BV-Blend: Uncertainty-Weighted Historical Baselines for Stable Critic-Free RL with Verifiable Rewards

Yupeng Chang[1] **Yuan Wu**[1*] **Yi Chang**[1,2,3]
[1]School of Artificial Intelligence, Jilin University
[2]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China
[3]International Center of Future Science, Jilin University
changyp23@mails.jlu.edu.cn, {yuanwu, yichang}@jlu.edu.cn

Abstract

Critic-free reinforcement learning with verifiable rewards (RLVR), exemplified by Group Relative Policy Optimization (GRPO), avoids training a value function (critic) and reduces memory and compute overhead relative to critic-based PPO pipelines for aligning large language models. However, GRPO-style advantage estimation depends on prompt-local (within-prompt-group) reward statistics and can be unstable. In particular, when all rollouts in a prompt group receive identical rewards, the within-group reward variance becomes zero, and group normalization yields *zero* advantages for that group, impeding learning in cold-start regimes with binary verifiers. We introduce **BV-Blend**, a critic-free framework that stabilizes advantage estimation by combining prompt-local on-policy statistics with semantic-cluster-conditioned historical moments. BV-Blend maintains EMA-tracked reward moments for each cluster, derives a confidence weight from a standard error of the mean (SEM) proxy, and uses this weight to blend historical and prompt-local baseline and variance statistics into a standardized advantage for PPO-style clipped updates. Experiments on verifiable reasoning benchmarks show that BV-Blend improves training stability and performance, and remains robust in regimes where group-normalized methods may stall.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has recently become a practical paradigm for aligning Large Language Models (LLMs) in domains with *objective* correctness signals, such as mathematics and code, where outputs can be automatically verified (e.g., exact match against a reference answer, formal proof checking, or passing unit tests) (Yu et al., 2025b; Yan et al., 2025; Gui et al., 2024). Compared with process-level supervision such as Process Reward Models (PRMs) (Lightman et al., 2023), RLVR optimizes outcome-based rewards and does not require scoring intermediate reasoning steps. When verifiers are reliable, collecting multiple rollouts per prompt naturally improves exploration and increases the chance of observing correct solutions, which has underpinned several recent reasoning-focused RL systems (Guo et al., 2025). Beyond structured domains, RLVR has also been extended to more diverse settings when reference signals or robust verifiers are available (Su et al., 2025).

However, directly applying the de facto RLHF recipe—PPO with a learned critic (Ziegler et al., 2019; Ouyang et al., 2022)—to RLVR exposes a practical tension. With sparse, trajectory-level verifiable rewards, learning an accurate value function can be challenging, while the additional critic introduces non-trivial memory and compute overhead, as well as additional training complexity, at LLM scale (Ouyang et al., 2022). These considerations have accelerated interest in critic-free alignment. Preference-optimization methods such as DPO (Rafailov et al., 2024) avoid explicit reward modeling and online RL, but are primarily studied under preference or utility supervision rather than settings where informative, programmatic outcome rewards are directly available. We therefore focus on *direct-reward, critic-free* policy optimization, where GRPO (Shao et al., 2024) is a prominent baseline that derives advantages through prompt-local reward normalization within each prompt group.

Despite its simplicity, prompt-local normalization can be unstable. The resulting advantage estimator may depend strongly on transient within-group statistics, leading to high variance and unstable learning dynamics, which has motivated a growing body of analyses and fixes (Liang, 2025; Chen et al., 2025; Mroueh, 2025; Yu et al., 2025a). Moreover, RLVR commonly encounters a cold-

*Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 1706–1725

start regime in which the current policy produces predominantly incorrect solutions; under binary verifiers, many prompt groups become effectively deterministic (e.g., all failures), yielding (near-)zero within-group reward variance. In this *zero-variance* regime, standard within-group normalization produces (near-)zero advantages, removing the learning signal for those prompt groups (Le et al., 2025).

To address this instability, we propose **BV-Blend**, a critic-free framework that stabilizes advantage estimation by combining (i) an on-policy prompt-local signal with (ii) low-variance historical moments aggregated over semantically similar prompts. BV-Blend computes an uncertainty-aware confidence weight from a standard-error-of-the-mean (SEM) proxy of the historical statistics, downweighting unreliable historical information while relying more on it when the historical estimate is better supported. As a result, BV-Blend mitigates advantage collapse on zero-variance prompt groups without training a critic and remains compatible with PPO-style optimization.

Our contributions are:

- We analyze the *zero-variance* failure mode of prompt-local (group-normalized) advantage estimation in direct-reward, critic-free RL, where the normalized learning signal vanishes when all rollouts in a prompt group receive identical rewards.
- We introduce **BV-Blend**, an uncertainty-aware historical blending mechanism that stabilizes prompt-local advantage estimation by leveraging semantic-cluster-conditioned historical moments, without training a critic.
- We empirically demonstrate that BV-Blend improves training stability and performance on verifiable reasoning benchmarks, and remains robust in regimes where standard group-normalized methods may stall.

2 Related Work

Critic-based and preference-based alignment. Policy-gradient methods typically reduce variance with a learned value-function baseline; GAE (Schulman et al., 2015) (commonly paired with PPO (Schulman et al., 2017)) yields lower-variance advantages than Monte Carlo estimators such as REINFORCE (Williams, 1992). In RLHF-style LLM alignment, PPO pipelines often train an auxiliary value head alongside the policy during RL fine-tuning (Ouyang et al., 2022). A separate line of work avoids online RL and critic training by directly optimizing objectives from preference or utility supervision, e.g., DPO (Rafailov et al., 2024), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024). These methods are primarily studied under preference or utility supervision (or implicit rewards), rather than explicit outcome-based verifiable rewards.

Direct-reward, critic-free policy optimization. Our work builds on critic-free policy optimization with verifiable rewards, exemplified by Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO avoids a learned critic by normalizing rewards within each prompt group, but the learning signal can be sensitive to transient within-group statistics and may collapse when within-group reward variance is near zero. Recent work has explored related issues and remedies, including correcting optimization bias in GRPO-style objectives (Dr. GRPO) (Liu et al., 2025a), temporally smoothing baselines via lightweight Bayesian or Kalman-style updates (KRPO) (Wang et al., 2025), extending GRPO-style optimization to multi-turn tool-use settings (ARPO) (Dong et al., 2025), and extracting learning signals from zero-variance prompts (RL-ZVP) (Le et al., 2025). Other work also mitigates zero-variance or low-signal training regimes through data- or sampling-level interventions, such as dynamic sampling (Yu et al., 2025a).

Our method is most closely related to approaches that stabilize GRPO-style advantage estimation without training a critic. Compared with KRPO, which primarily smooths statistics across training steps, BV-Blend maintains historical reward moments conditioned on semantic clusters of prompts. Compared with RL-ZVP, which specifically targets zero-variance prompts, BV-Blend uses uncertainty-aware blending of prompt-local and cluster-conditioned historical statistics to form a unified standardized advantage for all prompt groups. Compared with dynamic-sampling-based remedies, BV-Blend operates at the level of the advantage estimator rather than the data-selection policy. Overall, BV-Blend is a critic-free method that modifies the advantage estimator while remaining compatible with standard PPO-style training pipelines.

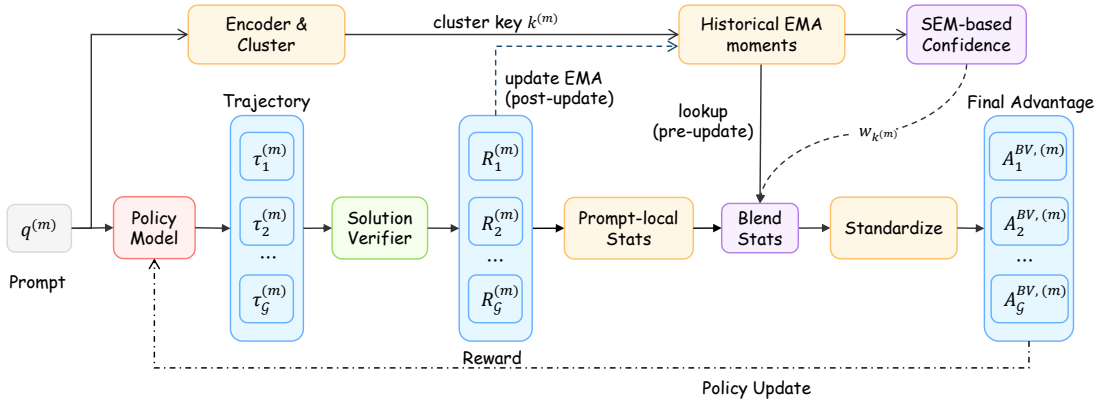


Figure 1: **BV-Blend overview.** For each prompt $q^{(m)}$, we sample G trajectories $\{\tau_i^{(m)}\}$ with the behavior policy and obtain verifier rewards $\{R_i^{(m)}\}$. We compute prompt-local statistics $(\mu_{\mathcal{G}}^{(m)}, \sigma_{\mathcal{G}}^{(m)})$, embed $q^{(m)}$, and assign it to a semantic cluster $k^{(m)}$. Using *pre-update* EMA moments $(\mu_{\text{hist}}(k), v_{\text{hist}}(k), N_k^{\text{eff}})$, we compute the SEM-based confidence w_k (Eq. (9); cold start: $w_k{=}0$ for unseen clusters), blend baseline and variance statistics to obtain $(b^{(m)}, s^{(m)})$ (Eq. (10)), and form advantages $A_i^{\text{BV},(m)}$ (Eq. (11)) for a PPO-style update. EMA moments are updated *post-update* using the current batch.

3 Method

Critic-free policy optimization for LLM alignment often relies on prompt-local (group-dependent) advantage normalization (e.g., GRPO-style estimators), but when within-group reward dispersion is small, prompt-local standardization can yield near-zero advantages and effectively remove the learning signal for that prompt group. We propose **BV-Blend**, which constructs a *single* per-trajectory advantage by combining prompt-local statistics with semantic-cluster-conditioned historical moments, using a confidence weight derived from a standard-error-of-the-mean (SEM) proxy. Concretely, BV-Blend blends baseline and variance statistics, then standardizes once using the resulting scale. BV-Blend keeps the PPO-style clipped objective unchanged (Schulman et al., 2017) and modifies only the advantage estimator: the SEM weight is computed from *pre-update* EMA statistics, while EMA moments are updated *post-update* using the current batch (Fig. 1). As in other group-based normalization methods, we stop gradients through all reward statistics and do not assume the resulting normalized estimator is unbiased.

3.1 Background: prompt-local normalization in critic-free RL

A training batch contains M prompts $\{q^{(m)}\}_{m=1}^{M}$. For each prompt $q^{(m)}$, the behavior policy $\pi_{\theta_{\text{old}}}$ samples G trajectories $\mathcal{G}(q^{(m)}) = \{\tau_i^{(m)}\}_{i=1}^{G}$. Each trajectory $\tau_i^{(m)} = (y_{i,1}^{(m)}, \ldots, y_{i,T_i^{(m)}}^{(m)})$ receives a scalar trajectory-level reward $R_i^{(m)}$ from an external verifier.

For token position t in $\tau_i^{(m)}$, define the importance ratio

$$r_{i,t}^{(m)}(\theta) = \frac{\pi_\theta\left(y_{i,t}^{(m)} \mid q^{(m)}, y_{i,<t}^{(m)}\right)}{\pi_{\theta_{\text{old}}}\left(y_{i,t}^{(m)} \mid q^{(m)}, y_{i,<t}^{(m)}\right)}, \tag{1}$$

and its clipped version $\tilde{r}_{i,t}^{(m)}(\theta) = \text{clip}\big(r_{i,t}^{(m)}(\theta), 1-\epsilon, 1+\epsilon\big)$. Since rewards are trajectory-level, we compute a single advantage $A_i^{(m)}$ per trajectory and apply it to all completion tokens. We restrict optimization to completion tokens using a mask $m_{i,t}^{(m)} \in \{0,1\}$ and define masked token means as

$$\mathbb{E}_{t\sim\tau_i^{(m)}}[f_{i,t}] \triangleq \frac{\sum_{t=1}^{T_i^{(m)}} m_{i,t}^{(m)} f_{i,t}}{\sum_{t=1}^{T_i^{(m)}} m_{i,t}^{(m)}}, \tag{2}$$

where each completion contains at least one generated token so the denominator is non-zero. Throughout, $\mathbb{E}_{t\sim\tau_i^{(m)}}[\cdot]$ denotes a *mean* over completion tokens (Eq. (2)) to avoid length-dependent gradient scaling; concretely, $m_{i,t}^{(m)} = 1$ for generated completion tokens up to (and including) EOS, and 0 for prompt tokens and padding. In implementation, $A_i^{(m)}$ is treated as a trajectory-level constant (no gradient through reward statistics).

GRPO-style training forms a prompt-local standardized advantage (Shao et al., 2024):

$$A_i^{\text{GRPO},(m)} = \frac{R_i^{(m)} - \mu_{\mathcal{G}}^{(m)}}{\sigma_{\mathcal{G}}^{(m)} + \delta}, \tag{3}$$

where $\mu_{\mathcal{G}}^{(m)}$ and $\sigma_{\mathcal{G}}^{(m)}$ are the mean and standard deviation of $\{R_i^{(m)}\}_{i=1}^{G}$, and $\delta > 0$ is a small constant. If $G < 2$, we set $\sigma_{\mathcal{G}}^{(m)} = 0$, yielding $A_1^{\text{GRPO},(m)} = 0$. The clipped surrogate objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_m\Bigg[\frac{1}{G}\sum_{i=1}^{G}\mathbb{E}_{t\sim\tau_i^{(m)}}\Big[\min\Big(r_{i,t}^{(m)}(\theta)\,A_i^{\text{GRPO},(m)}, \tilde{r}_{i,t}^{(m)}(\theta)\,A_i^{\text{GRPO},(m)}\Big)\Big]\Bigg]. \tag{4}$$

When $\sigma_{\mathcal{G}}^{(m)} \approx 0$, Eq. (3) collapses toward zero advantages, effectively removing the learning signal for that prompt group.

3.2 BV-Blend: SEM-driven blending of historical and prompt-local baselines

BV-Blend computes a single standardized advantage $A_i^{\text{BV},(m)}$ by blending *baseline and variance statistics* and then standardizing once (taking a square root to obtain the scale). It follows the common form $A = (R - b)/s$: GRPO uses prompt-local $(b, s) = (\mu_{\mathcal{G}}^{(m)}, \sigma_{\mathcal{G}}^{(m)})$, whereas BV-Blend interpolates between prompt-local statistics and *semantic-cluster-conditioned* EMA moments, with the interpolation controlled by a SEM-based confidence w_k.

3.2.1 Prompt clustering

Each prompt $q^{(m)}$ is embedded by a frozen encoder $E(\cdot)$ and assigned to a fixed K-means codebook $\{c_j\}_{j=1}^{K}$:

$$k^{(m)} = \arg\min_{j\in\{1,\dots,K\}} \|E(q^{(m)}) - c_j\|_2^2. \tag{5}$$

The codebook is trained offline on a representative prompt corpus and kept fixed during RL to avoid cluster-identity drift. We report the encoder choice, K, codebook training corpus, and K-means settings (implementation, seed, and iterations) for reproducibility.

3.2.2 Historical statistics with EMA moments

For each cluster k, we maintain EMA moments: mean $m_1(k)$, second raw moment $m_2(k)$, and EMA mass N_k^{eff} (here m_1, m_2 denote moments, not the prompt index m). Given batch $\mathcal{B}$, let $\mathcal{I}_{\mathcal{B}}(k) = \{(m,i) : k^{(m)} = k\}$ and $N_{\mathcal{B}}(k) = |\mathcal{I}_{\mathcal{B}}(k)|$. Define sufficient statistics

$$S_{1,\mathcal{B}}(k) = \sum_{(m,i)\in\mathcal{I}_{\mathcal{B}}(k)} R_i^{(m)},$$
$$S_{2,\mathcal{B}}(k) = \sum_{(m,i)\in\mathcal{I}_{\mathcal{B}}(k)} \big(R_i^{(m)}\big)^2.$$

In distributed training, we aggregate $\{S_{1,\mathcal{B}}(k), S_{2,\mathcal{B}}(k), N_{\mathcal{B}}(k)\}$ across workers and apply the EMA update *after* the policy optimization step. If $N_{\mathcal{B}}(k) > 0$, define the batch mean and second raw moment

$$\mu_{\mathcal{B}}(k) = \frac{S_{1,\mathcal{B}}(k)}{N_{\mathcal{B}}(k)}, \qquad \mu_{2,\mathcal{B}}(k) = \frac{S_{2,\mathcal{B}}(k)}{N_{\mathcal{B}}(k)}.$$

With EMA rate $\gamma \in (0, 1]$,

$$\begin{aligned} m_1(k) &\leftarrow (1-\gamma)m_1(k) + \gamma\,\mu_{\mathcal{B}}(k), \\ m_2(k) &\leftarrow (1-\gamma)m_2(k) + \gamma\,\mu_{2,\mathcal{B}}(k), \\ N_k^{\text{eff}} &\leftarrow (1-\gamma)N_k^{\text{eff}} + \gamma\,N_{\mathcal{B}}(k). \end{aligned} \tag{6}$$

We define

$$\begin{aligned} v_{\text{hist}}(k) &= \max\big(m_2(k) - m_1(k)^2,\, 0\big), \\ \sigma_{\text{hist}}(k) &= \sqrt{v_{\text{hist}}(k)}, \\ \mu_{\text{hist}}(k) &= m_1(k). \end{aligned} \tag{7}$$

Initialization and cold start. On first observation of cluster k, we initialize $N_k^{\text{eff}} \leftarrow N_0,\ m_1(k) \leftarrow \mu_{\mathcal{B}}(k),\ m_2(k) \leftarrow \mu_{\mathcal{B}}(k)^2 + V_{\text{prior}}$, with $N_0 > 0$ and $V_{\text{prior}} > 0$.

Update ordering. For each batch, we compute w_k using EMA statistics *before* incorporating the current batch. If a cluster k is first observed in the current batch (i.e., no prior EMA state exists at advantage-computation time), we set $w_k = 0$ for this batch (pure prompt-local normalization for A^{BV}) and create its EMA state using the initialization above *after* the policy optimization step. For previously seen clusters, EMA moments are updated *after* the policy optimization step using the aggregated sufficient statistics.

3.2.3 Uncertainty-to-confidence mapping

We quantify historical uncertainty using a SEM-style proxy

$$\text{SEM}_{\text{hist}}(k) = \frac{\sigma_{\text{hist}}(k)}{\sqrt{N_k^{\text{eff}} + \delta_N}}, \tag{8}$$

with $\delta_N > 0$. We map uncertainty to a confidence weight

$$w_k = \exp\left(-\frac{\text{SEM}_{\text{hist}}(k)}{T}\right), \tag{9}$$

where $T > 0$ controls sensitivity to the reward scale. For clusters with an EMA state, $w_k \in (0, 1]$: lower uncertainty yields larger w_k (more reliance on historical moments), while higher uncertainty yields smaller w_k (more reliance on prompt-local statistics). We report $T, \gamma, N_0, V_{\text{prior}}, \delta_N$ in experiments.

3.2.4 Baseline-and-scale blending and BV advantage

For each prompt group m, compute prompt-local statistics $\mu_{\mathcal{G}}^{(m)}$ and $\sigma_{\mathcal{G}}^{(m)}$ over $\{R_i^{(m)}\}_{i=1}^{G}$. Let $w^{(m)} = w_{k^{(m)}}$. We blend baseline and variance statistics, and take a square root to obtain the scale:

$$b^{(m)} = w^{(m)}\,\mu_{\text{hist}}(k^{(m)}) + \big(1 - w^{(m)}\big)\,\mu_{\mathcal{G}}^{(m)},$$
$$s^{(m)} = \sqrt{w^{(m)}\,v_{\text{hist}}(k^{(m)}) + \big(1 - w^{(m)}\big)\big(\sigma_{\mathcal{G}}^{(m)}\big)^2}, \tag{10}$$

and define the BV-Blend advantage

$$A_i^{\text{BV},(m)} = \frac{R_i^{(m)} - b^{(m)}}{s^{(m)} + \delta}. \tag{11}$$

When $\sigma_{\mathcal{G}}^{(m)} \approx 0$, the blended scale $s^{(m)}$ remains non-degenerate whenever the historical term provides non-zero variance, thereby preventing collapse of the learning signal.

4 Experiments

Domain and Datasets. We focus on mathematical reasoning, where evaluation is rigorous and objective. Compared with more subjective tasks, math offers (i) deterministic, programmatic verification against ground-truth final answers, reducing reliance on preference labels and learned judges that may be biased or exploitable (Huang et al., 2025; Zheng et al., 2025); (ii) scalable automated evaluation without costly human annotation; and (iii) diverse multi-step problems with unambiguous correctness criteria. Our training set contains 45,000 problems curated from the default 94k split of OpenR1-Math-220k (Face, 2025; Yan et al., 2025). OpenR1-Math-220k is built from NuminaMath 1.5 prompts and includes 2–4 reasoning traces generated by DeepSeek-R1; most traces are verified by *Math-Verify* (with a small portion additionally judged by an LLM), and each problem has at least one correct trace (Face, 2025; LI et al., 2024; Guo et al., 2025). We use *Math-Verify* to remove instances with invalid or unverifiable final answers, and further filter samples whose traces exceed 8,192 tokens. This curated set is used as (1) prompts for on-policy rollouts, (2) ground-truth answers for deterministic reward computation, and (3) a corpus of reasoning traces for training SFT baselines.

Evaluation. We evaluate BV-Blend on benchmarks probing two complementary aspects: (i) *in-domain* mathematical reasoning across a broad difficulty range—from pre-college competition problems (AMC (He et al., 2024)), through university-level problem solving (MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022)), to elite competition-level challenges (AIME 2024/2025 (Li et al., 2024)); and (ii) *out-of-distribution* (OOD) generalization to non-mathematical benchmarks, including abstract reasoning (ARC-C (Clark et al., 2018)), expert-level scientific knowledge (GPQA-diamond (Rein et al., 2023)), and multidisciplinary problem solving (MMLU-Pro (Wang et al., 2024)). Our main results use Qwen2.5-Math-7B (Yang et al., 2024c) as the backbone; additional backbones are reported in the *Implementation Details* paragraph below. Unless otherwise specified, we decode for evaluation with temperature 0.6. For multiple-choice benchmarks, we randomly permute answer options for each question (with labels remapped accordingly) to mitigate position bias. We report pass@1 on large-scale benchmarks (MATH-500, Minerva, and the OOD suite). For smaller and more challenging test sets (AIME and AMC), we instead report avg@32, defined as the mean success rate over 32 independent samples per problem, which is more stable under stochastic decoding.

Baselines. To contextualize BV-Blend, we compare against baselines built on the same Qwen2.5-Math-7B backbone (Table 1), grouped into two categories. **(1) On-policy RLVR methods** start from the base model and optimize using only on-policy rollouts with verifiable rewards (i.e., without SFT initialization or off-policy demonstrations). This category includes our replication of GRPO (Shao et al., 2024) and representative "Zero" RLVR systems trained under different reward designs and training recipes: SimpleRL-Zero (Zeng et al., 2025), Open-Reasoner-Zero (Hu et al., 2025), PRIME-Zero (Cui et al., 2025), and Oat-Zero (Liu et al., 2025b). **(2) Hybrid & off-policy methods** leverage additional external data beyond on-policy rollouts, including SFT, a sequential SFT→RL pipeline, ReLIFT (Ma et al., 2025),

Table 1: Main results on Qwen2.5-Math-7B. We compare BV-Blend with baselines on mathematical reasoning and OOD generalization benchmarks. Best and second-best in each column are in **bold** and underlined. Math Reasoning Avg. averages AIME 2024/2025, AMC, MATH-500, Minerva, and Olympiad; Generalization Avg. averages ARC-C, GPQA*, and MMLU-Pro.

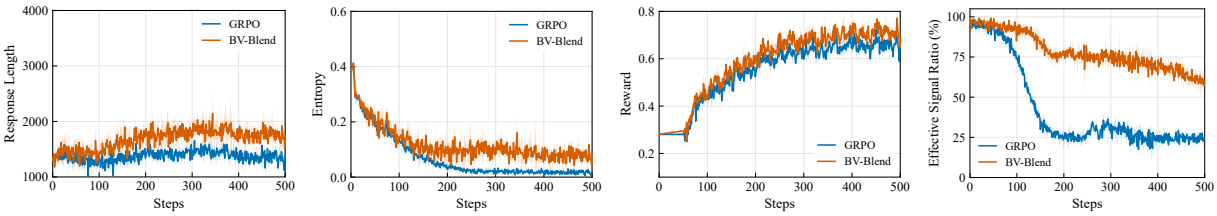
Model	Math Reasoning Performance						Generalization Performance			
	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	Avg.	ARC-C	GPQA*	MMLU-Pro	Avg.
Qwen-Base	11.5/4.9	31.3	43.6	7.4	15.6	19.1	18.2	11.1	16.9	15.4
Qwen-Instruct	12.5/10.2	48.5	80.4	32.7	41.0	37.6	70.3	24.7	34.1	43.0
Baselines: On-Policy RLVR										
GRPO (our replication)	25.1/15.3	62.0	84.4	39.3	46.8	45.5	82.3	40.4	49.3	57.3
SimpleRL-Zero	27.0/6.8	54.9	76.0	25.0	34.7	37.4	30.2	23.2	34.5	29.3
OpenReasoner-Zero	16.5/15.0	52.1	82.4	33.1	47.1	41.0	66.2	29.8	**58.7**	51.6
PRIME-Zero	17.0/12.8	54.0	81.4	39.0	40.3	40.8	73.3	18.2	32.7	41.4
Oat-Zero	33.4/11.9	61.2	78.0	34.6	43.4	43.8	70.1	23.7	41.7	45.2
Baselines: Hybrid & Off-Policy Methods										
SFT	22.2/22.3	52.8	82.6	40.8	43.7	44.1	75.2	24.7	42.7	47.5
SFT+RL	25.8/23.1	62.7	87.2	39.7	50.4	48.2	72.4	24.2	37.7	44.8
ReLIFT	28.2/20.1	64.9	87.4	33.8	52.5	47.8	76.2	37.9	52.5	55.5
LUFFY	29.4/23.1	65.6	87.6	37.5	57.2	50.1	80.5	39.9	53.0	57.8
LUFFY†	30.7/**25.5**	66.2	86.8	**41.2**	55.3	51.0	81.8	49.0	54.7	61.8
BV-Blend (*ours*)	**34.2**/23.6	**66.5**	**87.9**	40.7	**57.4**	**51.7**	**83.1**	**52.6**	56.5	**64.1**

and LUFFY (Yan et al., 2025). For LUFFY, we report both the standard and extended-training (†) variants when available.

Implementation Details. We evaluate BV-Blend across multiple backbones. Our primary testbed is Qwen2.5-Math-7B to align with prior RLVR work, and we additionally report results on Qwen2.5-Math-1.5B, Qwen2.5-7B-Instruct, and LLaMA-3.1-8B to assess robustness across model scales and instruction tuning. Unless otherwise specified, we use a shared training setup across all runs. We optimize with AdamW and a cosine learning-rate schedule with linear warmup, peaking at 1×10^{-6}. Each iteration collects a global rollout batch of 128 trajectories (16 prompts $\times$ 8 rollouts) with sampling temperature 1.0, and performs policy updates with a trajectory minibatch size of 64. Rewards are binary: we assign $+1$ if the extracted final answer is verified correct by *Math-Verify*, and 0 otherwise, with no intermediate or format-based shaping rewards. We use a PPO-style clipped objective shared by BV-Blend and PPO-style baselines, with clipping $\epsilon = 0.2$ and an entropy bonus of 0.01. Unless otherwise stated, we include a KL-to-reference term in the objective but set its coefficient to $\beta = 0$ in our main experiments, effectively disabling it to isolate the effect of different advantage estimators. For BV-Blend, we maintain cluster-conditioned historical moments using EMA with update rate $\gamma = 0.9$, and use $\delta = 10^{-8}$ for numerical stability in advantage computation. The remaining BV-Blend hyperparameters $(T, \delta_N, N_0, V_{\text{prior}})$ and clustering settings (encoder E, codebook size K, and codebook training corpus) follow Sec. 3.2; their concrete values are provided in [the corresponding paragraph / appendix / supplementary section]. We do not apply any additional global advantage normalization beyond the method-specific estimator (e.g., GRPO or BV-Blend). For purely on-policy approaches (including BV-Blend and our GRPO replication), all 8 rollouts per prompt are generated on-policy from the current policy. All models are trained for 500 iterations. All experiments are conducted on 8 NVIDIA RTX PRO 6000 GPUs (96 GB VRAM each).

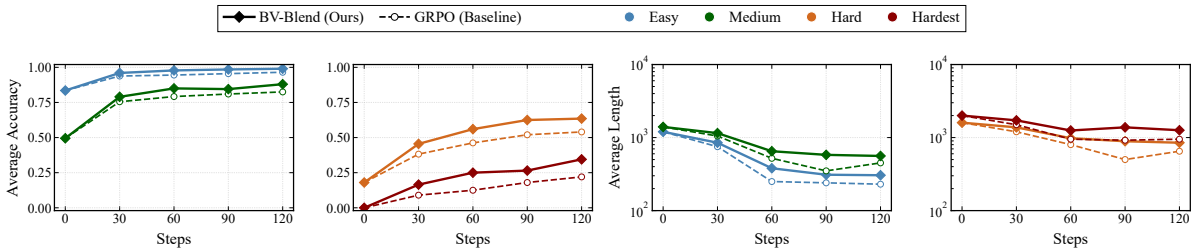
4.1 Main Results

Table 1 summarizes results on the Qwen2.5-Math-7B backbone. Overall, BV-Blend achieves the best average performance among the compared methods on both in-domain mathematical reasoning and OOD benchmarks. Notably, BV-Blend is a purely on-policy RLVR approach, yet it remains compet-



(a) Response length (b) Training entropy (c) Training reward (d) Effective signal ratio

Figure 2: **GRPO vs. BV-Blend.** We track (a) response length (tokens), (b) policy training entropy, (c) mean training reward (verifier score), and (d) the effective-signal ratio: the fraction of prompts whose method-specific normalization scale remains non-degenerate during training (GRPO: $\sigma_{\mathcal{G}}^{(m)}$; BV-Blend: $s^{(m)}$ in Eq. (10)).



(a) Accuracy: Standard (b) Accuracy: Complex (c) Length: Standard (d) Length: Complex

Figure 3: **Difficulty-stratified BV-Blend vs. GRPO.** We partition prompts into four difficulty buckets (Easy/Medium/Hard/Hardest) using a difficulty estimate computed *before* RL (see Appendix for details) and shared across methods, and track *verifier accuracy* (a,b; fraction of prompts with correct final answers) and *average response length* (c,d) across training checkpoints. The left pair (a,c) reports the *Standard* subset and the right pair (b,d) reports the *Complex* subset. Response length is measured on completion tokens and plotted on a log scale.

itive with hybrid/off-policy systems that additionally leverage external data.

In-Domain Mathematical Reasoning. We evaluate in-domain performance on five math benchmarks, where AIME is reported for two years (AIME 2024/2025), yielding six scores in total for the Math Reasoning Avg. AIME and AMC are reported as avg@32, while the remaining benchmarks use pass@1. Across these six scores, BV-Blend attains the highest average (51.7%), outperforming the strongest on-policy baseline Oat-Zero (43.8%) by 7.9 points. It also exceeds ReLIFT (47.8%) and LUFFY (50.1%), and is slightly above the extended-training LUFFY† variant (51.0%). At the per-benchmark level, BV-Blend achieves the best reported results on AIME 2024 (34.2%), AMC (66.5%), MATH-500 (87.9%), and Olympiad (57.4%). Meanwhile, it is marginally below the best baseline on Minerva (40.7% vs. 41.2%) and on AIME 2025 (23.6% vs. 25.5%), indicating broad but non-uniform gains across evaluation settings.

OOD Generalization. On the OOD suite (ARC-C, GPQA*, and MMLU-Pro), BV-Blend again achieves the best average score (64.1%), improving over the next-best baseline LUFFY† (61.8%) by 2.3 points. The largest gains appear on reasoning-centric benchmarks: BV-Blend achieves the best results on ARC-C (83.1%) and GPQA* (52.6%), exceeding LUFFY† by 3.6 points on GPQA*. On MMLU-Pro, BV-Blend is slightly below the best baseline (56.5% vs. 58.7%), suggesting that improvements are more pronounced on benchmarks emphasizing multi-step reasoning, while broad knowledge coverage may depend more on pretraining and instruction tuning.

Taken together, Table 1 shows that stabilizing the advantage signal can improve on-policy RLVR training, yielding substantial gains without requiring additional supervision beyond verifiable rewards, while remaining competitive with hybrid/off-policy approaches that leverage extra data.

4.2 Analysis of Training Dynamics

To diagnose how BV-Blend shapes optimization, Fig. 2 compares BV-Blend with the on-policy baseline GRPO across four training signals. BV-Blend consistently produces longer responses and reaches a higher, more stable length plateau (Fig. 2a), which is consistent with sustaining multi-step reasoning traces under the same rollout budget. While

Table 2: **Ablations of BV-Blend on Qwen2.5-Math-7B.** We ablate the SEM-based confidence weighting by constructing w_k from only the EMA effective count N_k^{eff} or only the historical scale $\sigma_{\text{hist}}(k)$. Avg. is computed over six scores (AIME 2024, AIME 2025, AMC, MATH-500, Minerva, Olympiad).

Model	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	Avg.
GRPO (baseline)	25.1/15.3	62.1	84.3	39.3	46.9	45.5
Naive historical averaging (w=0.5)	25.8/16.1	58.2	79.6	37.9	44.5	43.7
BV-Blend (w_k from N_k^{eff} only)	31.5/22.2	64.2	86.8	39.6	53.9	49.7
BV-Blend (w_k from σ_{hist} only)	29.1/21.3	65.2	87.3	40.3	51.2	49.1
BV-Blend (full SEM-based confidence)	**34.2/23.6**	**66.4**	**87.9**	**40.7**	**57.4**	**51.7**

entropy decreases for both methods early in training, BV-Blend maintains a higher residual policy entropy throughout (Fig. 2b), which is consistent with slower policy concentration and more persistent exploration. This is accompanied by slightly higher and noticeably smoother training rewards (Fig. 2c), with reduced volatility relative to GRPO.

Most importantly, the effective-signal ratio (Fig. 2d) highlights a difference in the resulting learning signal. We define the effective-signal ratio as the fraction of prompts whose normalization scale is non-degenerate under the corresponding estimator (GRPO: $\sigma_{\mathcal{G}}^{(m)}$; BV-Blend: $s^{(m)}$). With binary verifier rewards, prompt groups frequently become near-deterministic (all-correct or all-incorrect), causing $\sigma_{\mathcal{G}}^{(m)}$ to approach zero and thereby collapsing GRPO advantages toward zero, which reduces the number of informative prompt groups available for learning (Shao et al., 2024). In contrast, BV-Blend preserves a substantially higher effective-signal ratio by stabilizing the baseline and scale via cluster-conditioned historical moments when prompt-local dispersion is small, thereby maintaining usable learning signals and yielding more stable optimization dynamics overall.

4.3 Difficulty-stratified dynamics

To better understand where BV-Blend's gains arise and whether they are accompanied by undesirable verbosity, we analyze learning dynamics under a fixed difficulty stratification. Specifically, we assign each prompt to one of four buckets (Easy/Medium/Hard/Hardest) based on a baseline difficulty estimate computed before RL (see Appendix for details), and evaluate both BV-Blend and the on-policy baseline GRPO at the same checkpoints (Steps $0/30/60/90/120$). Fig. 3 reports bucket-wise average verifier accuracy and average completion length (log scale) on both *Standard* and *Complex* subsets. Overall, BV-Blend yields consistent accuracy improvements on harder buckets—most notably Hard and Hardest—while keeping response lengths comparable and avoiding systematic length blow-up. We observe mild non-monotonic fluctuations across checkpoints, which is expected in on-policy optimization due to sampling noise; importantly, BV-Blend shows a more consistent upward trend on difficult prompts, which is consistent with the hypothesis that historically stabilized advantage estimation improves training stability without evidence of systematic length inflation.

4.4 Ablation Study

Table 2 ablates the core design of BV-Blend on Qwen2.5-Math-7B. We compare GRPO, a naive fixed-weight historical mixture (w=0.5), two partial variants that compute the confidence weight using only one historical-uncertainty ingredient, and the full BV-Blend. All BV-Blend variants use the same baseline/scale blending in Eq. (10) and differ only in how w_k is computed from historical statistics. Concretely, the full method computes the SEM-style uncertainty $\text{SEM}_{\text{hist}}(k) = \sigma_{\text{hist}}(k)/\sqrt{N_k^{\text{eff}} + \delta_N}$ and maps it to confidence via Eq. (9). The N_k^{eff}-only variant drops the dependence on $\sigma_{\text{hist}}(k)$ (i.e., $\text{SEM}(k) \propto 1/\sqrt{N_k^{\text{eff}} + \delta_N}$), while the σ_{hist}-only variant drops the dependence on N_k^{eff} (i.e., $\text{SEM}(k) \propto \sigma_{\text{hist}}(k)$); in all cases, w_k is a monotone function of the corresponding uncertainty proxy.

Two observations emerge. First, simply injecting historical information is insufficient: naive averaging drops from 45.5 (GRPO) to 43.7, indicating that historical moments should be used selectively rather than uniformly. Second, either signal alone already improves over GRPO (49.7 and 49.1), and combining them through the SEM proxy performs best (51.7), improving by 6.2 points over GRPO and by 2.0 points over the best single-signal vari-

ant. The gains are particularly noticeable on harder benchmarks, such as Olympiad and AIME, which is consistent with the role of SEM-based confidence in stabilizing advantage estimation when prompt-local reward dispersion is small.

5 Conclusion

We identify an instability in critic-free RLVR: when prompt-local reward dispersion is small, group-normalized advantages can collapse and weaken the learning signal. We propose BV-Blend (BV-Blend), which stabilizes advantage estimation by blending prompt-local statistics with semantic-cluster-conditioned historical moments through an SEM-based confidence weight, while leaving the PPO-style clipped objective unchanged. Across experiments, BV-Blend improves training stability and performance on in-domain, OOD, and cross-backbone evaluations. Future work will explore more effective mechanisms for sharing historical information across prompts, especially under distribution shift.

Limitations

BV-Blend relies on a fixed prompt embedding and clustering pipeline; performance may depend on the encoder and codebook granularity, and may degrade under substantial distribution shift. It also requires maintaining cluster-conditioned EMA moments and aggregating per-cluster statistics in distributed training, which introduces additional bookkeeping and hyperparameter tuning. Finally, we primarily evaluate mathematical reasoning with verifiable outcome rewards and a limited set of backbones; broader domains and verifier/reward designs are needed to more fully assess generality. In addition, under extremely sparse reward regimes, if both the current prompt group and its relevant historical cluster provide little or no reward variation, BV-Blend may still offer limited learning signal.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No.2023YFF0905400), the National Natural Science Foundation of China (No.U2341229) and the Reform Commission Foundation of Jilin Province (No.2024C003).

References

Peter Chen, Xiaopeng Li, Ziniu Li, Xi Chen, and Tianyi Lin. 2025. Spectral policy optimization: Coloring your incorrect reasoning in grpo. *arXiv preprint arXiv:2505.11595*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Hugging Face. Chat templates. https://huggingface.co/docs/transformers/main/en/chat_templating. Transformers documentation (main). Accessed: 2026-01-05.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Jiayi Gui, Yiming Liu, Jiale Cheng, Xiaotao Gu, Xiao Liu, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Logicgame: Benchmarking rule-based reasoning abilities of large language models. *arXiv preprint arXiv:2408.15778*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv preprint arXiv:2510.10150*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand,*

August 11-16, 2024, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-Reasoner-Zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo M. Ponti, and Ivan Titov. 2025. Post-hoc reward calibration: A case study on length bias. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Hynek Kydlicek, Alina Lozovskaya, Nathan Habib, and Clémentine Fourrier. 2025. Fixing open llm leaderboard and introducing math-verify. https://huggingface.co/blog/math_verify_leaderboard. Hugging Face Blog. Accessed: 2026-01-05.

Thanh-Long V Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. 2025. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. *arXiv preprint arXiv:2509.21880*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. https://huggingface.co/datasets/Numinamath. Hugging Face repository, 13:9.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath.

Xu Liang. 2025. Group relative policy optimization for image captioning. *arXiv preprint arXiv:2503.01333*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025a. Understanding r1-zero-like training: A critical perspective. https://github.com/sail-sg/understand-r1-zero.

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

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Bin Cui, and 1 others. 2025. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. *arXiv preprint arXiv:2506.07527*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Youssef Mroueh. 2025. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/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, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*.

Meta Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Hu Wang, Congbo Ma, Ian Reid, and Mohammad Yaqub. 2025. Kalman filter enhanced grpo for reinforcement learning-based language model reasoning. *arXiv preprint arXiv:2505.07527*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

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*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024c. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, and 1 others. 2025b. Formalmath: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.02735*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Xinglang Zhang, Yunyao Zhang, ZeLiang Chen, Junqing Yu, Wei Yang, and Zikai Song. 2026a. Logical phase transitions: Understanding collapse in llm logical reasoning. *Preprint*, arXiv:2601.02902.

Yunyao Zhang, Yihao Ai, Zuocheng Ying, Qirui Mi, Junqing Yu, Wei Yang, and Zikai Song. 2026b. Coupling macro dynamics and micro states for long-horizon social simulation. *Preprint*, arXiv:2604.05516.

Yunyao Zhang, Zikai Song, Hang Zhou, Wenfeng Ren, Yi-Ping Phoebe Chen, Junqing Yu, and Wei Yang. 2025. $ga - s^3$: Comprehensive social network simulation with group agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8950–8970.

Yunyao Zhang, Xinglang Zhang, Junxi Sheng, Wenbing Li, Junqing Yu, Yi-Ping Phoebe Chen, Wei Yang, and Zikai Song. 2026c. Semantic-aware logical reasoning via a semiotic framework. *Preprint*, arXiv:2509.24765.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. 2025. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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

Appendix

Contents

A Experimental Setup

This appendix describes the experimental environment, task formulation, datasets, evaluation protocols, and base model configurations. Our goal is to provide sufficient detail for reproducibility.

A.1 Problem Formulation as a Markov Decision Process (MDP)

We model autoregressive generation as a finite-horizon episodic Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, H)$, where H is the maximum number of *generated completion tokens* (including EOS when present).

- **State Space ($\mathcal{S}$).** A state at step t is the prompt concatenated with the generated prefix: $s_t = [q; y_{<t}] = [q; y_1; \ldots; y_{t-1}]$, with initial state $s_1 = [q]$.
- **Action Space ($\mathcal{A}$).** The action space is the model vocabulary. At state s_t, the policy $\pi_\theta(\cdot \mid s_t)$ outputs a distribution over tokens, and we sample an action $a_t = y_t$.
- **Transition Dynamics (P).** Transitions are deterministic: after taking action $a_t = y_t$, the next state is $s_{t+1} = [q; y_{\le t}] = [s_t; y_t]$.
- **Reward (r) and Return.** We use sparse, outcome-based rewards. A scalar reward is given only at episode termination and is binary:

$$R(\tau) = \begin{cases} 1 & \text{if } \textit{Math-Verify}(\tau) = \text{True}, \\ 0 & \text{otherwise}, \end{cases}$$

  where $\tau = (y_1, \ldots, y_T)$ denotes the generated completion and *Math-Verify* returns `True` if the extracted final answer is judged equivalent to the gold answer under its comparison rules. No intermediate rewards are provided (i.e., $r_t = 0$ for $t < T$), and the terminal reward equals $r_T = R(\tau)$. In optimization (Sec. 3), this trajectory-level reward induces a single per-trajectory advantage that is applied to completion tokens, with gradients stopped through all reward statistics.
- **Termination and Horizon (H).** An episode terminates when either (i) the model emits an end-of-sequence token (EOS), or (ii) the number of generated completion tokens reaches the maximum length limit H. We set $H = 8192$ in all experiments.

A.2 Justification for the Mathematical Reasoning Domain

We select mathematical reasoning as our primary experimental domain for three reasons:

1. **Objective, verifiable rewards.** Mathematical problems admit a well-defined notion of correctness, enabling programmatic outcome verification against ground-truth final answers (e.g., by extracting the final answer, normalizing it, and checking equivalence). This reduces reliance on subjective annotations or learned judges that may introduce noise or exploitable biases.
2. **A natural stress test for prompt-local normalization under sparse binary rewards.** With sparse binary rewards, difficult prompts

often yield rollout groups where most samples share the same outcome (e.g., all incorrect), making within-prompt reward dispersion small. This setting makes it possible to directly study failure cases of prompt-local normalization methods, including GRPO-style estimators, where the normalization scale can become (near-)degenerate and standardized advantages weaken the effective learning signal (Sec. 3).

3. **Scalable and rigorous evaluation.** Automated verification supports efficient evaluation on large test sets at low cost, improving scalability, consistency, and reproducibility across methods and backbones.

A.3 Reward Computation and Visualized Example

Reward Computation. We compute sparse, outcome-based rewards using *Math-Verify*. Given a completed trajectory τ, *Math-Verify* (i) extracts a candidate final answer from the model output using a priority-based extraction pipeline (e.g., preferring content inside `\boxed{·}` when present), (ii) normalizes the extracted text and parses it into a canonical symbolic representation (e.g., via ANTLR-based parsing and `latex2sympy2_extended`), and (iii) checks equivalence against the gold answer using rule-based and SymPy-backed comparisons (Kydlicek et al., 2025). We assign a binary trajectory-level reward

$$R(\tau) = \begin{cases} 1 & \text{if } \textit{Math-Verify}(\tau) = \text{True}, \\ 0 & \text{otherwise}, \end{cases}$$

where parsing failures or non-equivalence return 0. Some answer representations receive specialized handling (e.g., relations, sets, and intervals), following *Math-Verify*'s documented comparison rules (Kydlicek et al., 2025). This outcome-based formulation is standard in settings with verifiable rewards (Hao et al., 2025). Related work has also examined reasoning structure, collapse phenomena, and long-horizon agent-based simulation from complementary perspectives (Zhang et al., 2026c,a, 2025, 2026b).

A.4 Base Models and Prompting Format

To support reproducibility, we list all base models and their sources.

- **Primary Model.** Our main experiments and ablations use `Qwen2.5-Math-7B` (Yang et al., 2024b), loaded from the Hugging Face Hub (`Qwen/Qwen2.5-Math-7B`).

Figure 4: An illustrative example showing a prompt, a successful and a failed generation trajectory, and their corresponding rewards under *Math-Verify*.

Prompt
Let $f(x) = x^2 - 2x + 3$. Find the vertex of the parabola.
Successful Generation Trajectory (τ_1)
The vertex of a parabola $f(x) = ax^2 + bx + c$ is at $x = -b/(2a)$. Here $a = 1, b = -2$. So $x = -(-2)/(2 \cdot 1) = 1$. The y-coordinate is $f(1) = 1^2 - 2(1) + 3 = 2$. The vertex is $(1, 2)$. `\boxed{(1, 2)}`
Reward for τ_1: $R(\tau_1) = 1$
Failed Generation Trajectory (τ_2)
The vertex of a parabola $f(x) = ax^2 + bx + c$ is at $x = b/(2a)$. Here $a = 1, b = -2$. So $x = -2/(2 \cdot 1) = -1$. The y-coordinate is $f(-1) = (-1)^2 - 2(-1) + 3 = 6$. The vertex is $(-1, 6)$. `\boxed{(-1, 6)}`
Reward for τ_2: $R(\tau_2) = 0$

- **Robustness-Test Models.** To test robustness across model scales and instruction tuning, we additionally evaluate `Qwen/Qwen2.5-Math-1.5B`, `Qwen/Qwen2.5-7B-Instruct` (Yang et al., 2024a), and `meta-llama/Meta-Llama-3.1-8B-Instruct` (Team, 2024). All checkpoints are obtained from the Hugging Face Hub.

- **Prompt Format.** We use each checkpoint's tokenizer-provided chat template to render prompts, via `tokenizer.apply_chat_template(. . . , add_generation_prompt=True)` for both training and evaluation (Face). A typical rendered prompt (shown here for clarity) has the following structure:

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{problem_description}<|im_end|>
<|im_start|>assistant
```

  where `{problem_description}` is the problem text from the dataset.

B Implementation Details and Pseudocode

This section describes the implementation of BV-Blend (BV-Blend), including the historical-moments buffer, update ordering, and simplified pseudocode for advantage computation and training.

B.1 Historical Moments Buffer

BV-Blend maintains *semantic-cluster-conditioned* historical reward moments, rather than per-prompt (UID-level) statistics. Let the clustering codebook have K clusters (Sec. 3.2.1). For each cluster $k \in \{1, \ldots, K\}$, we store three EMA statistics: (i) the EMA mean $m_1(k)$, (ii) the EMA second raw moment $m_2(k)$, and (iii) the EMA effective mass N_k^{eff}. In practice, we implement these as dense length-K buffers `m1[K]`, `m2[K]`, `n_eff[K]`, together with a boolean `seen[K]` flag, which simplifies distributed aggregation and implementation.

Cold-start initialization and update ordering. For each batch, BV-Blend computes the confidence weight w_k from the *pre-update* EMA state (i.e., before incorporating the current batch). If a cluster k has not been observed previously (`seen[k]=False`), we treat it as cold start: we set $w_k = 0$ for this batch (pure prompt-local normalization), and initialize its EMA state *after* the policy update using batch statistics:

$$N_k^{\text{eff}} \leftarrow N_0, \quad m_1(k) \leftarrow \mu_{\mathcal{B}}(k), \quad m_2(k) \leftarrow \mu_{\mathcal{B}}(k)^2 + v_{\text{prior}}.$$

For clusters with an existing EMA state, we update moments *post-update* using aggregated sufficient statistics from the current batch (Eq. (6)).

Distributed aggregation. Given batch $\mathcal{B}$, we compute per-cluster sufficient statistics $S_{1,\mathcal{B}}(k) = \sum_{(m,i)\in\mathcal{I}_{\mathcal{B}}(k)} R_i^{(m)}$, $S_{2,\mathcal{B}}(k) = \sum_{(m,i)\in\mathcal{I}_{\mathcal{B}}(k)} (R_i^{(m)})^2$, and $N_{\mathcal{B}}(k) = |\mathcal{I}_{\mathcal{B}}(k)|$. In distributed training, we all-reduce (sum) these dense buffers across workers before applying EMA updates.

B.2 BV-Blend Training Loop and Advantage Computation

Algorithm 2 shows the overall loop. Algorithm 1 details BV-Blend advantage computation, which blends *baseline and variance statistics* and then standardizes once (Eq. (10)–(11)). As in the main text, we stop gradients through all reward statistics, including $\mu_{\mathcal{G}}^{(m)}$, $\sigma_{\mathcal{G}}^{(m)}$, and EMA moments.

B.3 Model and Optimization

Model Architecture. We perform full-parameter fine-tuning on all base models. We do not use parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2022), and we do not add adapters or custom heads; the transformer architecture remains unchanged.

Optimization Objective. During each policy update, we optimize a PPO-style clipped surrogate objective (Schulman et al., 2017) using the BV-Blend advantages $A_i^{\text{BV},(m)}$ (Sec. 3). Concretely, we maximize

$$\begin{aligned}\mathcal{J}(\theta) = \mathbb{E}_m\Bigg[\frac{1}{G}\sum_{i=1}^{G}\mathbb{E}_{t\sim\tau_i^{(m)}}\Big[\min\Big(r_{i,t}^{(m)}(\theta)\,A_i^{\text{BV},(m)},\\ \tilde{r}_{i,t}^{(m)}(\theta)\,A_i^{\text{BV},(m)}\Big)\Big]\Bigg]\\ -\beta\,\mathbb{E}_{m,i}\Big[\mathbb{E}_{t\sim\tau_i^{(m)}}\big[\mathrm{KL}\Big(\pi_\theta(\cdot\mid s_t)\,\Big\|\\ \pi_{\text{ref}}(\cdot\mid s_t)\Big)\big]\Big]\\ +\lambda_{\text{ent}}\,\mathbb{E}_{m,i}\Big[\mathbb{E}_{t\sim\tau_i^{(m)}}\big[\mathcal{H}(\pi_\theta(\cdot\mid s_t))\big]\Big],\end{aligned} \tag{12}$$

where $r_{i,t}^{(m)}(\theta)$ and $\tilde{r}_{i,t}^{(m)}(\theta)$ are defined in Eq. (1), and $\mathbb{E}_{t\sim\tau_i^{(m)}}[\cdot]$ denotes the completion-token mean with the same masking convention as Eq. (2). We implement training by minimizing the loss $\mathcal{L}(\theta) = -\mathcal{J}(\theta)$. Unless otherwise specified, we set $\lambda_{\text{ent}} = 0.01$ and use a fixed reference policy π_{ref}; in our main comparisons, we set $\beta = 0$.

B.4 Prompting Strategy

We format inputs using each model's tokenizer-provided chat template via `tokenizer.apply_chat_template(. . . , add_generation_prompt=True)` for both training and evaluation, which improves reproducibility and avoids formatting mismatches across backbones (Face). Unless otherwise stated, we use a generic assistant system message and provide the problem statement as the user message. We do not require additional task-specific instructions for Qwen-based math models; they typically produce multi-step solutions by default. For Llama-3.1-Instruct in the robustness experiments, we optionally prepend a short reasoning cue (e.g., "Let's think step by step.") to encourage step-by-step solutions (Wei et al., 2022).

Rendered prompt for Qwen-based chat models (schematic)

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{problem_description}<|im_end|>
<|im_start|>assistant
```

Rendered prompt for Llama-3.1-Instruct (schematic)

User: {problem_description}
Assistant: Let's think step by step.

Notation. For convenience, Table 3 summarizes the key symbols used throughout the paper, consistent with Sec. 3. We group notation into (i) trajectories and PPO-style optimization, (ii) prompt-local (GRPO-style) reward statistics, and (iii) BV-Blend's semantic clustering, historical EMA moments, and SEM-based confidence weighting. Unless otherwise stated, prompts are indexed by m, trajectories by i, completion-token positions by t, clusters by k, and codebook entries by j. All reward/statistics terms used to form advantages (e.g., $\mu_{\mathcal{G}}^{(m)}$, $\sigma_{\mathcal{G}}^{(m)}$, and statistics derived from historical EMA moments) are treated as stop-gradient constants.

C Extended Experimental Results

This section presents supplementary quantitative results and ablations that complement the main paper. Throughout, we compare against the on-policy GRPO baseline under the same PPO-style clipped objective and vary only the advantage estimator.

C.1 Further Ablation Studies

Confidence-to-weight mapping. We ablate the functional form that maps historical uncertainty to the confidence weight w_k. All variants compute the same SEM-style uncertainty $\mathrm{SEM}_{\mathrm{hist}}(k)$ from *pre-update* EMA statistics (Eq. (8)) and use the same baseline/variance blending (Eq. (10)); they differ only in the mapping $w_k = g(u_k)$, where $u_k \triangleq \mathrm{SEM}_{\mathrm{hist}}(k)/T$. Table 4 reports results on AIME 2024 (avg@32). Among the tested monotone mappings, the exponential form $w_k = \exp(-u_k)$ performs best, which is consistent with the use of a smooth confidence schedule rather than a hard threshold.

C.2 Robustness Across Diverse Models

To assess robustness beyond our primary Qwen2.5-Math-7B setting, we evaluate BV-Blend on three additional backbones: Qwen2.5-Math-1.5B, Qwen2.5-7B-Instruct, and LLaMA-3.1-8B. As shown in Fig. 5, BV-Blend improves over the best baseline included in our comparison on all three settings: +2.1 points on Qwen2.5-Math-1.5B and +1.6 points on Qwen2.5-7B-Instruct. The most challenging case is LLaMA-3.1-8B, where standard prompt-local estimators exhibit pronounced instability under our RLVR setup and can substantially degrade final performance; in this regime, BV-Blend reaches 19.9, improving by 2.8 points over the best-performing baseline. While absolute performance on LLaMA-3.1-8B remains modest under this setup, the relative trend is still informative: BV-Blend reduces degradation and remains more stable than prompt-local normalization when within-group reward dispersion is small. Overall, these results suggest that BV-Blend is a practical replacement for prompt-local advantage normalization in on-policy RLVR across heterogeneous backbones and training conditions.

C.3 Proof that GRPO Advantages Vanish Under Uniform Prompt-Group Rewards

We show that the GRPO-style prompt-local standardized advantage is exactly zero when all rollouts within a prompt group receive the same reward.

Theorem 1 (Prompt-local collapse in GRPO)**.** *Fix a prompt $q^{(m)}$ with rollout set $\mathcal{G}(q^{(m)}) = \{\tau_i^{(m)}\}_{i=1}^{G}$. If $R_i^{(m)} = c$ for all $i \in \{1, \dots, G\}$ and some constant c, then $A_i^{\mathrm{GRPO},(m)} = 0$ for all i.*

Proof. By Eq. (3),

$$A_i^{\mathrm{GRPO},(m)} = \frac{R_i^{(m)} - \mu_{\mathcal{G}}^{(m)}}{\sigma_{\mathcal{G}}^{(m)} + \delta},$$

where $\mu_{\mathcal{G}}^{(m)}$ and $\sigma_{\mathcal{G}}^{(m)}$ are the mean and standard deviation of $\{R_i^{(m)}\}_{i=1}^{G}$, and $\delta > 0$. If $R_i^{(m)} = c$ for all i, then $\mu_{\mathcal{G}}^{(m)} = c$ and $\sigma_{\mathcal{G}}^{(m)} = 0$. Substituting yields

$$A_i^{\mathrm{GRPO},(m)} = \frac{c - c}{0 + \delta} = 0,$$

for all i. □

D Bias–Variance Analysis of the BV-Blend Advantage Estimator

This section analyzes the statistical behavior of BV-Blend's advantage construction. Rather than mixing two advantage estimators directly, BV-Blend blends a *baseline* and *variance statistics* (which determine the scale) and then performs a single standardization:

$$A_i^{\mathrm{BV},(m)} = \frac{R_i^{(m)} - b^{(m)}}{s^{(m)} + \delta}, \qquad (13)$$

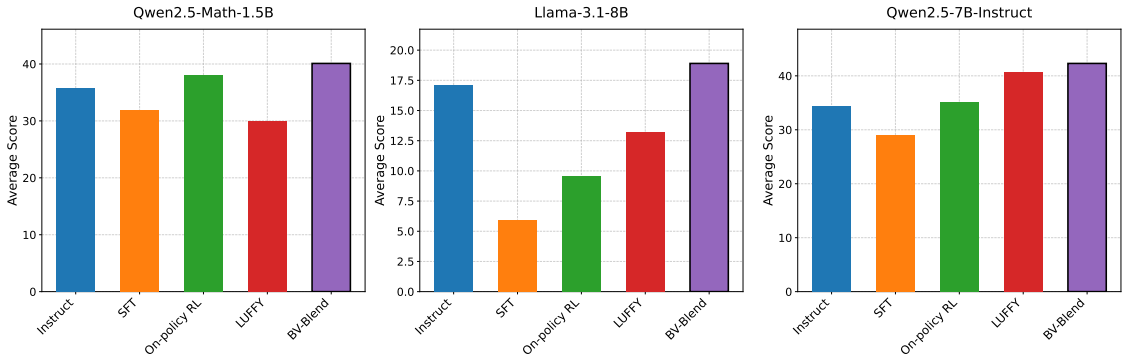


Figure 5: **Cross-backbone robustness.** Performance of BV-Blend relative to baselines across diverse model backbones under the same RLVR setup.

where $(b^{(m)}, s^{(m)})$ are defined by Eq. (10). Throughout, we treat the confidence weight w_k as fixed within the current batch because it is computed from *pre-update* EMA moments (Sec. 3.2.3) and gradients are stopped through all reward statistics.

Conditional mean and variance (given $b^{(m)}, s^{(m)}$). Conditioned on the (stop-gradient) statistics $(b^{(m)}, s^{(m)})$, BV-Blend is an affine transform of the trajectory reward:

$$\mathbb{E}\Big[A_i^{\mathrm{BV},(m)} \mid b^{(m)}, s^{(m)}\Big] = \frac{\mathbb{E}[R_i^{(m)}] - b^{(m)}}{s^{(m)} + \delta}, \tag{14}$$

$$\mathrm{Var}\Big(A_i^{\mathrm{BV},(m)} \mid b^{(m)}, s^{(m)}\Big) = \frac{\mathrm{Var}(R_i^{(m)})}{(s^{(m)} + \delta)^2}. \tag{15}$$

Thus, $s^{(m)}$ directly controls the magnitude of the learning signal entering the PPO-style surrogate objective.

Why BV-Blend can mitigate prompt-local collapse. Under GRPO-style prompt-local normalization, $s^{(m)} = \sigma_{\mathcal{G}}^{(m)}$. If rewards in a prompt group are constant (all-correct or all-incorrect), then $\sigma_{\mathcal{G}}^{(m)} = 0$ and the standardized advantages are exactly zero (Theorem 1), eliminating the learning signal for that group.

In BV-Blend, the blended scale satisfies

$$(s^{(m)})^2 = w^{(m)} v_{\mathrm{hist}}(k^{(m)}) + (1 - w^{(m)})(\sigma_{\mathcal{G}}^{(m)})^2.$$

Therefore, when $\sigma_{\mathcal{G}}^{(m)} \approx 0$, the scale remains non-degenerate whenever $w^{(m)} > 0$ and $v_{\mathrm{hist}}(k^{(m)}) > 0$. If a prompt group has constant reward $R_i^{(m)} = c$ for all i, then

$$b^{(m)} = (1 - w^{(m)})c + w^{(m)} \mu_{\mathrm{hist}}(k^{(m)}),$$
$$s^{(m)} = \sqrt{w^{(m)} v_{\mathrm{hist}}(k^{(m)})},$$

which yields

$$A_i^{\mathrm{BV},(m)} = \frac{w^{(m)}\big(c - \mu_{\mathrm{hist}}(k^{(m)})\big)}{\sqrt{w^{(m)} v_{\mathrm{hist}}(k^{(m)})} + \delta}. \tag{16}$$

Thus, BV-Blend can retain a non-zero advantage even when prompt-local dispersion vanishes, provided $w^{(m)} > 0$ and $v_{\mathrm{hist}}(k^{(m)}) > 0$; conversely, if $w^{(m)} = 0$ (e.g., an unseen cluster) or $v_{\mathrm{hist}}(k^{(m)}) = 0$, the advantage may still collapse for that batch.

A bias–variance viewpoint via moment shrinkage. BV-Blend can be viewed as constructing *shrunk* estimators of the reward mean and variance (within a semantic cluster), and then standardizing once. Let $\mu_\star(k)$ and $v_\star(k)$ denote the (hypothetical) population moments for cluster k under the current policy, and let $\mu_{\mathcal{G}}^{(m)}$ and $(\sigma_{\mathcal{G}}^{(m)})^2$ be the prompt-local sample moments computed from G rollouts. BV-Blend uses

$$b^{(m)} = w^{(m)} \mu_{\mathrm{hist}}(k^{(m)}) + (1 - w^{(m)}) \mu_{\mathcal{G}}^{(m)}, \tag{17}$$

$$(s^{(m)})^2 = w^{(m)} v_{\mathrm{hist}}(k^{(m)}) + (1 - w^{(m)})(\sigma_{\mathcal{G}}^{(m)})^2, \tag{18}$$

i.e., convex combinations of a low-variance historical estimate (aggregating many past samples) and a high-variance prompt-local estimate (based on G rollouts). For intuition, under the approximation that (i) $w^{(m)}$ is fixed within the batch and (ii) the prompt-local moments are approximately unbiased for $(\mu_\star, v_\star)$ up to standard finite-sample effects, the

bias of $b^{(m)}$ inherits the historical bias scaled by $w^{(m)}$:

$$\mathrm{Bias}\Big(b^{(m)}\Big) \approx w^{(m)}\,\mathrm{Bias}\Big(\mu_{\mathrm{hist}}(k^{(m)})\Big). \quad (19)$$

Similarly, the variance of $b^{(m)}$ is reduced relative to the prompt-local estimator when the historical estimate is substantially more certain:

$$\mathrm{Var}\Big(b^{(m)}\Big) \approx (1-w^{(m)})^2\mathrm{Var}\Big(\mu_{\mathcal{G}}^{(m)}\Big) + (w^{(m)})^2\mathrm{Var}\Big(\mu_{\mathrm{hist}}(k^{(m)})\Big), \quad (20)$$

where we typically expect the cross-covariance to be small because μ_{hist} is computed from past batches. These relations are intended as explanatory approximations; they ignore estimation error in $w^{(m)}$ and $s^{(m)}$, as well as the additional nonlinearity introduced by the square root and the final normalization.

Why SEM-based weighting is a useful proxy. BV-Blend sets w_k as a monotone function of the SEM-style uncertainty proxy $\mathrm{SEM}_{\mathrm{hist}}(k) = \sigma_{\mathrm{hist}}(k)/\sqrt{N_k^{\mathrm{eff}} + \delta_N}$ (Eq. (8)), matching the classical scaling of the standard error of the sample mean with the standard deviation and sample size. This makes w_k large when historical estimates are both low-variance and supported by a large effective count, and small otherwise, thereby adapting the shrinkage strength to estimated uncertainty.

D.1 Discussion of Key Assumptions

The analysis above is intended as an explanatory approximation and relies on standard assumptions:

- **Stop-gradient statistics.** We treat $(b^{(m)}, s^{(m)}, w_k)$ as fixed w.r.t. policy gradients, consistent with our implementation (Sec. 3).
- **Approximate independence across batches.** Historical EMA moments are computed from past batches, so their covariance with current prompt-local moments is often small, though not exactly zero.
- **Limited non-stationarity.** Historical moments are most informative when the reward distribution within a semantic cluster does not shift arbitrarily fast; the SEM-based weight is designed to reduce reliance on history when uncertainty is high.
- **No strict unbiasedness claim.** Because BV-Blend uses normalization and heuristic shrinkage (and PPO further clips the objective), we do not claim that the resulting standardized quantity is a strictly unbiased estimator of the true MDP advantage.

E Reproducibility Details

All experiments were conducted on a single multi-GPU node. We report the main hardware and software stack to facilitate reproduction.

Hardware. Experiments were conducted on one node equipped with 8 NVIDIA RTX PRO 6000 GPUs (96 GB memory per GPU). Training uses PyTorch Fully Sharded Data Parallel (FSDP) across all 8 GPUs.

Software. We build on PyTorch 2.4.0 and the Hugging Face Transformers/Accelerate stack (with exact package versions provided in the accompanying environment specification). For generation, we use vLLM (v0.6.3) with CUDA 12.1. We additionally use FlashAttention (v2.7.3), TensorDict (v0.5.0), and the `verl` library for RL orchestration. All experiments run in a containerized CUDA 12.1 environment.

Algorithm 1 Compute BV-Blend Advantages A^{BV} (Simplified)

Require: Prompts $\{q^{(m)}\}$, rewards $\{R_i^{(m)}\}$, EMA buffers $(m_1, m_2, N^{\mathrm{eff}})$ and flags `seen`, encoder $E(\cdot)$, codebook $\{c_j\}$.

Ensure: Advantages $\{A_i^{\mathrm{BV},(m)}\}$.

for each prompt m **do**
  Compute prompt-local mean/std $\mu_{\mathcal{G}}^{(m)}, \sigma_{\mathcal{G}}^{(m)}$ from $\{R_i^{(m)}\}_{i=1}^{G}$.
  Assign cluster $k^{(m)} \leftarrow \arg\min_j \|E(q^{(m)}) - c_j\|_2^2$.
  if `seen[`$k^{(m)}$`]=False` (unseen cluster, pre-update) **then**
    Set $w^{(m)} \leftarrow 0$. (cold start; pure prompt-local)
    Set $b^{(m)} \leftarrow \mu_{\mathcal{G}}^{(m)}$, $s^{(m)} \leftarrow \sigma_{\mathcal{G}}^{(m)}$.
    Mark $k^{(m)}$ as "needs init" for post-update initialization.
  else
    $\mu_{\mathrm{hist}} \leftarrow m_1(k^{(m)})$; $v_{\mathrm{hist}} \leftarrow \max(m_2(k^{(m)}) - m_1(k^{(m)})^2, 0)$.
    $\mathrm{SEM}_{\mathrm{hist}} \leftarrow \sqrt{v_{\mathrm{hist}}} / \sqrt{N_{k^{(m)}}^{\mathrm{eff}} + \delta_N}$.
    $w^{(m)} \leftarrow \exp(-\mathrm{SEM}_{\mathrm{hist}}/T)$.
    Set

$$b^{(m)} \leftarrow w^{(m)}\mu_{\mathrm{hist}} + (1 - w^{(m)})\mu_{\mathcal{G}}^{(m)},$$
$$s^{(m)} \leftarrow \sqrt{w^{(m)}v_{\mathrm{hist}} + (1 - w^{(m)})(\sigma_{\mathcal{G}}^{(m)})^2}.$$

  end if
  for each trajectory $i = 1, \ldots, G$ **do**
    $A_i^{\mathrm{BV},(m)} \leftarrow (R_i^{(m)} - b^{(m)})/(s^{(m)} + \delta)$.
  end for
end for
return $\{A_i^{\mathrm{BV},(m)}\}$.

Algorithm 2 BV-Blend (BV-Blend) Training Loop (Simplified)

Initialize policy parameters θ, reference policy π_{ref}, and fixed clustering codebook $\{c_j\}_{j=1}^{K}$.
Initialize EMA buffers `m1[K]`, `m2[K]`, `n_eff[K]` and flags `seen[K]` $\leftarrow$ `False`.
for iteration $t = 1, 2, \ldots$ **do**
  Set $\theta_{\mathrm{old}} \leftarrow \theta$. (behavior snapshot)
  Sample M prompts $\{q^{(m)}\}_{m=1}^{M}$; for each prompt, sample G trajectories $\{\tau_i^{(m)}\}_{i=1}^{G}$ from $\pi_{\theta_{\mathrm{old}}}$.
  Compute verifier rewards $R_i^{(m)}$ for all trajectories.
  Compute BV-Blend advantages $A_i^{\mathrm{BV},(m)}$ using Algorithm 1 (pre-update EMA).
  Update θ by optimizing the PPO-style clipped objective using $A_i^{\mathrm{BV},(m)}$ (and optional KL/entropy terms).
  Post-update: aggregate per-cluster sufficient statistics $\{S_{1,\mathcal{B}}(k), S_{2,\mathcal{B}}(k), N_{\mathcal{B}}(k)\}_{k=1}^{K}$ across workers.
  Post-update: update EMA buffers for clusters with $N_{\mathcal{B}}(k) > 0$ using Eq. (6); for previously unseen clusters, initialize using $(N_0, V_{\mathrm{prior}})$ and set `seen[k]` $\leftarrow$ `True`.
end for

Table 3: Summary of key symbols (aligned with Sec. 3).

Symbol	Definition
	Trajectories and PPO-style optimization
π_θ, $\pi_{\theta_{\text{old}}}$	Current policy and behavior (rollout) policy snapshot.
π_{ref}	Fixed reference policy used for optional KL regularization.
M	Number of prompts in a training batch.
$q^{(m)}$	The m-th prompt in a training batch.
G	Number of rollouts (trajectories) per prompt.
$\mathcal{G}(q^{(m)})$	Rollout group for prompt $q^{(m)}$: $\{\tau_i^{(m)}\}_{i=1}^{G}$.
$\tau_i^{(m)}$	The i-th trajectory sampled for prompt $q^{(m)}$.
$T_i^{(m)}$	Number of generated tokens in trajectory $\tau_i^{(m)}$.
$R_i^{(m)}$	Trajectory-level verifier reward for $\tau_i^{(m)}$.
$r_{i,t}^{(m)}(\theta)$, $\tilde{r}_{i,t}^{(m)}(\theta)$	Importance ratio and its clipped version (Eq. (1)).
ϵ	PPO clipping parameter.
$m_{i,t}^{(m)}$	Completion-token mask used in token means (Eq. (2)).
$\mathbb{E}_{t\sim\tau}[\,\cdot\,]$	Generic notation for the masked mean over completion tokens (Eq. (2)).
	Prompt-local (GRPO-style) statistics
$\mu_{\mathcal{G}}^{(m)}$, $\sigma_{\mathcal{G}}^{(m)}$	Prompt-local mean/std of $\{R_i^{(m)}\}_{i=1}^{G}$.
$A_i^{\text{GRPO},(m)}$	GRPO-style standardized advantage (Eq. (3)).
δ	Small constant for numerical stability in advantage computation.
	BV-Blend: clustering and historical EMA moments
$E(\cdot)$	Frozen prompt encoder.
K, $\{c_j\}_{j=1}^{K}$	Number of clusters and fixed K-means codebook.
$k^{(m)}$	Cluster assignment for prompt $q^{(m)}$ (Eq. (5)).
$m_1(k)$, $m_2(k)$	EMA mean and EMA second raw moment for cluster k.
N_k^{eff}	EMA effective mass for cluster k.
$\mu_{\text{hist}}(k)$, $v_{\text{hist}}(k)$, $\sigma_{\text{hist}}(k)$	Historical mean/variance/std derived from EMA moments (Eq. (7)).
γ	EMA update rate (Eq. (6)).
$\mathcal{B}$	Current rollout batch.
$S_{1,\mathcal{B}}(k)$, $S_{2,\mathcal{B}}(k)$, $N_{\mathcal{B}}(k)$	Per-cluster sufficient statistics in batch $\mathcal{B}$.
N_0, V_{prior}	Cold-start initialization hyperparameters for new clusters.
	BV-Blend: confidence and advantage
$\text{SEM}_{\text{hist}}(k)$	SEM-style uncertainty proxy (Eq. (8)).
δ_N	Small constant in SEM computation (Eq. (8)).
T	Temperature controlling sensitivity in w_k (Eq. (9)).
w_k	Historical confidence weight computed from pre-update EMA statistics.
$w^{(m)}$	Confidence weight for prompt m: $w^{(m)} = w_{k^{(m)}}$.
$b^{(m)}$, $s^{(m)}$	Blended baseline and scale for prompt m (Eq. (10)).
$A_i^{\text{BV},(m)}$	BV-Blend advantage (Eq. (11)).

Table 4: Ablation on the confidence mapping $w_k = g(u_k)$ on AIME 2024 (avg@32), where $u_k = \mathrm{SEM}_{\mathrm{hist}}(k)/T$.

Confidence Mapping $g(u)$	**AIME 2024 (%)**
$1/(1+u)$	32.5
Linear decay: $\max(0, 1-u)$	30.1
Exponential decay: $\exp(-u)$ **(ours)**	**34.2**