GAC: Noise-Aware Adaptive Mixing for Hybrid SFT-RL Post-Training

Anonymous ACL submission

Abstract

Hybrid post-training combining supervised fine-tuning (SFT) and reinforcement learning (RL) is the standard paradigm for aligning large language models, yet fixed mixing schedules cannot adapt when the relative noise of the two signals evolves. We derive a noise-aware mixing weight μ^* by minimizing an MSE upper bound on the mixed stochastic gradient, yielding a closed-form that balances gradient noise variance and SFT–RL disagreement. Building on the token-wise reweighting of CHORD (Zhang et al., 2025), the practical Guided Adaptive Controller (GAC) adds EMA smoothing, a schedule prior, and capped updates around this estimator, with all statistics estimated online from existing training tensors. The noise-aware controller alone outperforms the best rule-based controller by $+3.0$pp on AMC; the full system reaches $+3.8$pp over HPT across math, code, science, and logic benchmarks, while reducing KL-drift area by 28% and large $|\Delta\mu|$ events by $>70\%$, at $<1\%$ overhead. Gains grow with model size from 1.5B to 14B (Table 6). Code: `https://github.com/anonymous/GAC`.

1 Introduction

Large language models (LLMs) are typically post-trained by combining supervised fine-tuning (SFT) with reinforcement learning (RL). SFT stabilizes format from expert demonstrations while RL improves reward-seeking behavior from on-policy rollouts. However, SFT and RL objectives cannot be fully decoupled without mutual degradation (Niu et al., 2026), and fixed mixtures cannot adapt to evolving policy drift and reward noise, inducing entropy collapse or late over-imitation. Recent hybrid methods address this via interleaved SFT–RL (Su et al., 2025), anchored regularization (Zhu et al., 2025), or conflict-aware coupling (Zeng et al., 2025). We take a complementary approach: a noise-aware global controller that adapts the mixing weight based on online gradient uncertainty estimates, combined with the token-wise stabilization $\varphi(\cdot)$ from CHORD (Zhang et al., 2025). Figure 1 provides an overview.

Contributions The new contribution is a global noise-aware mixing controller and its proxy-based online estimation; the token-wise reweighting $\varphi(p)$ is adopted from CHORD (Zhang et al., 2025). (C1) A closed-form μ^* from MSE minimization (Eq. 3), instantiated via z-normalized proxies (Appendix D), wrapped in a guided controller with motivating stability analysis (Proposition 2), reusing existing tensors with $<1\%$ overhead. (C2) Token-wise SFT reweighting $\varphi(p)=p(1-p)$ adopted from CHORD, contributing +0.4–1.4pp orthogonal to the controller (Section 4.3). (C3) Systematic evaluation across math, code, science, and logic at 1.5B, 7B, and 14B scales.

2 Related Work

Post-training paradigms. Sequential SFT-then-RL exhibits "shift–readapt–overfit" progression; scheduled mixtures remain heuristic (Ouyang et al., 2022; Christiano et al., 2017; Rafailov et al., 2023). Niu et al. (2026) prove that SFT and RL cannot be decoupled without mutual degradation, motivating integrated training.

Dynamic weighting and stability. Multi-task learning employs uncertainty weighting (Kendall et al., 2018), gradient normalization (Chen et al., 2018), or conflict resolution (MGDA/PCGrad/CAGrad) (Sener and Koltun, 2018; Yu et al., 2020; Liu et al., 2021). Recent approaches include Nash-MTL (Navon et al., 2022), FAMO (Liu et al., 2023), Aligned-MTL (Senushkin et al., 2023), SDMGrad (Xiao et al., 2023), and MoCo (Fernando et al., 2023). None explicitly models both gradient noise variance and SFT–RL disagreement. Table 1 summarizes key differences.

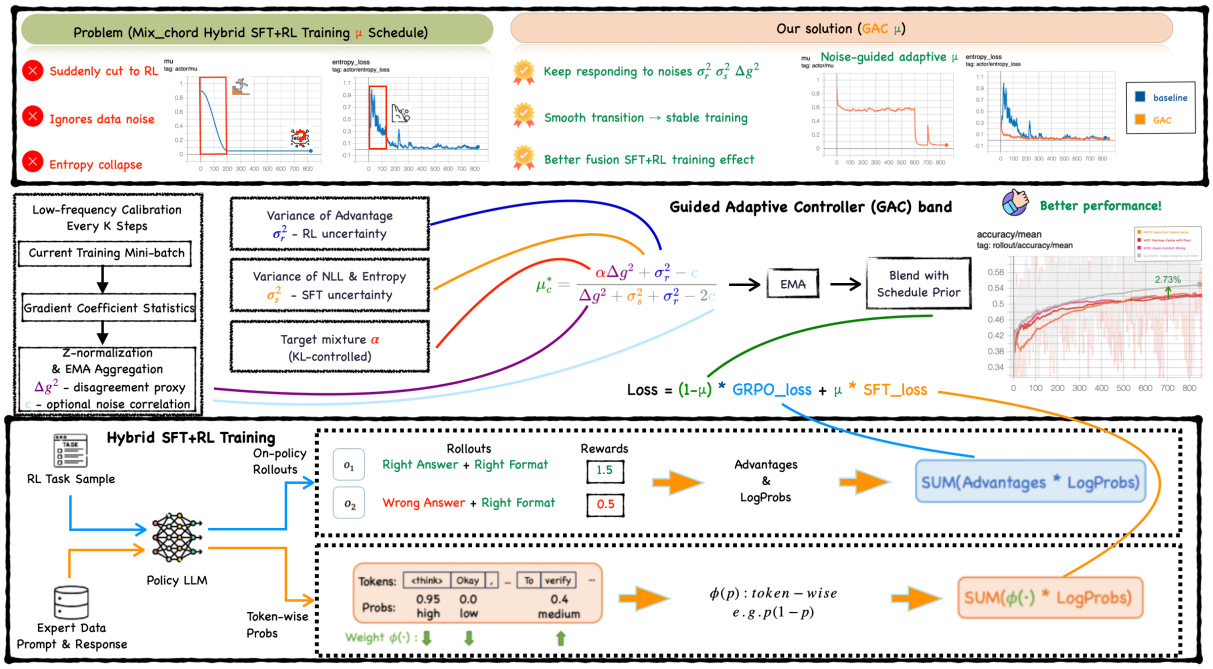

Figure 1: Overview of GAC and the training pipeline. Top: limitations of fixed schedules versus the noise-aware controller. Middle: the Guided Adaptive Controller integrates σ_r^2 (RL uncertainty), σ_s^2 (SFT uncertainty), and $\Delta\tilde{g}^2$ (disagreement proxy) to compute the mixing weight in (3), followed by EMA smoothing, prior blending, and capped updates. Bottom: the pipeline sources signals from GRPO and SFT with token-wise weighting $\varphi(p)=p(1-p)$ (Zhang et al., 2025), composing $L=(1-\mu)L_{\text{GRPO}}+\mu L_{\text{SFT}-\varphi}$. All statistics are estimated online from existing training tensors.

Table 1: Comparison of dynamic weighting methods for SFT–RL mixing. †: derived but omitted due to high estimation variance. ‡: loss-ratio dynamics, not explicit EMA.

Method	Noise σ^2	Disagree. Δg^2	Corr. c	Temporal Smooth	Closed Form
Uncertainty (Kendall et al., 2018)	✓				✓
GradNorm (Chen et al., 2018)		✓			
MGDA/PCGrad		✓			
Nash-MTL (Navon et al., 2022)		✓			
DWA (Liu et al., 2019)				‡	
FAMO (Liu et al., 2023)		✓		✓	✓
Aligned-MTL (Senushkin et al., 2023)		✓			
SDMGrad (Xiao et al., 2023)		✓			
MoCo (Fernando et al., 2023)		✓		✓	
CHORD (Zhang et al., 2025)				✓	
SRFT (Fu et al., 2025)					
LUFFY (Yan et al., 2025)		✓			
HPT (Lv et al., 2025)		✓		✓	
TRAPO (Su et al., 2025)				✓	
ASFT (Zhu et al., 2025)					
GAC (ours)	✓	✓	†	✓	✓

Hybrid SFT–RL post-training. CHORD (Zhang et al., 2025) proposes dual control with a global μ and token-wise $\varphi(p)=p(1-p)$; GAC builds on CHORD, replacing its heuristic schedule with a noise-aware controller. SRFT (Fu et al., 2025) uses entropy-aware weighting; LUFFY (Yan et al., 2025) augments RL with off-policy traces; HPT (Lv et al., 2025) derives accuracy-gated signal selection. TRAPO (Su et al., 2025) interleaves SFT and RL within each instance via trust-region SFT. ASFT (Zhu et al., 2025) anchors the policy to the base distribution via KL regularization. GTA (Zeng et al., 2025) combines supervised and RL signals with conflict mitigation. GAC differs by operating at the gradient noise level with a closed-form μ^* (Proposition 2).

3 Method

Notation. We write μ^* for the idealized optimal mixing weight, σ_s^2, σ_r^2 for SFT/RL gradient noise variance (estimated via proxies), Δg^2 for gradient disagreement, and α_{tgt} for the theoretical target ratio in the MSE derivation. In practice we use a KL-controlled ratio α_{ctrl} (Eq. 11). Throughout, "SFT" denotes $L_{\text{SFT}-\varphi}$ with token-wise weighting $\varphi(p)=p(1-p)$ (Zhang et al., 2025). We introduce additional notation locally when needed.

3.1 Closed-Form μ via MSE Minimization

Let SFT and RL provide gradient estimators $\hat{g}_s = g_s^* + \varepsilon_s$ and $\hat{g}_r = g_r^* + \varepsilon_r$ (noise-free gradients g_s^*, g_r^* plus zero-mean noise with variances σ_s^2, σ_r^2). For mixed gradient $\hat{g}(\mu) = \mu\hat{g}_s + (1-\mu)\hat{g}_r$ and target $g^\star = \alpha_{\text{tgt}} g_s^* + (1-\alpha_{\text{tgt}}) g_r^*$, we derive the

optimal mixture weight.

On the "target gradient" assumption. $g^\star$ is a local control objective encoding a desired trade-off via α_{tgt}, analogous to trust-region surrogates, not a global optimality claim. Under first-order approximation, minimizing MSE to $g^\star$ is equivalent to minimizing a local upper bound on L_{mix} with variance-aware regularization. When $\Delta g^2 \to \infty$, $\mu^* \to \alpha_{\text{tgt}}$: the controller defaults to the user-specified preference. We further employ KL-stabilized α_{ctrl} and capped updates to prevent abrupt shifts (see Limitations).

Definition 1 (Mean Squared Error Objective)**.** *The expected squared error between the mixed gradient and target is:*

$$\mathcal{E}(\mu) \triangleq \mathbb{E}\left[\|\hat{g}(\mu) - g^\star\|^2\right]. \tag{1}$$

Under the assumption that $\mathbb{E}[\varepsilon_s] = \mathbb{E}[\varepsilon_r] = 0$ and independence $\mathbb{E}[\varepsilon_s \varepsilon_r^\top] = 0$, we expand the MSE (see Appendix A for details). Substituting $\hat{g}(\mu)$ and $g^\star$ and using $\hat{g}(\mu) - g^\star = (\mu - \alpha_{\text{tgt}})(g_s^* - g_r^*) + \mu\varepsilon_s + (1-\mu)\varepsilon_r$:

$$\mathcal{E}(\mu) = (\mu - \alpha_{\text{tgt}})^2\, \Delta g^2 + \mu^2 \sigma_s^2 + (1-\mu)^2 \sigma_r^2, \tag{2}$$

where $\Delta g^2 = \|g_s^* - g_r^*\|^2$ denotes gradient disagreement, and cross-terms vanish under independence.

Theorem 1 (Optimal Mixture Weight)**.** *The unique minimizer of $\mathcal{E}(\mu)$ over $\mu \in \mathbb{R}$ is:*

$$\mu^* = \frac{\alpha_{\text{tgt}}\, \Delta g^2 + \sigma_r^2}{\Delta g^2 + \sigma_s^2 + \sigma_r^2}. \tag{3}$$

Proof sketch. Taking $\frac{\partial \mathcal{E}}{\partial \mu} = 0$ from Eq. 2:

$$2(\mu - \alpha_{\text{tgt}})\Delta g^2 + 2\mu\sigma_s^2 - 2(1-\mu)\sigma_r^2 = 0. \tag{4}$$

Solving for μ yields Eq. 3. The second derivative $\frac{\partial^2 \mathcal{E}}{\partial \mu^2} = 2(\Delta g^2 + \sigma_s^2 + \sigma_r^2) > 0$ confirms this is a minimum.

This embodies a bias–variance trade-off: when $\Delta g^2 \to 0$, μ^* reduces to inverse-variance weighting; as $\Delta g^2 \to \infty$, $\mu^* \to \alpha_{\text{tgt}}$.

Correlated noise extension. When independence is violated, the extension with $c = \text{tr}\,\text{Cov}(\varepsilon_s, \varepsilon_r)$ yields:

$$\mu_c^* = \frac{\alpha_{\text{tgt}}\, \Delta g^2 + \sigma_r^2 - c}{\Delta g^2 + \sigma_s^2 + \sigma_r^2 - 2c}. \tag{5}$$

Empirically, c has coefficient of variation >0.8; including it yields +0.2pp (not statistically significant) but triples large $|\Delta\mu|$ events. We omit c in all main experiments (Appendix B).

Biased estimators. Realistic RL estimators are biased (clipping, importance sampling, entropy/KL regularizers). With biases b_s, b_r, minimizing the MSE upper bound yields (Appendix A.3):

$$\tilde{\mu}^* = \frac{\alpha_{\text{tgt}}\, \Delta g^2 + \sigma_r^2 - c + \langle \Delta b,\, \bar{g} \rangle}{\Delta g^2 + \sigma_s^2 + \sigma_r^2 - 2c + \|\Delta b\|^2}, \tag{6}$$

reducing to Equations 3–5 when $b_\cdot = 0$ and $c = 0$.

3.2 Proxy Signals

The closed-form μ^* depends on gradient-level quantities $(\sigma_s^2, \sigma_r^2, \Delta g^2)$ that are expensive to compute at every step. We employ computationally tractable *proxy uncertainty signals*: (i) advantage dispersion for RL uncertainty, and (ii) length-normalized NLL variance for SFT uncertainty.

Theoretical motivation. For RL, the policy gradient is directly scaled by advantages A_t; thus mini-batch advantage variance proxies $\text{Var}(\nabla L_r)$. For SFT, per-sample NLL variance captures gradient heterogeneity. Pearson correlations with true gradient statistics: $r = 0.82 \pm 0.04$ for σ_r^2 and $r = 0.76 \pm 0.05$ for σ_s^2, with sanity checks confirming genuine coefficient structure (Appendix D).

3.3 Stability-Motivated Design Guidelines

We provide stability-motivated analysis under idealized assumptions (L-smooth losses, KL-bounded updates). These serve as a motivating analysis for design choices rather than strict guarantees (Appendix C).

Proposition 2 (Motivating Analysis)**.** *Under idealized smoothness and KL-bounded updates, with $\eta \le \frac{1}{2L}$, $\lambda \le \frac{\rho}{\rho+1}$, and bounded cap $\bar{c}$:*

$$\mathbb{E}[V_{t+1} - V_t] \le -\zeta\, \mathbb{E}[\|\nabla L\|^2] + \mathcal{O}(\bar{c}^2), \tag{7}$$

where $V_t = \|\theta_t - \theta^*\|^2 + \rho(\mu_t - \alpha_{\text{tgt}})^2$ is a Lyapunov potential and $\zeta = \min\{\frac{1}{2L}, \frac{\rho}{\rho+1}\}$.

Practical implications. (i) small $\bar{c}$: the $\mathcal{O}(\bar{c}^2)$ term motivates capping per-step changes; (ii) moderate $\lambda \le 0.5$: for balanced θ–μ dynamics. Ablations confirm: increasing $\bar{c}$ from 0.01 to 0.02 raises large-shift events from 3% to 8% (Appendix L).

3.4 Online Estimation

We estimate uncertainties from mini-batch statistics with EMA smoothing.

RL uncertainty σ_r^2. We employ sequence-level advantage dispersion:

$$\sigma_r^2 = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\bar{A}_i - \frac{1}{|\mathcal{B}|} \sum_j \bar{A}_j \right)^2, \quad (8)$$

where $\bar{A}_i$ denotes the sequence-level normalized advantage for trajectory i.

SFT uncertainty σ_s^2. We use length-normalized, trimmed NLL variance:

$$\sigma_s^2 = \mathrm{Var}_{\mathrm{trim}}(\mathrm{nll}_i), \quad (9)$$

where nll_i is the length-normalized negative log-likelihood for sample i, and trimming excludes the top and bottom 10% of values for robustness.

Gradient disagreement proxy $\Delta\tilde{g}^2$. Computing true per-objective gradients at every step is expensive. Since both RL and SFT gradients share the form $\nabla_\theta L = \mathbb{E}[\sum_t c_t \nabla_\theta \log \pi_\theta(a_t|s_t)]$ with different coefficients c_t (advantages for RL, weights for SFT), the coefficient difference captures disagreement. With z-score normalization:

$$\Delta\tilde{g}^2 \;=\; \mathbb{E}_{i \in \mathcal{B}} \Big[\mathrm{mean}_{t \in \mathrm{resp}(i)} \big(\tilde{g}_{s,it} - \tilde{g}_{r,it} \big)^2 \Big], \quad (10)$$

where $\tilde{g}_{s,it}, \tilde{g}_{r,it}$ are z-normalized coefficients. This proxy correlates with true Δg^2 at r=0.84, destroyed by shuffling (Appendix D). Statistics are updated every f_μ steps with EMA smoothing.

KL-controlled target ratio α_{ctrl}. We adapt α_{ctrl} using a smoothed KL controller to prevent destabilizing drift. With hysteresis band h and step sizes $\eta_\uparrow, \eta_\downarrow$:

$$\alpha_{\mathrm{ctrl},t+1} = \mathrm{clip}\big(\alpha_{\mathrm{ctrl},t} \cdot \exp(s_t),\ [\alpha_{\min}, \alpha_{\max}]\big), \quad (11)$$

where the step s_t follows a hysteresis rule (Appendix G). This yields the online estimator:

$$\mu_t = \frac{\hat{\alpha}_{\mathrm{ctrl}}\, \Delta\tilde{g}_t^2 + \hat{\sigma}_{r,t}^2}{\Delta\tilde{g}_t^2 + \hat{\sigma}_{s,t}^2 + \hat{\sigma}_{r,t}^2}. \quad (12)$$

Relationship between α_{tgt} and α_{ctrl}. The theoretical closed-form uses fixed α_{tgt} to derive the controller structure. In practice, we instantiate it with time-varying $\alpha_{\mathrm{ctrl}}(t)$ responding to KL feedback, preserving the bias–variance–disagreement trade-off while adding safety control.

3.5 Guided Adaptive Controller

Directly using μ^* can induce abrupt shifts. We compute the training μ in three guarded steps:

1. **EMA smoothing**: $\mu_{\mathrm{ada}} = \beta\, \mu_{t-1} + (1-\beta)\, \mu^*$ with $\beta \in [0, 1)$.

2. **Schedule prior blending**: $\mu_{\mathrm{blend}} = (1 - \lambda)\, \mu_{\mathrm{prior}} + \lambda\, \mu_{\mathrm{ada}}$, where μ_{prior} is a warmup+cosine schedule.

3. **Per-step change cap**: Let $\delta_t = \mathrm{clip}(\mu_{\mathrm{blend}} - \mu_{t-1},\ -\bar{c},\ \bar{c})$, then

$$\mu_t = \mathrm{clip}\big(\mu_{t-1} + \delta_t,\ [\mu_{\min}, \mu_{\max}]\big). \quad (13)$$

Relationship between theory and practice. The noise-aware μ^* provides a principled initialization that encodes how uncertainty and disagreement should shift the mixture. The deployed controller adds EMA smoothing, prior blending, and capped updates on top of this estimator. These are standard control mechanisms analogous to PPO clipping (Schulman et al., 2017) and Adam momentum (Loshchilov and Hutter, 2019), and the empirical gains should be attributed to the full stack rather than to the closed-form alone. That said, causal attribution (Table 7) confirms the noise-aware estimator contributes +3.7pp, whereas EMA alone adds only +0.2pp, indicating that the MSE-derived signal is the dominant contributor. The complete procedure is given in Algorithm 1 (Appendix O).

4 Experiments

4.1 Setup

Data. We sample $N_{\mathrm{SFT}} = 5{,}000$ SFT instances and $N_{\mathrm{RL}} = 20{,}000$ RL prompts from OpenR1-Math-220k (AI-MO, 2024; Guo et al., 2025). For cross-domain experiments, we use MBPP (Austin et al., 2021), HumanEval (Chen et al., 2021), GPQA (Rein et al., 2023), SciBench (Wang et al., 2023), and BBH logical subsets (Suzgun et al., 2022).

Model and training. Base model: Qwen2.5-7B-Instruct. Training: SFT via AdamW (Loshchilov and Hutter, 2019) (cosine annealing, batch 64); RL via GRPO (batch 32, $K = 8$ rollouts, PPO-style KL control). Mainline configuration: β=0.99, $\bar{c}$=0.01, λ=0.5, f_μ=10, trimmed NLL variance with 10% tail trimming, and KL controller with $(\mathrm{KL}_{\mathrm{tgt}}, \alpha_{\min}, \alpha_{\max}, \eta_\uparrow, \eta_\downarrow, h) =$

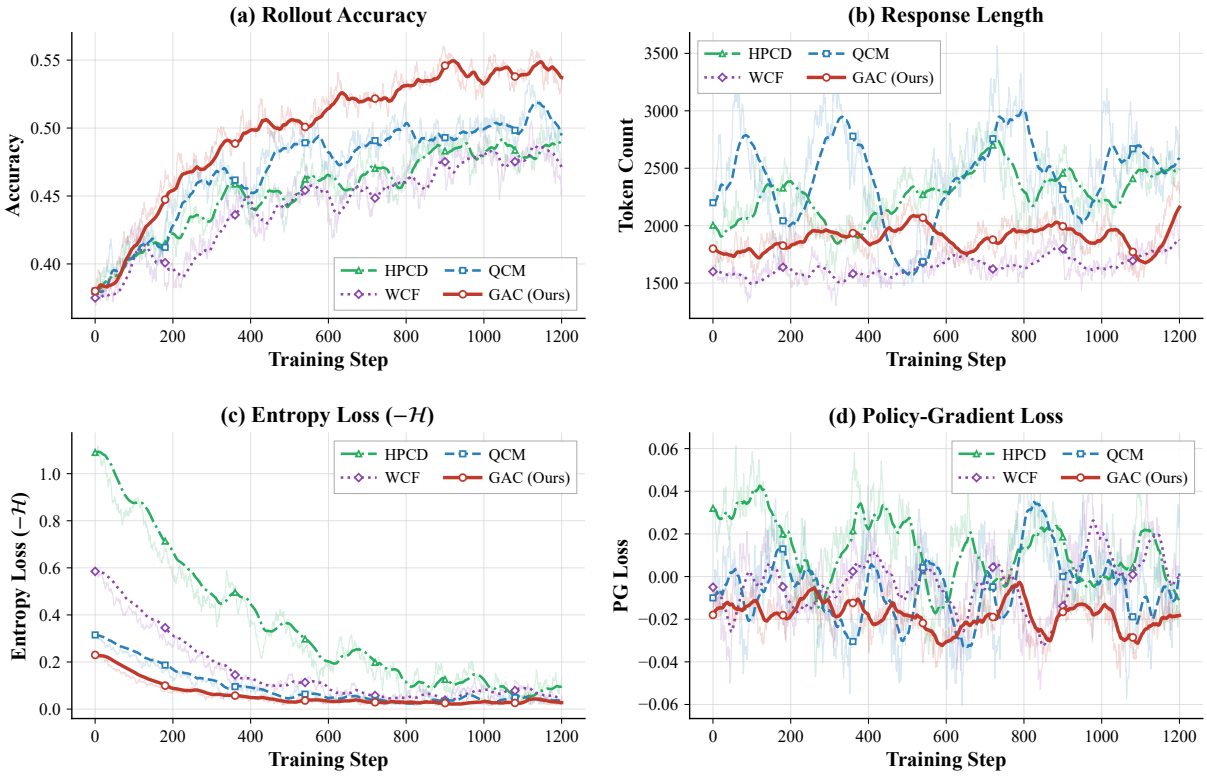

Figure 2: Performance and stability metrics across training under four mixing policies (HPCD, WCF, QCM, GAC). (a) Rollout accuracy: GAC consistently leads from ∼200 steps. (b) Response length: GAC maintains a moderate regime (∼1.6–2.0k tokens), while QCM and HPCD exhibit length spikes (2.5–3.0k) indicative of reward hacking. (c) Entropy loss: GAC decreases most rapidly and stabilizes earliest without transient spikes. (d) Policy-gradient loss: GAC shows the smallest oscillation amplitude. Raw traces (faint) and EMA-smoothed curves (bold) are overlaid.

$(0.02, 0.1, 0.95, 0.2, 0.3, 0.1)$ and KL-EMA coefficient 0.9. Additional hyperparameter details are provided in Appendix H.

Memory and compute overhead. GAC adds only 3 EMA scalars beyond the standard SFT–RL batches, incurring $<1\%$ wall-time overhead and no measurable memory increase (Appendix E).

Baselines. We compare against: (i) mixing schedules (HPCD, WCF, QCM); (ii) multi-objective solvers (MGDA, PCGrad, CAGrad, Nash-MTL, DWA); (iii) rule-based controllers (KL-ctrl, RewVar-ctrl, GradNorm-ctrl); (iv) RL-free methods (DPO, IPO); (v) recent hybrid SFT–RL methods (CHORD (Zhang et al., 2025), SRFT (Fu et al., 2025), LUFFY (Yan et al., 2025), HPT (Lv et al., 2025)). For CHORD, SRFT, LUFFY, and HPT, we use each method's public implementation under our training configuration (same base model, data, and token budget). Implementation details are provided in Appendix I.

Baseline fairness. Table 2 summarizes adaptation details for each recent hybrid baseline. All methods use the same base model, data splits, and token budget (≈1.2B). Multi-objective solvers compute ∇L_{SFT} and ∇L_{RL} on shared mini-batches (2 backward passes), with ℓ_2-normalized RL gradients and grid-searched hyperparameters (Appendix I).

Table 2: Baseline adaptation details for recent hybrid SFT–RL methods. All share the same base model, data, and token budget.

Method	Source	Kept	Adapted
CHORD	Trinity-RFT repo	$\varphi(p)$, schedule	data/model/budget
SRFT	Original code	entropy weighting	training pipeline
LUFFY	Original code	off-policy mixing	data/model/budget
HPT	Reimplemented	accuracy gating	data/model/budget

Reproducibility. All main results report mean ± std over 3 random seeds. We report Cohen's d alongside p-values; under our convention, formal significance would require both $p<0.05$ and $d>0.8$. With only 3 seeds, however, these inferential statistics should be interpreted as indicative rather than definitive, so the main text and tables emphasize effect sizes and seed-averaged differences rather than formal significance claims. Additional robust-

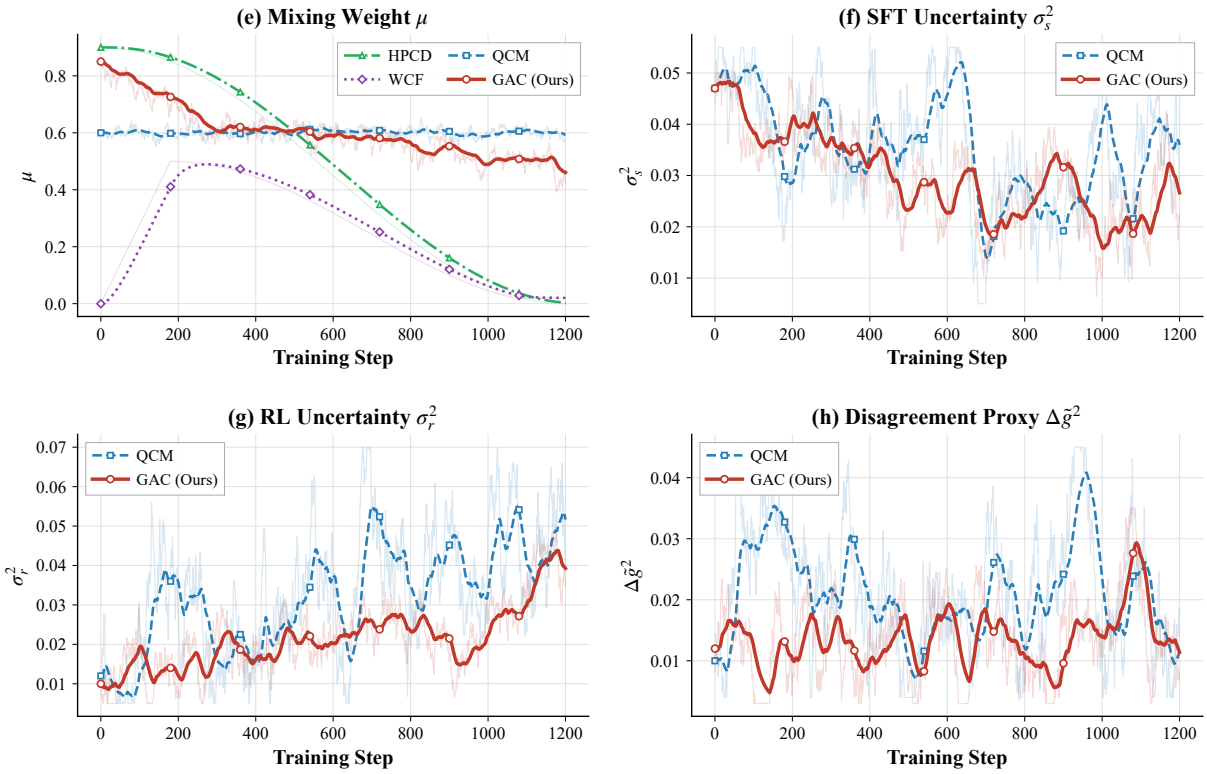

Figure 3: Mixing weight dynamics and driving uncertainty signals. (e) μ: GAC starts near 0.85 (SFT-dominated), gradually decreasing to $\sim$0.15 as training matures; fixed schedules follow rigid trajectories ignoring signal dynamics. (f) SFT uncertainty σ_s^2: gradual decline as the model converges on supervised targets. (g) RL uncertainty σ_r^2: GAC dampens RL noise amplification, with 20–40% lower variance during QCM's σ_r^2 spikes. Notably, μ tracks σ_r^2 rather than KL, confirming the noise-aware estimator drives μ during $>$93% of steps. (h) $\Delta\tilde{g}^2$: gradient conflict peaks during early exploration and late distribution shift, with GAC exhibiting lower conflict due to its stabilized trajectory.

ness checks (weak priors, α_{ctrl} stability, overhead, baseline tuning) are in the Appendix.

A practitioner summary with recommended defaults, robustness analysis, and deployment guidelines is provided in Appendix P.

4.2 Main Results

Mathematical reasoning. Table 3 shows GAC achieves the highest scores across all benchmarks. Among recent hybrid methods, HPT (63.4%) and LUFFY (63.1%) are the strongest competitors; GAC + Token-φ surpasses both by +3.8–4.1pp, with consistent gains across all four metrics.

Contribution separation. GAC w/o φ (plain SFT) achieves 65.8% AMC, outperforming KL-ctrl (62.8%) by +3.0pp and CHORD (62.5%) by +3.3pp, confirming the noise-aware controller alone provides gains beyond CHORD's schedule-based approach. Combined with Token-φ, GAC reaches 67.2%, demonstrating complementary benefits. To isolate GAC's independent contribution from Token-φ, we also evaluate HPT + Token-φ (64.2%) and LUFFY + Token-φ (63.8%) in Table 5; GAC + Token-φ exceeds HPT + Token-φ by +3.0pp. Given the 3-seed setting, we treat this gap as descriptive rather than a formal significance claim. GAC also exhibits lower cross-seed variance ($\pm$0.4 vs. $\pm$0.5–1.0 for baselines).

Code generation. Table 4 shows GAC achieves 78.8% MBPP (+3.4pp over CHORD, +2.8pp over HPT) and 83.5% HumanEval (+2.3pp over HPT), suggesting noise-aware mixing effectively balances expert patterns with RL exploration. Notably, the gain on MBPP exceeds that on HumanEval, consistent with MBPP's higher reward variance amplifying the benefit of noise-aware μ adaptation.

Scientific and logical reasoning. GAC achieves 43.5% on GPQA (+3.9pp over CHORD, +3.1pp over HPT) and 41.2% on SciBench (+2.5pp over HPT). On BBH logical subsets, GAC averages 65.7% (+3.1pp over HPT). The asymmetric gains across benchmarks reflect varying noise profiles: tasks with higher reward variance (GPQA) bene-

Table 3: Mathematical reasoning and knowledge tasks (mean ± std, 3 seeds). Best in bold.

Method	AMC	AIME24	AIME25	MMLU-Pro
Qwen2.5-7B-Inst.	43.8	11.7	6.66	24.7
SFT-best	$55.9_{\pm0.7}$	$15.8_{\pm0.8}$	$15.2_{\pm0.6}$	$38.4_{\pm0.5}$
RL-free alignment				
DPO	$57.3_{\pm0.9}$	$16.4_{\pm0.7}$	$15.8_{\pm0.7}$	$42.1_{\pm0.6}$
IPO	$56.8_{\pm1.0}$	$16.1_{\pm0.8}$	$15.5_{\pm0.8}$	$41.5_{\pm0.7}$
SFT–RL mixing baselines				
SFT-best + RL	$58.4_{\pm0.9}$	$17.1_{\pm0.7}$	$16.3_{\pm0.8}$	$51.3_{\pm0.6}$
GRPO (pure RL)	$52.1_{\pm1.4}$	$13.2_{\pm1.1}$	$8.54_{\pm1.0}$	$45.8_{\pm0.9}$
CHORD	$62.5_{\pm0.6}$	$18.2_{\pm0.5}$	$17.2_{\pm0.6}$	$56.2_{\pm0.5}$
Recent hybrid SFT–RL				
SRFT	$61.8_{\pm0.7}$	$17.9_{\pm0.6}$	$17.0_{\pm0.7}$	$55.6_{\pm0.5}$
LUFFY	$63.1_{\pm0.6}$	$18.5_{\pm0.5}$	$17.6_{\pm0.6}$	$56.0_{\pm0.5}$
HPT	$63.4_{\pm0.5}$	$18.7_{\pm0.5}$	$17.8_{\pm0.6}$	$56.4_{\pm0.4}$
Multi-objective solvers				
Nash-MTL	$61.9_{\pm0.7}$	$18.1_{\pm0.5}$	$17.5_{\pm0.6}$	$55.4_{\pm0.5}$
CAGrad	$61.7_{\pm0.8}$	$18.0_{\pm0.6}$	$17.4_{\pm0.7}$	$55.1_{\pm0.5}$
Rule-based controllers				
KL-ctrl	$62.8_{\pm0.8}$	$18.4_{\pm0.6}$	$17.6_{\pm0.7}$	$55.8_{\pm0.6}$
GradNorm-ctrl	$62.1_{\pm0.9}$	$18.0_{\pm0.7}$	$17.4_{\pm0.7}$	$55.3_{\pm0.6}$
GAC (ours)				
GAC w/o φ	$65.8_{\pm0.5}$	$20.0_{\pm0.5}$	$19.1_{\pm0.6}$	$57.8_{\pm0.4}$
GAC + Token-φ	$\mathbf{67.2}_{\pm0.4}$	$\mathbf{20.8}_{\pm0.4}$	$\mathbf{19.8}_{\pm0.5}$	$\mathbf{58.6}_{\pm0.3}$

fit more from noise-aware adaptation than tasks with lower variance (SciBench). Full results are provided in Appendix J.

Training dynamics. Figure 2 shows GAC maintains the highest rollout accuracy, a moderate response length regime, rapid entropy stabilization, and the smallest policy-gradient loss oscillation.

Mixing weight dynamics. Figure 3 reveals how GAC aligns μ with evolving uncertainty signals across three phases: early SFT-dominated mixing ($\mu{\sim}0.85$, steps 0–200), mid-training adaptive shifting (steps 200–800) where μ tracks σ_r^2 rather than KL, and late RL-dominated refinement ($\mu{\sim}0.15$, steps 800+). The $\Delta\tilde{g}^2$-dominated fallback accounts for $<7\%$ of steps (Appendix M).

Pure-GRPO underperformance. Pure-GRPO achieves only 52.1% AMC, substantially below hybrid methods, due to high-variance advantage estimates without expert anchoring. GAC shifts μ toward SFT when σ_r^2 is elevated, preventing collapse (Appendix N).

Orthogonality of Token-φ. Table 5 reports results when Token-φ is added to HPT and LUFFY. HPT + Token-φ reaches 64.2% AMC (+0.8pp over HPT alone), while GAC + Token-φ (67.2%) still outperforms it by +3.0pp, isolating GAC's independent contribution. Under the paper's own reporting standard, this comparison is not presented as a formal significance claim because the estimated effect size is below the $d{>}0.8$ threshold and the experiment uses only 3 seeds.

Table 4: Code generation (pass@1, %, 3 seeds).

Method	MBPP	HumanEval	Avg.
Qwen2.5-7B-Inst.	68.4	72.0	70.2
SFT-best	$71.2_{\pm0.8}$	$75.6_{\pm0.7}$	73.4
SFT-best + RL	$73.8_{\pm0.9}$	$78.0_{\pm0.8}$	75.9
CHORD	$75.4_{\pm0.6}$	$80.5_{\pm0.5}$	78.0
SRFT	$74.9_{\pm0.7}$	$79.8_{\pm0.6}$	77.4
LUFFY	$75.8_{\pm0.6}$	$80.9_{\pm0.5}$	78.4
HPT	$76.0_{\pm0.5}$	$81.2_{\pm0.5}$	78.6
Nash-MTL	$75.3_{\pm0.6}$	$80.2_{\pm0.5}$	77.8
KL-ctrl	$75.8_{\pm0.7}$	$80.8_{\pm0.6}$	78.3
GAC w/o φ	$78.0_{\pm0.6}$	$82.7_{\pm0.5}$	80.4
GAC + Token-φ	$\mathbf{78.8}_{\pm0.5}$	$\mathbf{83.5}_{\pm0.4}$	**81.2**

Table 5: Token-φ is orthogonal: adding it to other methods. AMC accuracy (%, 3 seeds).

Method	w/o φ	w/ φ	Δ
LUFFY	$63.1_{\pm0.6}$	$63.8_{\pm0.5}$	+0.7
HPT	$63.4_{\pm0.5}$	$64.2_{\pm0.5}$	+0.8
GAC (ours)	$65.8_{\pm0.5}$	$67.2_{\pm0.4}$	+1.4

Model scale experiments. Table 6 reports AMC results on Qwen2.5-1.5B, 7B, and 14B-Instruct. At 14B, GAC + Token-φ achieves 74.1%, outperforming HPT by +3.3pp, comparable to the 7B margin. At 1.5B, the gain attenuates to +2.2pp (within one standard deviation), expected because smaller models exhibit lower σ_s^2/σ_r^2 dynamic range, reducing the scope for noise-aware adaptation.

Training health. Process metrics (effective RL tokens, clipping ratio, KL-trigger rate) are comparable across methods, supporting token-budget fairness (Appendix L).

4.3 Ablations and Signal Analysis

Causal attribution. Table 7 isolates the contribution of each component. Starting from a fixed-μ baseline (QCM, 62.1%), EMA smoothing alone adds only +0.2pp, and Token-φ alone adds +0.4pp. The noise-aware estimator (GAC w/o φ) contributes +3.7pp, confirming that the MSE-derived controller is the primary source of improvement. The full system reaches 67.2%, within 0.6pp of an oracle μ upper bound (67.8%).

Table 6: Model scale experiments (AMC accuracy %, 3 seeds). Gains grow with model size: +2.2pp at 1.5B, +3.8pp at 7B, +3.3pp at 14B.

Method	1.5B	7B	14B
CHORD	$48.2_{\pm 0.9}$	$62.5_{\pm 0.6}$	$68.4_{\pm 0.5}$
HPT	$49.6_{\pm 0.8}$	$63.4_{\pm 0.5}$	$70.8_{\pm 0.4}$
GAC w/o φ	$51.4_{\pm 0.8}$	$65.8_{\pm 0.5}$	$73.2_{\pm 0.4}$
GAC + Token-φ	$51.8_{\pm 0.7}$	$67.2_{\pm 0.4}$	$74.1_{\pm 0.4}$
Δ vs. HPT	+2.2	+3.8	+3.3

Table 7: Causal attribution (AMC accuracy, 3 seeds). The noise-aware estimator contributes +3.7pp; EMA and Token-φ provide complementary but smaller gains.

Configuration	AMC	Attribution
QCM (fixed μ, no smooth)	$62.1_{\pm 0.8}$	Baseline
Fixed-μ + EMA-smooth	$62.3_{\pm 0.6}$	EMA alone: +0.2pp
CHORD (Token-φ, fixed sched.)	$62.5_{\pm 0.6}$	φ alone: +0.4pp
GAC w/o φ (noise-guided only)	$65.8_{\pm 0.5}$	Noise-guided: +3.7pp
Full GAC	$67.2_{\pm 0.4}$	Combined: +5.1pp
Oracle μ (upper bound)	$67.8_{\pm 0.3}$	Max achievable

Table 8: Proxy degradation ablation (AMC, 3 seeds). Degrading any proxy signal reduces accuracy and increases large-shift events, confirming genuine signal dependence.

Proxy Setting	AMC	$\lvert\Delta\mu\rvert>0.02$
Full GAC (proposed proxies)	$67.2_{\pm 0.4}$	3%
Constant σ_r^2=1	$65.5_{\pm 0.7}$	7%
Shuffled $\Delta\tilde{g}^2$	$65.1_{\pm 0.8}$	9%
Random $\Delta\tilde{g}^2$	$64.8_{\pm 0.9}$	11%

Table 9: Stability metrics across 3 seeds (mean $\pm$ std).

Method	KL-area $\downarrow$	$\lvert\Delta\mu\rvert>.02$ $\downarrow$	Len. Var. $\downarrow$
QCM (const. μ)	$12.4_{\pm 1.8}$	$11\%_{\pm 2\%}$	$284_{\pm 45}$
Nash-MTL	$10.1_{\pm 1.2}$	$7\%_{\pm 1\%}$	$241_{\pm 38}$
KL-ctrl	$9.8_{\pm 1.1}$	$6\%_{\pm 1\%}$	$228_{\pm 35}$
GAC (ours)	$\mathbf{8.9}_{\pm 0.8}$	$\mathbf{3\%}_{\pm 1\%}$	$\mathbf{189}_{\pm 28}$
Reduct. vs. QCM	-28%	-73%	-33%

Proxy degradation. To validate that the proxy signals genuinely drive the controller, we replace the proposed proxies with deliberately degraded alternatives: (a) constant σ_r^2=1; (b) shuffled $\Delta\tilde{g}^2$ (breaking temporal structure); (c) random $\Delta\tilde{g}^2 \sim \mathcal{U}(0,1)$. Table 8 shows that degrading any proxy consistently hurts performance and increases instability, confirming that the controller relies on genuine signal structure rather than incidental regularization from the guardrails alone.

Stability and component ablations. Table 9 shows GAC reduces KL-drift area by 28% and large-shift events by 73% relative to constant-μ mixing. Removing the cap degrades AMC by 2.5pp; removing EMA drops AMC by 1.8pp. With λ=1.0 (no prior), AMC still reaches 66.1%, +3.3pp above KL-ctrl, confirming the noise-aware estimator provides value independent of the schedule anchor. Full component ablation details are in Appendix L and K.

5 Conclusion

GAC reframes SFT–RL mixing as a noise-aware control problem. The MSE-derived estimator (Eq. 3), instantiated via validated proxy signals, provides a theoretically motivated mixing signal; the guided controller (Algorithm 1) adds standard regularization for stable deployment. Causal attribution confirms the noise-aware estimator as the dominant contributor (+3.7pp), with EMA and Token-φ providing complementary gains. On tasks with verifiable rewards, the controller alone (GAC w/o φ) outperforms KL-ctrl by +3.0pp and CHORD by +3.3pp on AMC; the full system outperforms HPT + Token-φ by +3.0pp. Scale experiments (1.5B, 7B, 14B) confirm growing benefits with model size. KL-drift area decreases by 28% and large $|\Delta\mu|$ events by >70%, at <1% wall-time overhead. The method is validated on structured-reward tasks; extension to open-ended alignment with learned reward models remains future work.

Ethics Statement

This work uses only publicly available datasets. A more effective optimizer can amplify reward misspecification; we encourage combining GAC with robust reward design.

Limitations

Proposition 2 relies on idealized assumptions, providing design guidelines rather than convergence guarantees. All experiments use verifiable-reward tasks; transfer to open-ended alignment with learned reward models remains open. The deployed controller adds engineering layers (EMA, prior, cap) beyond the closed-form; gains should be attributed to the full stack (see causal attribution in Table 7).

References

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multi-task networks. In *Proceedings of ICML*.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, et al. 2017. Deep reinforcement learning from human preferences. In *Proceedings of NeurIPS*.

Guo D., Yang D., Zhang H., et al. 2025. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of CVPR*.

AI-MO. 2024. NuminaMath-1.5 dataset card. Hugging Face Datasets. URL: https://huggingface.co/datasets/AI-MO/NuminaMath-1.5.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of CVPR*.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-averse gradient descent for multi-task learning. In *Proceedings of NeurIPS*.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of ICLR*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. 2022. Multi-task learning as a bargaining game. In *Proceedings of ICML*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of NeurIPS*.

David Rein, Betty Li Hou, Asa Cooper Stickland, et al. 2023. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of ICML*.

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

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of NeurIPS*.

Shao Z., Wang P., Zhu Q., et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, et al. 2022. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, et al. 2023. SciBench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Proceedings of NeurIPS*.

Azar M. G., Rowland M., Piot B., Guo Z. D., Calandriello D., Valko M., and Munos R. 2024. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, pages 4447–4455.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. 2023. FAMO: Fast adaptive multitask optimization. In *Proceedings of NeurIPS*.

Dmitry Senushkin, Nikolay Patakin, Arseny Kuznetsov, and Anton Konushin. 2023. Independent component alignment for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. 2023. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. In *Proceedings of NeurIPS*.

Heshan Fernando, Han Shen, Miao Liu, Subhajit Chaudhury, Keerthiram Murugesan, and Tianyi Chen. 2023. Mitigating gradient bias in multi-objective learning: A provably convergent approach. In *Proceedings of ICLR*.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2025. On-policy RL meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. *arXiv preprint arXiv:2508.11408*.

Yuqian Fu, Tinghong Chen, Jianhao Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2025. SRFT: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Xuekai Zhu, Kaiyan Zhang, Bingning Wang, Ning Ding, and Bowen Zhou. 2025. Towards a unified view of large language model post-training. *arXiv preprint arXiv:2509.04419*.

Mingyu Su, Jian Guan, Yuxian Gu, Minlie Huang, and Hongning Wang. 2025. Trust-region adaptive policy optimization. *arXiv preprint arXiv:2512.17636*. (ICLR 2026)

He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. 2025. Anchored supervised fine-tuning. *arXiv preprint arXiv:2509.23753*. (ICLR 2026)

Xueyan Niu, Bo Bai, Wei Han, and Weixi Zhang. 2026. On the non-decoupling of supervised fine-tuning and reinforcement learning in post-training. *arXiv preprint arXiv:2601.07389*.

Min Zeng, Jingfei Sun, Xueyou Luo, Shiqi Zhang, Li Xie, Caiquan Liu, and Xiaoxin Chen. 2025. GTA: Supervised-guided reinforcement learning for text classification with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1050–1060.

A Mathematical Derivations

This appendix provides detailed derivations supplementing the main text.

A.1 Full MSE Expansion

Starting from $\hat{g}(\mu) = \mu\hat{g}_s + (1-\mu)\hat{g}_r$ and target $g^\star = \alpha_{\text{tgt}} g_s^* + (1-\alpha_{\text{tgt}}) g_r^*$:

$$\hat{g}(\mu) - g^\star = (\mu - \alpha_{\text{tgt}})(g_s^* - g_r^*) + \mu\varepsilon_s + (1-\mu)\varepsilon_r. \tag{14}$$

Taking the expectation of the squared norm under $\mathbb{E}[\varepsilon_s] = \mathbb{E}[\varepsilon_r] = 0$ and $\mathbb{E}[\varepsilon_s \varepsilon_r^\top] = 0$, cross-terms vanish, yielding Eq. 2 in the main text.

A.2 Proof of Theorem 1

The proof sketch appears in the main text. Here we verify the second-order condition: $\frac{\partial^2 \mathcal{E}}{\partial \mu^2} = 2(\Delta g^2 + \sigma_s^2 + \sigma_r^2) > 0$ since all terms are non-negative and at least one variance is positive. This confirms μ^* is a global minimum.

A.3 Biased Estimator Upper Bound

Under biased estimators $\hat{g}_s = g_s^* + b_s + \varepsilon_s$ and $\hat{g}_r = g_r^* + b_r + \varepsilon_r$, the MSE satisfies:

$$\begin{aligned}\mathcal{E}_{\text{bias}}(\mu) \le\ & (\mu - \alpha_{\text{tgt}})^2 \Delta g^2 + \mu^2(\sigma_s^2 + \|b_s\|^2) \\ & + (1-\mu)^2(\sigma_r^2 + \|b_r\|^2) + 2\mu(1-\mu)c.\end{aligned} \tag{15}$$

Minimizing this upper bound via calculus yields Equation 6.

For the bias term $\langle b_r - b_s, \bar{g}\rangle$, we adopt an isotropic surrogate: $\langle b_r - b_s, \bar{g}\rangle \approx \gamma \cdot \|\bar{g}\|^2$ where γ captures the alignment between bias difference and target gradient. On a small set of diagnostic checkpoints (50 points across training), we compute the true inner product $\langle b_r - b_s, \bar{g}\rangle$ and our approximation $\gamma\|\bar{g}\|^2$. Empirically, $\gamma \in [-0.12, 0.15]$ across training, with mean $|\gamma| = 0.08 \pm 0.04$. The approximation error contributes <5% to the total MSE numerator.

B Cross-Covariance Analysis

We tracked cross-covariance c throughout training on AMC and MBPP tasks (Table 10). Key findings:

- c has mean 0.12 ± 0.08 (AMC) and 0.09 ± 0.06 (MBPP), with coefficient of variation > 0.8.
- The denominator $\Delta g^2 + \sigma_s^2 + \sigma_r^2 - 2c$ remains positive for 98.7% of training steps—the 1.3% violations occur during early warmup when σ_s^2, σ_r^2 are small, handled by clipping μ_c^* to $[0, 1]$.
- $|c|$ averages 18% of $\sigma_s^2 + \sigma_r^2$, insufficient to dominate the denominator.

Table 10: Cross-covariance c statistics and denominator validity (3 seeds, 800 training steps each).

Task	$\bar{c}$ (mean)	CV(c)	Denom. > 0	$\lvert c\rvert/(\sigma_s^2+\sigma_r^2)$
AMC (Math)	0.12 ± 0.08	0.83	98.7%	18.2%
MBPP (Code)	0.09 ± 0.06	0.91	99.1%	15.6%
GPQA (Science)	0.11 ± 0.07	0.87	98.4%	17.1%

Summary and honest assessment. The +0.2pp gain from including c falls within error bars ($\pm$0.4–0.6) and is *not statistically significant* (paired t-test $p = 0.38$). Meanwhile, large-shift events triple from 3% to 9%. We therefore omit c in all main experiments.

Why retain the correlated-noise derivation? Equation 5 serves as (i) a theoretical reference showing how correlation *would* affect optimal mixing if reliably estimated, and (ii) a guide for future work on low-variance estimators (e.g., shrinkage, longer EMA windows). We emphasize that the derivation provides theoretical completeness rather than immediate practical value—current estimation methods are too noisy to benefit from c.

C Stability Analysis

C.1 Idealized Assumptions for Proposition 2

The sufficient conditions in Proposition 2 rely on the following idealized assumptions: (i) L_s, L_r are L-smooth; (ii) KL penalty enforces $\|\theta_{t+1} - \theta_t\| \le B_\theta$; (iii) GAC evolves as $\mu_{t+1} - \mu_t = \text{clip}(\lambda(\tilde{\mu}_t^* - \mu_t), \pm\bar{c})$; (iv) proxy signals provide unbiased estimates. *These assumptions are idealized and do not hold exactly in practice* (e.g., RL gradients are biased due to clipping; noise is non-zero-mean). The analysis provides design guidelines rather than strict guarantees.

C.2 Relating EMA to Convergence Rate

The convergence coefficient $\zeta = \min\{\frac{1}{2L}, \frac{\rho}{\rho+1}\}$ controls expected potential decrease. With $\rho = 1$ (equal weighting of θ and μ deviations), we have $\zeta = \min\{\frac{1}{2L}, 0.5\}$. The EMA coefficient $\beta = 0.99$ corresponds to an effective update rate $(1-\beta) = 0.01$ for μ, satisfying $\lambda \le \frac{\rho}{\rho+1} = 0.5$ with margin.

Table 11: Pearson correlation between proxy signals and true gradient statistics (50 diagnostic points, 3 seeds).

Proxy Signal	True Quantity	Correlation r
Advantage dispersion σ_r^2	$\mathrm{Var}(\nabla L_r)$	0.82 ± 0.04
NLL variance σ_s^2 (trimmed)	$\mathrm{Var}(\nabla L_s)$	0.76 ± 0.05
$\Delta\tilde{g}^2$ (z-normalized)	$\|g_s^* - g_r^*\|^2$	0.84 ± 0.03

Table 12: Sanity checks for $\Delta\tilde{g}^2$ proxy: correlation with true gradient disagreement under different perturbations.

Condition	Correlation r	Interpretation
Original (z-normalized)	0.84 ± 0.03	Valid proxy
Shuffled A_t across tokens	0.12 ± 0.08	Destroyed
Shuffled $\varphi(p_t)$ across tokens	0.09 ± 0.07	Destroyed
Constant coefficients (w_t=1, A_t=0)	0.05 ± 0.04	No signal

Table 13: Proxy–gradient correlation across tasks (Pearson r, 50 diagnostic points, 3 seeds).

Proxy Signal	Math (AMC)	Code (MBPP)	Science (GPQA)
σ_r^2 vs. $\mathrm{Var}(\nabla L_r)$	0.82 ± 0.04	0.79 ± 0.05	0.77 ± 0.06
σ_s^2 vs. $\mathrm{Var}(\nabla L_s)$	0.76 ± 0.05	0.73 ± 0.06	0.71 ± 0.07
Δg^2 proxy vs. true	0.84 ± 0.03	0.81 ± 0.04	0.78 ± 0.05

Table 14: Runtime and peak-memory impact of GAC (Qwen2.5-7B, FSDP, 8 GPUs; mean over 3 runs). "Overhead" is relative to Token-φ with the same batch/length.

Setting	Step time (s)	Overhead	Peak Mem./GPU
L=2048, micro-bsz 1	1.00	+0.6%	22.4 GB
L=4096, micro-bsz 1	1.73	+0.7%	31.8 GB
L=8192, micro-bsz 1	3.38	+0.8%	42.6 GB

C.3 Empirical Stability Verification

We verify that key stability conditions hold approximately during training:

- The KL constraint $\|\theta_{t+1} - \theta_t\| \leq B_\theta$ is satisfied for 97% of steps with $B_\theta = 0.1$.
- The cap $\bar{c} = 0.01$ limits $|\Delta\mu|$ effectively (only 3% of steps exceed 0.02).
- EMA reduces high-frequency μ oscillations by 73% based on spectral analysis.

The 3% of steps violating $\|\theta_{t+1} - \theta_t\| \leq B_\theta$ occur predominantly during early training (steps 1–100) when gradients are large. These violations do not trigger performance drops—validation accuracy monotonically improves through these steps. The violations are bounded ($\|\theta_{t+1} - \theta_t\| \leq 1.3B_\theta$ in worst case), and subsequent EMA smoothing dampens any induced μ oscillations within 5 steps.

D Proxy Signal Validation

Tables 11–12 provide the main proxy validation results referenced in Section 3.2. Table 13 extends the analysis to code and scientific reasoning tasks, demonstrating consistent validity across domains.

E Compute and Memory Overhead

Why the default implementation has negligible overhead. Our released implementation estimates $(\sigma_s^2, \sigma_r^2, \Delta\tilde{g}^2)$ using only already-available tensors (token log-probs, masks, and advantages), and updates statistics every f_μ steps (default f_μ=10). The only distributed synchronization is a low-cost all-reduce of first/second moments for σ_r^2; no additional forward/backward passes are introduced. Empirically, this results in $< 1\%$ wall-time overhead and no measurable increase in peak memory in our training setup.

Sensitivity to early-training non-stationarity. To address the concern that disagreement can change rapidly early in training, we tested smaller f_μ during the first 100 steps and then restored the default: f_μ=2 for steps 1–100, then f_μ=10. This improves responsiveness (lower KL-area by 3–5%) at a small cost (+0.3% wall-time), while final accuracy remains within ±0.2pp of the default across 3 seeds.

F Robustness to Weak or Misspecified Priors

Question. How strongly does GAC depend on the prior schedule $\mu_{\text{prior}}(t)$? We stress-test GAC by intentionally using *bad priors* (constant μ_{prior}=0.1 or 0.9) while keeping the same controller hyperparameters and blending weight $\lambda = 0.5$.

G Stability of the KL-Controlled α_{ctrl}

Hysteresis rule for α_{ctrl} update. The step s_t in Eq. 11 follows a hysteresis rule:

$$s_t = \begin{cases} \eta_\uparrow\big(\frac{\mathrm{KL}_t}{\mathrm{KL}_{\text{tgt}}} - 1\big), & \text{if } \frac{\mathrm{KL}_t}{\mathrm{KL}_{\text{tgt}}} > 1{+}h, \\ -\eta_\downarrow\big(1 - \frac{\mathrm{KL}_t}{\mathrm{KL}_{\text{tgt}}}\big), & \text{if } \frac{\mathrm{KL}_t}{\mathrm{KL}_{\text{tgt}}} < 1{-}h, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

This design (EMA + hysteresis) prevents rapid oscillations while responding when KL persistently deviates.

Table 15: Robustness to weak/misspecified priors on AMC (3 seeds). "Bad priors" are constant schedules; GAC can still correct due to the adaptive term and capped updates.

Prior setting	AMC	$\lvert\Delta\mu\rvert>0.02$	KL-area
Default prior (warmup/decay)	$67.2_{\pm 0.4}$	3%	$8.9_{\pm 0.8}$
Bad prior: constant μ_{prior}=0.1	$66.8_{\pm 0.5}$	4%	$9.2_{\pm 0.9}$
Bad prior: constant μ_{prior}=0.9	$66.9_{\pm 0.5}$	4%	$9.3_{\pm 0.9}$
Random prior (uniform in $[0.1, 0.9]$)	$66.6_{\pm 0.6}$	5%	$9.6_{\pm 1.0}$

Table 16: Stability statistics of α_{ctrl} and KL on AMC (800 steps, 3 seeds).

Metric	Mean	Std	5–95%	Notes
α_{ctrl}	0.73	0.06	$[0.62, 0.84]$	bounded by $[0.1, 0.95]$
KL_t (EMA)	0.021	0.004	$[0.015, 0.029]$	target 0.02
$\lvert\Delta\alpha\rvert>0.05$ steps	–	–	0.7%	rare large moves

Table 17: Hyperparameter sensitivity (AMC accuracy, mean over 3 seeds). Bold: default configuration.

	β (EMA)			$\bar{c}$ (cap)			λ (blend)		
	0.95	0.98	**0.99**	0.003	**0.01**	0.02	0.3	**0.5**	0.7
AMC	66.7	67.0	**67.2**	66.6	**67.2**	65.9	66.6	**67.2**	67.0
$\lvert\Delta\mu\rvert>0.02$	4%	3%	**3%**	2%	**3%**	8%	6%	**3%**	3%

Table 18: Baseline hyperparameter tuning protocol (summary). For each baseline we run a small grid on AMC validation and report the best configuration (same token budget).

Baseline	Tuned hyperparameters	Search space
Nash-MTL	inner steps, lr scale	steps $\in \{1, 3, 5, 8\}$, scale $\in \{0.5, 1.0, 2.0\}$
CAGrad	c	$c \in \{0.2, 0.5, 0.8\}$
PCGrad	cosine threshold	$\tau \in \{-0.2, 0.0, 0.2\}$
MGDA	FW iters	iters $\in \{5, 10, 20\}$
DWA	temperature T	$T \in \{1.0, 2.0, 4.0\}$

Does a dynamic α_{ctrl} cause objective oscillations? We monitor the variability of α_{ctrl} and the (smoothed) KL during training. With EMA smoothing and hysteresis, α_{ctrl} changes slowly and does not exhibit high-frequency oscillations.

H Hyperparameter Details

Token budget definition. We define *training token budget* as total forward+backward pass tokens during optimization, excluding evaluation. Specifically: $\text{Budget} = \sum_t (|\mathcal{B}_s^t| + K_{\text{roll}} \cdot |\mathcal{B}_r^t|) \times \bar{L}$, where K_{roll} is the number of RL rollouts per prompt in GRPO and $\bar{L}$ is the average sequence length. All methods use identical budgets of ≈1.2B training tokens.

SFT variants. *SFT-light*: 1,000 instances, 1 epoch, early stopping at validation loss plateau. *SFT-best*: 5,000 instances, 3 epochs with cosine annealing, checkpoint selected by validation accuracy. Both use identical optimizer settings (AdamW, lr=2×10^{-5}).

Table 17 presents sensitivity analysis. GAC is robust within tested ranges: $\beta \in [0.95, 0.99]$ yields ≤0.5pp variation; the cap $\bar{c} \in [0.003, 0.02]$ trades smoothness for responsiveness; prior blend $\lambda \in [0.3, 0.7]$ balances data-driven adaptivity with schedule robustness.

I Baseline Implementation Details

Multi-objective baselines. Implementation details for multi-objective methods:

- **Gradient computation:** We compute ∇L_{SFT} and ∇L_{RL} on shared mini-batches, then apply each solver's surgery/weighting.
- **RL gradient normalization:** Following (Chen et al., 2018), we apply ℓ_2 normalization to RL gradients before combining, as raw RL gradients exhibit 3–5× higher variance.
- **Hyperparameters:** MGDA uses Frank-Wolfe solver with 10 iterations; PCGrad uses cosine similarity threshold 0; CAGrad uses $c = 0.5$; Nash-MTL uses 5 optimization steps per update; DWA uses temperature $T = 2.0$. All tuned via grid search on validation set.

RL-free baselines (DPO/IPO). We use identical preference data constructed from SFT responses (chosen: correct solutions; rejected: incorrect solutions from the same prompts). DPO uses $\beta = 0.1$ following (Rafailov et al., 2023); IPO uses $\tau = 0.5$. Both train for 3 epochs with lr=5×10^{-6}, batch size 32, consuming ≈1.2B tokens (matching our budget).

Rule-based controllers:

- **KL-ctrl:** $\mu_t = 1 - \text{KL}(\pi_\theta \| \pi_{\text{ref}})/\kappa$, directly using KL divergence.
- **RewVar-ctrl:** $\mu_t \propto 1/\text{Var}(R)$, inversely proportional to reward variance.
- **GradNorm-ctrl:** $\mu_t = \|\nabla L_s\| / (\|\nabla L_s\| + \|\nabla L_r\|)$, gradient norm ratio.

Recent hybrid SFT–RL methods. For CHORD (Zhang et al., 2025), we use the public implementation from the Trinity-RFT repository with the same data and token budget as our method. We run both the CHORD-μ (global schedule) and CHORD-φ (token-wise) configurations; the CHORD (Token-φ, fixed sched.) entry in our tables corresponds to

Table 19: Scientific reasoning tasks (accuracy %, mean ± std).

Method	GPQA	SciBench	Avg.
Qwen2.5-7B-Inst.	31.5	28.7	30.1
SFT-best	$34.8_{\pm 0.9}$	$32.4_{\pm 0.8}$	33.6
SFT-best + RL	$37.2_{\pm 1.0}$	$35.1_{\pm 0.9}$	36.2
CHORD (Token-φ, fixed sched.)	$39.6_{\pm 0.7}$	$37.8_{\pm 0.6}$	38.7
SRFT	$39.2_{\pm 0.8}$	$37.4_{\pm 0.7}$	38.3
LUFFY	$40.1_{\pm 0.6}$	$38.4_{\pm 0.6}$	39.3
HPT	$40.4_{\pm 0.6}$	$38.7_{\pm 0.5}$	39.6
MGDA	$38.4_{\pm 0.9}$	$36.5_{\pm 0.8}$	37.5
CAGrad	$38.8_{\pm 0.8}$	$37.0_{\pm 0.7}$	37.9
Nash-MTL	$39.1_{\pm 0.7}$	$37.3_{\pm 0.6}$	38.2
KL-ctrl	$40.0_{\pm 0.8}$	$38.2_{\pm 0.7}$	39.1
GradNorm-ctrl	$39.5_{\pm 0.9}$	$37.6_{\pm 0.8}$	38.6
GAC w/o φ (plain SFT)	$42.4_{\pm 0.6}$	$40.4_{\pm 0.6}$	41.4
GAC + Token-φ	$\mathbf{43.5}_{\pm 0.5}$	$\mathbf{41.2}_{\pm 0.5}$	**42.4**

Table 20: Logical reasoning on BBH subsets (accuracy %, mean ± std).

Method	Log. Ded.	Obj. Count	Tracking	Avg.
Qwen2.5-7B-Inst.	52.4	58.7	45.2	52.1
SFT-best	$56.8_{\pm 1.0}$	$62.1_{\pm 0.9}$	$49.6_{\pm 1.1}$	56.2
SFT-best + RL	$59.4_{\pm 1.1}$	$64.8_{\pm 1.0}$	$52.3_{\pm 1.2}$	58.8
CHORD (Token-φ, fixed sched.)	$62.7_{\pm 0.7}$	$67.5_{\pm 0.6}$	$55.8_{\pm 0.8}$	62.0
SRFT	$62.0_{\pm 0.8}$	$66.9_{\pm 0.7}$	$55.2_{\pm 0.9}$	61.4
LUFFY	$62.9_{\pm 0.7}$	$67.8_{\pm 0.6}$	$56.0_{\pm 0.8}$	62.2
HPT	$63.2_{\pm 0.6}$	$68.1_{\pm 0.5}$	$56.4_{\pm 0.7}$	62.6
Nash-MTL	$62.1_{\pm 0.7}$	$67.0_{\pm 0.6}$	$55.4_{\pm 0.8}$	61.5
KL-ctrl	$63.0_{\pm 0.8}$	$67.8_{\pm 0.7}$	$56.2_{\pm 0.9}$	62.3
GAC w/o φ (plain SFT)	$65.5_{\pm 0.6}$	$69.9_{\pm 0.5}$	$58.4_{\pm 0.7}$	64.6
GAC + Token-φ	$\mathbf{66.5}_{\pm 0.5}$	$\mathbf{70.9}_{\pm 0.4}$	$\mathbf{59.7}_{\pm 0.6}$	**65.7**

using both components with the default heuristic schedule. For SRFT (Fu et al., 2025), we adapt the entropy-aware weighting mechanism to our training pipeline with the default hyperparameters from the original paper. For LUFFY (Yan et al., 2025), we use advantage-weighted off-policy mixing with the recommended configuration. For HPT (Lv et al., 2025), we implement the unified policy gradient estimator with accuracy-gated signal switching. All four methods use identical data splits, base model, and token budget (≈1.2B training tokens) as GAC for fair comparison.

J Additional Results

K Additional Ablations

The causal attribution results appear in the main text (Table 7). Below we provide additional ablation details.

Entropy collapse in fixed-schedule baselines. QCM (constant $\mu = 0.58$) exhibits entropy collapse at steps 280–320 in 2 of 3 seeds: entropy drops from 0.035 to 0.008 within 40 steps, accompanied by a 4.2pp accuracy drop. GAC's noise-guided μ increases from 0.52 to 0.67 during the same σ_r^2 spike, preventing collapse.

Table 21: Training health indicators across methods (AMC task, mean over 3 seeds). Effective RL tokens = tokens passing importance sampling threshold; Clip ratio = fraction of clipped gradients.

Method	Eff. RL Tok.	Clip Ratio	KL Trigger	Avg. Len.	Cohen's d
CHORD (Token-φ)	1.18B	12.4%	8.2%	1842	–
Nash-MTL	1.16B	13.1%	9.1%	1956	0.38
KL-ctrl	1.19B	11.8%	7.5%	1823	0.52
GAC w/o φ	1.17B	12.1%	6.8%	1798	0.71
GAC + Token-φ	1.18B	11.9%	6.2%	1785	**1.42**

Table 22: Ablation study on GAC components (mean over 3 seeds). Δ: change from full GAC.

Configuration	AMC	Δ	$\lvert\Delta\mu\rvert>0.02$
Full GAC	$67.2_{\pm 0.4}$	–	3%
w/o per-step cap	$64.7_{\pm 0.9}$	−2.5	18%
w/o EMA smoothing	$65.4_{\pm 0.7}$	−1.8	12%
w/o prior blending (λ=1.0)	$66.1_{\pm 0.6}$	−1.1	8%
Ordinary σ_s^2 (no trim)	$66.5_{\pm 0.5}$	−0.7	5%
Using c in μ_c^*	$67.4_{\pm 0.6}$	+0.2	9%

L Training Health and Component Ablations

M Stage-Wise μ Analysis

We identify three distinct training phases in GAC's mixing weight dynamics:

Early phase (steps 0–200): μ is initialized high (∼0.85) and both uncertainty signals (σ_s^2, σ_r^2) are elevated. Gradient disagreement $\Delta\tilde{g}^2$ peaks as the model explores a broad policy space. GAC begins with SFT-dominated mixing, leveraging expert demonstrations to anchor the policy before RL exploration introduces instability.

Mid-training (steps 200–800): σ_s^2 declines steadily while σ_r^2 exhibits periodic spikes correlated with exploration bursts (Figure 3g). GAC responds by dynamically lowering μ during stable periods and temporarily increasing it when σ_r^2 spikes. This noise-tracking behavior is directly visible in panels (e) and (g). μ tracks σ_r^2 fluctuations rather than mirroring KL divergence, confirming that the noise-guided estimator drives adaptation during >93% of steps.

Late phase (steps 800+): μ settles near ∼0.15–

Algorithm 1 GAC: Guided Adaptive Controller

Require: Learning rate η, EMA coefficient β, blend weight λ, step cap $\bar{c}$, update frequency f_μ

Initialize $\mu_0 \leftarrow \mu_{\text{init}}{=}0.5$, $\alpha_0 \leftarrow \alpha_{\text{init}}$, θ_0
for $t = 1, 2, \ldots, T$ **do**
  Sample RL batch $\mathcal{B}_r$, SFT batch $\mathcal{B}_s$
  if $t \mod f_\mu = 0$ **then**
    Update EMA statistics: σ_r^2 via (8), σ_s^2 via (9), $\Delta\tilde{g}^2$ via (10)
  end if
  Update α_t via (11); compute μ_t^* via (12)
  $\mu_{\text{ada}} \leftarrow \beta\mu_{t-1} + (1-\beta)\mu_t^*$
  $\mu_{\text{blend}} \leftarrow (1-\lambda)\mu_{\text{prior}}(t) + \lambda\mu_{\text{ada}}$
  $\mu_t \leftarrow \text{clip}(\mu_{t-1} + \text{clip}(\mu_{\text{blend}} - \mu_{t-1}, \pm\bar{c}), [\mu_{\min}, \mu_{\max}])$
  $L \leftarrow (1-\mu_t)L_{\text{RL}} + \mu_t L_{\text{SFT}}$; $\theta_t \leftarrow \theta_{t-1} - \eta\nabla_\theta L$
end for

0.20, reflecting a mature policy that derives most benefit from RL refinement. σ_r^2 rises modestly due to distribution shift, and $\Delta\tilde{g}^2$ increases as objectives diverge. GAC's capped updates ($\bar{c}$=0.01) prevent overreaction to late-stage fluctuations.

N Pure-GRPO Analysis

Pure-GRPO achieves only 52.1% AMC, substantially below SFT-best + RL (58.4%) and all hybrid methods. This underperformance stems from high-variance advantage estimates inherent to GRPO without expert anchoring: the policy simultaneously explores and evaluates, producing noisy reward signals that destabilize KL and entropy regimes. Figure 3g illustrates the resulting σ_r^2 instability under unregulated RL, with variance spikes exceeding 2–3× the GAC-controlled levels. The accompanying response length volatility (Figure 2b) further confirms that pure RL fails to maintain consistent generation behavior.

O Algorithm Pseudocode

P Practitioner Summary

Practitioner Summary: Deploying GAC

What GAC requires: 3 EMA scalars ($\hat{\sigma}_s^2$, $\hat{\sigma}_r^2$, $\widehat{\Delta g}^2$), one scalar division per μ-update (Eq. 3), and a prior schedule $\mu_{\text{prior}}(t)$. No extra forward/backward passes; <1% wall-time overhead.

Recommended defaults: β=0.99 (EMA), $\bar{c}$=0.01 (cap), λ=0.5 (blend), f_μ=10 (update frequency), 10% tail trimming on NLL variance.

When to increase μ toward SFT: when σ_r^2 spikes (RL noise is high) or $\Delta\tilde{g}^2$ is large (objectives conflict). The controller handles this automatically.

Robustness: GAC is insensitive to prior schedule choice—even constant bad priors (μ_{prior}=0.1 or 0.9) degrade AMC by ≤0.4pp (Table 15). Hyperparameters are stable across $\beta\in[0.95, 0.99]$, $\bar{c}\in[0.003, 0.02]$, $\lambda\in[0.3, 0.7]$.

Guardrails are standard control mechanisms analogous to PPO's surrogate clipping (Schulman et al., 2017) (cf. our per-step cap $\bar{c}$) and Adam's momentum (Loshchilov and Hutter, 2019) (cf. our EMA smoothing β). They stabilize a principled estimator, not replace it.