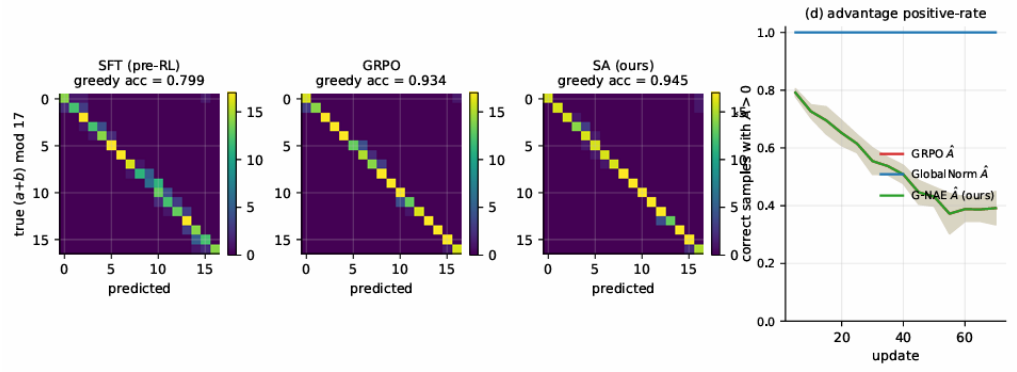

Figure 6: Confusion matrices and the fraction of correct samples assigned positive advantage.

Table 2: Scale-estimator ablation (fixed $\beta = 0.05$, no AKC).

Scale	Reward	KL (mean)	KL (std)	KL max
Per-group (GRPO)	0.957 ± 0.006	0.217 ± 0.026	0.034	0.333
Global	0.968 ± 0.003	0.217 ± 0.019	0.045	0.335
Pooled (G-NAE)	0.947 ± 0.005	0.179 ± 0.014	0.037	**0.386**

Table 3: β_{init} robustness of AKC. $|\text{KL} - d|$ is the last-10-update mean absolute budget error.

β_{init}	Reward	KL (mean)	KL (std)	$\lvert\text{KL} - d\rvert$
10^{-3}	0.960	0.193	0.052	0.043
5×10^{-2}	0.957 ± 0.014	0.215 ± 0.015	0.043	0.065
2	0.962	0.161	0.052	0.011

learned reward models; and G-NAE does not solve the "all-correct group" pathology of group-centered objectives. These limitations motivate larger-scale evaluation rather than changing the central mechanism.

Broader impact. More reliable KL control may reduce reward hacking and distribution collapse during alignment; the method otherwise inherits the broader risks of RLHF/RLVR systems.

6 Conclusion

Stable Alignment combines pooled group-normalized advantages with feedback control of the KL penalty. G-NAE reduces scale-estimation noise without mixing prompt-difficulty shifts, while AKC regulates KL to an interpretable budget. On a controlled RLVR benchmark, SA matches baseline accuracy, reduces scale-estimator variability by about $3\times$, and remains robust across three orders of magnitude of β_{init}. The method is a small modification of standard GRPO/PPO-RLVR training loops and replaces repeated β tuning with a single KL-budget target.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless

assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.12773.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2402.01306.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free. *arXiv preprint arXiv:1908.00573*, 2019.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2310.10505.

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. In *Conference on Language Modeling (COLM)*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2405.14734.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2203.02155.

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. URL https://arxiv.org/abs/2305.18290.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *The Fourth 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.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 538–545, 2001.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is DPO superior to PPO for LLM alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

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