WHEN SHARPENING BECOMES COLLAPSE: SAMPLING BIAS AND SEMANTIC COUPLING IN RL WITH VERIFIABLE REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) is a central paradigm for turning large language models (LLMs) into reliable problem solvers, especially in logic-heavy domains. Despite its empirical success, it remains unclear whether RLVR elicits novel capabilities or merely sharpens the distribution over existing knowledge. We study this by formalizing over-sharpening, a collapse of the policy onto a limited set of modes that suppresses valid alternatives. Theoretically, we show that finite-batch updates intrinsically bias learning toward sampled modes, triggering a collapse that propagates globally via semantic coupling. To mitigate this, we propose inverse-success advantage calibration to prioritize difficult queries and distribution-level calibration to diversify sampling via a memory network. Extensive experiments show that our methods preserve a healthier exploration-exploitation balance and yield sustained accuracy gains over state-of-the-art RLVR baselines.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a cornerstone approach for specializing large language models (LLMs) to domains where correctness is checkable (DeepSeek-AI, 2025a; Yang et al., 2025). Yet there is still no consensus on what RLVR fundamentally contributes. One line of work (DeepSeek-AI, 2025b; Team, 2025) argues that RLVR unlocks new capabilities, citing emergent chain-of-thought reasoning and strong performance on challenging benchmarks. In contrast, recent studies (Karan & Du, 2025; Yue et al., 2025; Song et al., 2025) suggest a more conservative view: with sufficiently large sampling budgets, pre-trained models can often match or even outperform their RL-fine-tuned counterparts. This indicates that RLVR may primarily sharpen the output distribution, making models more confident in knowledge and reasoning patterns already present before RL training.

We argue that these views are not mutually exclusive. From a distributional perspective, pre-training and supervised fine-tuning (SFT) provide the model with a rich, low-dimensional manifold of knowledge and latent reasoning procedures (Aghajanyan et al., 2021). RLVR then performs a search over this manifold, guided by verifiable rewards. We formalize the core mechanism of this process as sharpening and distinguish between two critical regimes: ***moderate and over-sharpening***. Moderate sharpening is beneficial, since it shifts probability mass toward high-reward semantic modes, making the model more reliable under finite sampling. Over-sharpening, however, collapses the policy onto a narrow set of modes, prematurely discarding valid alternative strategies. This restricts exploration, hinders the discovery of new reasoning chains, and eventually limits capability growth.

Our work is closely related to studies of entropy collapse in RLVR, where the predictive distribution becomes prematurely concentrated. In many respects, over-sharpening serves as a more rigorous and formal characterization of this collapse. Existing mitigation strategies include token-level heuristics, such as selectively updating high-entropy tokens (Wang et al., 2025a) or truncating gradients for highly confident ones (Cui et al., 2025); batch-level interventions, such as filtering degenerate samples (Xiong et al., 2025) or using curated synthetic data (Amin et al., 2025); and explicit entropy regularization, including DAPO-style clipping (Yu et al., 2025; Wu et al., 2025) and direct entropy penalties (Cheng et al., 2025; Prabhudesai et al., 2025). Although effective in practice,

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

these methods largely treat entropy as a finite resource to be preserved, leaving the mechanism behind over-sharpening underexplored. In particular, it remains unclear whether over-sharpening is an inevitable consequence of improved performance or a structural artifact of algorithmic bias.

Contribution I: a formal framework of sharpening in RLVR. We identify two mechanisms that drive over-sharpening in RLVR, namely sampling bias and semantic coupling. First, practical training can sample only a small number of responses per query. These samples are typically drawn from high-probability modes and therefore cover only a subset of valid solutions. Updates are consequently biased toward reinforcing these sampled modes, gradually collapsing the policy around them. Second, because parameters are shared across inputs, an update induced by one query can affect semantically related queries. This semantic coupling allows local collapse to propagate globally. Over time, the effective solution space for many inputs shrinks, the model gains little new information from additional RLVR steps, and collapse is further reinforced. Section 3 analyzes these mechanisms both theoretically and empirically.

Contribution II: a practical calibration with an exploratory sampling extension. Our primary method-level contribution is IAC, a one-line, compute-free calibration of trajectory contributions that prevents easy, already-solved queries from dominating optimization. As an exploratory extension, we further propose DLC, which reshapes the rollout distribution via a memory network to avoid repeatedly sampling the same high-probability patterns; it is effective but incurs non-trivial rollout overhead. Together, these techniques help keep RLVR in the moderate-sharpening regime: the model becomes more decisive where beneficial while preserving diverse and potentially useful behaviors.

Contribution III: extensive experiments over multiple LLMs and benchmark datasets. The performance of the proposed methods over six mathematical reasoning benchmarks consistently outperforms state-of-the-art baselines across both AVG@8 and PASS@8 by approximately 2%$\sim$3%. We also show that our methods are compatible with existing entropy-regularization techniques (e.g., DAPO), delivering additive performance improvements.

2 Related Work

Since OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (DeepSeek-AI, 2025a), RLVR, including GRPO and its variants, has become a prominent approach for improving LLM reasoning. Its advantage over SFT may partly stem from explicitly using negative samples (Xiong et al., 2025). To stabilize training, much work has refined advantage estimation, including RLOO (Kool et al., 2019), DrGRPO (Liu et al., 2025), and Reinforce++ with baseline (Hu et al., 2025). Another line of work studies entropy collapse: Cui et al. (2025) showed that performance gains often trade off against entropy, causing predictable saturation. Accordingly, methods such as high-entropy token selection (Wang et al., 2025a), multi-temperature sampling (Zhuang et al., 2025), entropy-aware clipping (Yu et al., 2025; Wu et al., 2025), and explicit entropy regularization (Prabhudesai et al., 2025; He et al., 2025) have been proposed. Nevertheless, a parallel empirical line questions what RLVR fundamentally adds: with sufficiently large sampling budgets, base models can match or even outperform their RL-fine-tuned counterparts (Karan & Du, 2025; Yue et al., 2025; Song et al., 2025), suggesting that RLVR primarily reshapes the output distribution rather than creating new capabilities.

Our work reconciles these two views at the mechanism level: the failures of RLVR correspond to over-sharpening, a structural artifact of finite-batch updates and parameter sharing, and the proposed calibrations keep the policy in the beneficial moderate-sharpening regime.

3 Analyzing Distribution Collapse

We first show that infinite-sample RLVR yields moderate sharpening, then demonstrate how finite-batch sampling fundamentally alters this behavior and propagates over-sharpening to other queries via semantic coupling.

3.1 Problem Setup

We consider a policy model π_θ initialized from a reference model π_{ref}. For a query q, we coarsen the sequence-level output space into K_1 mutually exclusive correct modes $\mathcal{O}^+ = \{o_1, \ldots, o_{K_1}\}$ and K_2

incorrect modes $\mathcal{O}^-$. Each mode can be viewed as an equivalence class of outputs that implement the same high-level reasoning pattern or solution. In the extreme, one may treat each individual output sequence as a separate mode and thus our analysis remains general. We focus on the RLVR setting with binary verifiable rewards, where correct modes share a non-negative advantage $A_+ \geq 0$, and incorrect modes share $A_- < 0$. Definition 3.1 formalizes two possible behaviors of an updated policy relative to π_{ref}.

Definition 3.1. A policy π_{new} exhibits moderate sharpening if the probability of every correct mode increases (i.e., $\pi_{\text{new}}(o|q) \geq \pi_{\text{ref}}(o|q)$ for all $o \in \mathcal{O}^+$), and over-sharpening if the probability of any correct mode decreases.

Moderate sharpening reflects a desirable confidence increase without forgetting valid reasoning patterns. Over-sharpening instead suppresses some correct modes in favor of others, which could be problematic because the suppressed patterns might be essential for solving other queries. As this suppression intensifies, the policy approaches a deterministic distribution. Thus, entropy collapse can be viewed as the limiting case of severe over-sharpening. Conceptually, the two notions operate at different levels: entropy collapse is a macroscopic empirical observation about the output distribution, whereas over-sharpening is a microscopic mechanism that formalizes when and why it occurs. On easy queries whose correct modes are already dominant, over-sharpening suppresses the remaining valid modes and forfeits the benefits that sharpening would otherwise bring; through semantic coupling, this suppression propagates to related queries and can permanently halt the unlocking of capabilities on hard ones. The two accounts are therefore complementary: existing entropy-based methods treat the symptom, while our analysis traces the mechanism.

3.2 Sampling Bias

Infinite-sample sharpening behavior. We begin with the standard KL-regularized RL objective:

$$\max_{\theta} \ \mathbb{E}_{q\sim\mathcal{D}}[\mathbb{E}_{o\sim\pi_\theta(\cdot|q)}[A(q,o)] - \beta D_{KL}(\pi_\theta(\cdot|q)\|\pi_{ref}(\cdot|q)), \tag{1}$$

where $\mathcal{D}$ is the query distribution, $A(q,o)$ is the advantage, and $\beta \geq 0$ controls regularization. The optimal solution has the following closed form.

Theorem 3.1. *The maximizer of Equation* (1)*, denoted as π^*, is given by*

$$\pi^*(o|q) = \frac{1}{Z(q)}\,\pi_{ref}(o|q)\,\exp\left(\frac{A(q,o)}{\beta}\right), \tag{2}$$

where $Z(q)$ is the partition function ensuring normalization. See Appendix A for proof.

Then, as can be seen in Corollary 3.1, Equation (1) guarantees moderate sharpening.

Corollary 3.1. *Suppose $A(q,o_i) = A_+ \geq 0$ for $o_i \in \mathcal{O}^+$ and $A(q,o_j) = A_- \leq 0$ for $o_j \in \mathcal{O}^-$, then Equation* (2) *yields moderate sharpening.*

Proof. From Equation (2), for $o_i \in \mathcal{O}^+$, $\pi^*(o_i|q) = \pi_{\text{ref}}(o_i|q)\,e^{A_+/\beta} \,/\, Z(q)$. Since $A_+ \geq 0$, we have $e^{A_+/\beta} \geq 1$. Moreover, $Z(q) = \sum_{o_i\in\mathcal{O}^+}\pi_{\text{ref}}(o_i|q)e^{A_+/\beta} + \sum_{o_j\in\mathcal{O}^-}\pi_{\text{ref}}(o_j|q)e^{A_-/\beta} \leq e^{A_+/\beta}$ because $e^{A_-/\beta} \leq 1$. Thus $\pi^*(o_i|q) \geq \pi_{\text{ref}}(o_i|q)$ for $o_i \in \mathcal{O}^+$. □

Finite-sample sharpening behavior. In practice, Equation (1) is approximated with a finite batch $\{o^{(1)},\ldots,o^{(G)}\} \sim \pi_\theta(\cdot|q)$. The resulting empirical objective drives the policy toward a batch-optimal distribution $\hat{\pi}$. Let N_i be the number of occurrences of mode o_i in the batch, with $\sum_i N_i = G$. Then:

Theorem 3.2. *The batch-optimal policy $\hat{\pi}$ is given by*

$$\hat{\pi}(o_i|q) = \frac{1}{Z'(q)}\,\pi_{ref}(o_i|q)\,\exp\left(\frac{N_i\,A(q,o_i)}{\beta G}\right), \tag{3}$$

where $Z'(q) = \sum_j \pi_{ref}(o_j|q)\exp\big(N_j A(q,o_j)/(\beta G)\big)$. See Appendix B for proof.

A single gradient step does not generally reach $\hat{\pi}$, but it can be viewed as a geometric interpolation toward this target (see Theorem C.1). This is not merely an idealization: in practice, the cosine

similarity between clipped (DAPO-style) and unclipped policy gradients computed on the same checkpoints and batches remains above 0.97 throughout training (Appendix K). The qualitative behavior is therefore preserved.

The key difference from the infinite-sample case lies in unsampled modes[1]. For any o_i with $N_i = 0$, Equation (3) gives

$$\hat{\pi}(o_i|q) = \frac{\pi_{\text{ref}}(o_i|q)}{Z'(q)}, \Delta_i = \pi_{\text{ref}}(o_i|q)(\frac{1}{Z'(q)} - 1).$$

Thus, if $Z'(q) > 1$, all unsampled modes are suppressed. Since unsampled modes may include valid reasoning paths, this directly induces over-sharpening.

Theorem 3.3. *(Proof in Appendix D.) Let S^+ and S^- be the indices of sampled correct and incorrect modes. Then the batch partition function $Z'(q)$ in Theorem 3.2 satisfies:*

$$Z'(q) \geq 1 + \frac{1}{\beta G}\big[A_+ \sum_{i \in S^+} N_i \, \pi_{\text{ref}}(o_i \mid q) - |A_-| \sum_{j \in S^-} N_j \, \pi_{\text{ref}}(o_j \mid q)\big].$$

For normalized binary advantages $A_+ = (1 - p^+)/\sigma, A_- = -p^+/\sigma$ with batch accuracy p^+ and standard deviation σ, this bound simplifies to: $Z'(q) \geq 1 + \frac{p^+(1-p^+)}{\beta\sigma}\Delta_\pi$. where $\Delta_\pi = \min_{i \in S^+} \pi_{\text{ref}}(o_i|q) - \max_{j \in S^-} \pi_{\text{ref}}(o_j|q)$.

As shown in Theorem 3.3, the sharpening behavior is governed by the probability gap Δ_π. Whenever $\Delta_\pi > 0$, we see $Z'(q) > 1$ and therefore a uniform suppression of unsampled modes. Intuitively, $\Delta_\pi > 0$ arises when the model is already performing well on the given query. In fact, this also explains many existing methods (e.g., filtering out high-confidence/easy samples (Yu et al., 2025; Xiong et al., 2025; Amin et al., 2025; Bae et al., 2025)) as implicit ways of preventing $Z'(q)$ from becoming too large. Furthermore, while the term $p^+(1 - p^+)/\sigma$ stabilizes updates from skewed batches (vanishing as $p^+ \to \{0, 1\}$), it is insufficient to counteract the intrinsic optimization bias that favors suppressing unsampled modes once model confidence is established.

3.3 Semantic Coupling

We next study how over-sharpening propagates from a query q to a semantically related query q'. For example, two math problems may rely on the same theorem, so an update on one may affect the other. We absorb the sign of the advantage into the gradient direction and make the following assumption.

Assumption 3.1. For positively semantically related pairs (q, o) and (q', o'), their logit gradients are non-negatively aligned: $\nabla_\theta f(q, o)^\top \nabla_\theta f(q', o') \geq 0$.

The batch-optimal policy $\hat{\pi}$ prescribes target logit shifts $\mathbf{y} \in \mathbb{R}^G$, where $y_i = \frac{N_i|A(q,o_i)|}{\beta G}$[2]. With a sufficiently small learning rate (Jacot et al., 2018), fitting these targets yields the parameter update[3]:

$$\Delta\theta = \mathbf{J}_q^\top (\mathbf{J}_q \mathbf{J}_q^\top)^{-1} \mathbf{t} = \mathbf{J}_q^\top \mathbf{K}_{qq}^{-1} \mathbf{y},$$

where $\mathbf{J}_q$ is the Jacobian of the logits with respect to parameters, and $\mathbf{K}_{qq} = \mathbf{J}_q \mathbf{J}_q^\top$ is the empirical kernel matrix on the current batch. The update induces a logit shift at (q', o'):

$$\Delta f(q', o') = \nabla_\theta f(q', o')^\top \Delta\theta = \mathbf{k}_{q'}^\top \mathbf{K}_{qq}^{-1} \mathbf{y},$$

where $\mathbf{k}_{q'}$ is the kernel vector with entries $[\mathbf{k}_{q'}]_i = k((q', o'), (q, o^{(i)})) \triangleq \nabla_\theta f(q', o')^\top \nabla_\theta f(q, o^{(i)})$.

Let $\mathbf{M}(\lambda, \rho) = (\lambda - \rho)\mathbf{I} + \rho \mathbf{1}\mathbf{1}^\top$, which possesses a principal eigenvalue $\lambda + (G - 1)\rho$ associated with the constant vector $\mathbf{1}$, and a $(G - 1)$-degenerate eigenvalue $\lambda - \rho$ for the remaining subspace. Then, we can always find tight spectral bounds to encapsulate the actual spectrum of $\mathbf{K}_{qq}$:

[1] Following the logic of Corollary 3.1, sampled modes ($N_i > 0$) reallocate mass according to advantage signs.

[2] Due to the sign-absorption convention, the advantage term takes its absolute value here.

[3] For readers unfamiliar with the optimization dynamics in the linearized regime, we provide the necessary background in Appendix E.

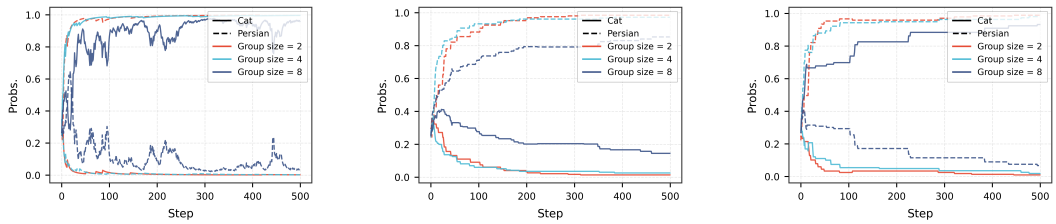


Figure 1: Effect of advantage estimation and group size on over-sharpening. Probability of Cat and Persian over training for raw (left), mean-shifted (mid), and normalized (right) advantages.

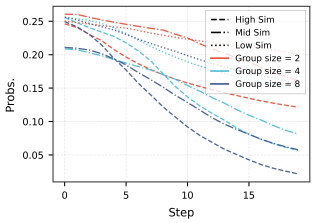


Figure 2: Semantic coupling accelerates collapse. Accuracy on the held-out Siamese class while training only on Persian. Stronger semantic coupling and larger group sizes both lead to faster degradation of Siamese accuracy.

$0 \preceq \mathbf{M}(\lambda_{\min}, \rho_{\min}) \preceq \mathbf{K}_{qq} \preceq \mathbf{M}(\lambda_{\max}, \rho_{\max})$. Here, $\lambda_{\min}$ and $\lambda_{\max}$ bound the diagonal entries (self-alignment of each mode), while $\rho_{\min}$ and $\rho_{\max}$ bound the off-diagonal entries (cross-alignment between different modes). In typical scenarios, updates meant for a particular mode influence that mode more than any other, which translates to $\rho_{\min} \leq \rho_{\max} < \lambda_{\min} \leq \lambda_{\max}$. We formalize this as Assumption 3.3. See Appendix F for empirical validation of Assumption 3.1 and Assumption 3.3. Then, to make the analysis manageable, we make Assumption 3.2 that models the shift from a source query to a target query as a signal decay[4]. Intuitively, η can be interpreted as the similarity between q' and q from the model's perspective. We then obtain the bound on the induced logit shifts, as shown in Theorem 3.4.

Assumption 3.2. Gradient correlations decay by a factor $\eta \in [0, 1]$ when shifting from query q to q', modeled as $k((q', o'), (q, o^{(s)})) \approx \eta K((q, o'), (q, o^{(s)}))$.

Assumption 3.3. Self-alignment consistently exceeds cross-alignment: $\rho_{\min} \leq \rho_{\max} < \lambda_{\min} \leq \lambda_{\max}$.

Theorem 3.4. *Let $\mathcal{A}_{sum} \triangleq \sum_{i=1} y_i$. The induced logit shift $\Delta f(q', o')$ satisfies the following bounds based on whether o' appears in the training batch $\{o^{(i)}, i = 1, \cdots, G\}$:*

- *If $o' \notin \{o^{(i)}\}$, the logit increase is upper-bounded by:*

$$\Delta f(q', o') \leq \frac{\eta \rho_{\max} \mathcal{A}_{sum}}{\lambda_{\min} + (G-1)\rho_{\min}}.$$

- *If $o' = o^{(i)}$ for some $i \in \{1, \ldots, G\}$, the logit increase is lower-bounded by*

$$\Delta f(q', o') = \Delta f(q', o_i) \geq \frac{\eta N_{o'} (\lambda_{\min} - \rho_{\min})}{\lambda_{\max} - \rho_{\max}} (y_k - \frac{\rho_{\max} \mathcal{A}_{sum}}{\lambda_{\max} + (G-1)\rho_{\max}}).$$

See Appendix G for proof.

We leverage Theorem 3.4 to characterize when over-sharpening occurs at q'. Consider a semantically coupled pair (q, q') and suppose that at q' there exists at least one mode o' that appears in the batch

[4]Assumption 3.2 can be relaxed to a bounded interval $[\eta_{\min}, \eta_{\max}]$ for greater generality, which would simply introduce a scaling factor $\frac{\eta_{\min}}{\eta_{\max}}$ to the ratio derived in Corollary 3.2.

(Case 2) and at least one mode $\bar{o}$ that does not (Case 1). Over-sharpening at q' occurs when the logit increase for the batch-seen mode dominates that for the batch-unseen mode, because after normalization the latter becomes suppressed. To quantify this, Corollary 3.2 compare the guaranteed minimum increase for a seen mode (the lower bound of Case 2) to the maximum possible increase for an unseen mode (the upper bound of Case 1).

Corollary 3.2. *Define the ratio $\mathcal{R}$ of the lower bound in Case 2 to the upper bound in Case 1:*

$$\mathcal{R} \triangleq \frac{\lambda_{\min} - \rho_{\min}}{\lambda_{\max} - \rho_{\max}} \left[\frac{N_{o'} y_k}{\mathcal{A}_{sum}} \cdot \left(\frac{\lambda_{\min} + (G-1)\rho_{\min}}{\rho_{\max}} \right) - \frac{\lambda_{\min} + (G-1)\rho_{\min}}{\lambda_{\max} + (G-1)\rho_{\max}} \right].$$

In particular, if $\lambda_{\min} \approx \lambda_{\max} \approx \lambda$ and $\rho_{\min} \approx \rho_{\max} \approx \rho$ (i.e., $\mathbf{K}_{qq}$ is nearly a diagonally dominant matrix with uniform diagonal and off-diagonal entries), $\mathcal{R}$ simplifies to $\left[\frac{N_{o'} y_k}{\mathcal{A}_{sum}} \left(\frac{\lambda}{\rho} + G - 1 \right) - 1 \right]$. See Appendix H for proof.

A large $\mathcal{R}$ means that a batch-seen mode at q' is reinforced much more strongly than an unseen but valid mode, producing over-sharpening through semantic coupling. This is encouraged by:

- large $N_{o'} y_k / \mathcal{A}_{\text{sum}}$, meaning that the update budget concentrates on one overlapping mode.
- large λ/ρ, meaning that features are highly mode-specific and updates leak weakly to other modes.
- larger group size G, which amplifies the relative reinforcement of seen modes.

3.4 Empirical Illustration of Over-Sharpening

We use a controlled softmax classification setting to separately probe sampling bias and semantic coupling. We consider two superclasses (Cat, Dog) and two subclasses (Persian, Siamese) with fixed feature embeddings:

$$\mathbf{e}_{\text{Cat}} = [0.5, 0.5, 0.5, 0.1],\ \mathbf{e}_{\text{Persian}} = [0.75, 0.5, 0.25, 0.1],$$

$$\mathbf{e}_{\text{Dog}} = [0.1, 0.1, 0.1, 0.9],\ \mathbf{e}_{\text{Siamese}} = [0.25, 0.5, 0.75, 0.1].$$

Persian and Siamese align with the Cat superclass but differ in subclass dimensions. Treating inputs as queries and labels as outputs, we train on Persian inputs for 500 SGD steps. At each step, we sample G labels from the current softmax policy and assign reward 1 to Cat or Persian, and 0 otherwise. Thus, both the superclass and subclass labels are correct for Persian. We compare raw, mean-shifted, and normalized advantages, and track $\pi(o = \text{Cat} | q = \text{Persian})$ and $\pi(o = \text{Persian} | q = \text{Persian})$ under different G values, as shown in Figure 1.

Across all settings, the model eventually collapses onto a single label (either Cat or Persian, driven randomly by early sampling noise): once one mode is sampled more often, it dominates subsequent updates. Larger G delays but does not eliminate collapse, and the advantage estimator mainly controls the collapse rate: raw rewards are most aggressive, while normalized advantages slow the process, consistent with our analysis of $Z'(q)$. Momentum and adaptive optimizers such as MI and AdamW often worsen over-sharpening by accumulating early biased gradients (Appendix I).

We next study semantic coupling by monitoring the held-out Siamese subclass ($\pi(o = \text{Siamese} \mid q = \text{Siamese})$) while training on Persian. Our theory predicts that collapse propagation depends on the transfer strength η and the suppression ratio $\mathcal{R}$, with $\mathcal{R}$ increasing with G. To test this, we vary the Siamese embedding similarity by shifting mass from the Persian-specific coordinate to the Siamese-specific coordinate:

$$[0.25, 0.5, 0.75, 0.1], [0.1, 0.5, 0.9, 0.1], [0.0, 0.5, 1.0, 0.1],$$

As shown in Figure 2, Siamese accuracy is strongly squeezed during training. This effect worsens with higher similarity, confirming that increased η (high similarity) accelerates collapse. Similarly, larger G exacerbates over-sharpening. Although this may appear to contradict the role of G in reducing sampling bias, the two effects are distinct: a larger G improves gradient fidelity, but these higher-fidelity updates can more strongly reinforce shared semantic structures and thereby propagate collapse. Finally, to complement this controlled experiment, we provide a multi-hop LLM case study in Appendix I.

4 APPROACH

We propose two complementary mechanisms to keep RLVR in the moderate-sharpening regime. Inverse-success advantage calibration (IAC) controls how strongly each trajectory affects the policy update, while distribution-level calibration (DLC) reshapes the rollout distribution to avoid repeatedly sampling the same high-probability modes. The full pipeline is given in Appendix J and Algorithm 1, together with softmax-case visualizations showing how these methods mitigate over-sharpening.

4.1 INVERSE-SUCCESS ADVANTAGE CALIBRATION

Theorem 3.3 shows that over-sharpening arises when $Z'(q) > 1$, which occurs when $\Delta_\pi > 0$. The inflation of $Z'(q)$ scales with $p^+(1-p^+)\Delta_\pi/(\beta\sigma)$, where p^+ is the batch success rate. Thus, easy queries with high success rates and large probability gaps can contribute disproportionately to suppressing unsampled modes, whereas hard queries contribute less. A natural remedy is therefore to reduce the influence of easy, high-success batches on the partition function. Theorem 3.3 suggests three general mitigation paths: enlarging the group size G so that batches cover more modes (but rollouts already dominate the wall-clock cost of RLVR, and the required G is unknown a priori); filtering out high-success queries entirely (which prunes the solution space and discards their gradient signal); or rebalancing trajectory contributions so that easy queries retain a small but non-zero influence. IAC instantiates the third path: it is a few-line modification with no extra forward passes, and its advantage over filtering is quantified in Table 5.

Idealized inverse-probability reweighting. Let $\tilde{A}_k = A(q, o_k)/\pi_{\text{ref}}(o_k)$. One ideal solution is to reweight advantages by the inverse reference probability:

$$\begin{aligned} Z'(q) &\geq 1 + \frac{1}{\beta G}\sum_{k=1}^{G} N_k \tilde{A}_k \pi_{\text{ref}}(o_k) \\ &= 1 + \frac{1}{\beta G}\sum_{k=1}^{G} N_k \left(\frac{A_k}{\pi_{\text{ref}}(o_k)}\right)\pi_{\text{ref}}(o_k) = 1 + \frac{1}{\beta G}\sum_{k=1}^{G} N_k A_k = 1 + 0 = 1. \end{aligned}$$

where the last equality holds since $\sum_k N_k A_k = 0$ (zero-mean advantages within each batch); the reference-probability bias in the suppression term is thus exactly canceled, and the bound collapses to $Z'(q) \geq 1$. This cancels the reference-probability bias in the bound. However, this ideal solution is not directly usable for two reasons: 1) sequence-level probabilities in LLMs have extremely high variance, and inverse weighting would make gradients unstable, and, 2) negative samples are often partially correct or truncated rather than entirely wrong; excessively amplifying their negative advantages may destroy useful intermediate reasoning.

Inverse-success advantage calibration. We thus adopt a coarser and more stable query-level proxy. IAC calibrates only positive advantages and keeps negative ones unchanged. For each query q, let S^+ be the set of positive trajectories among G rollouts. We define the calibrated advantages:

$$\begin{aligned} \tilde{A}(q, o_k^+) &= A(q, o_k^+) \cdot (G - |S^+|)^\alpha, \quad \alpha \geq 0, \\ \tilde{A}(q, o_k^-) &= A(q, o_k^-), \end{aligned}$$

where o_k^+ and o_k^- denote positive and negative trajectories, respectively. Here, $G - |S^+|$ serves as a difficulty proxy: hard queries with few positive samples receive larger weights, while easy queries are down-weighted. The exponent α controls calibration strength, with $\alpha = 0$ recovering the baseline. In this way, IAC reallocates the advantage budget from easy to difficult queries, limiting excessive growth of $Z'(q)$ and reducing over-sharpening. Leaving negative advantages unchanged further avoids destabilizing updates on partially correct or length-truncated trajectories.

4.2 DISTRIBUTION-LEVEL CALIBRATION

IAC controls the update strength of a batch, but it does not solve a complementary issue: once some valid trajectories are suppressed to very low probability, they are unlikely to reappear in later rollouts. We thus also calibrate the sampling distribution.

A naive approach is to increase the global sampling temperature, but this flattens all tokens uniformly and cannot distinguish saturated modes from under-explored ones. Instead, DLC uses a memory

network to track the empirical rollout distribution. Specifically, we introduce an auxiliary memory network m_ϕ with logit function $f_\phi(q, o)$, trained on sampled trajectories using a standard cross-entropy loss. It assigns higher logits to frequently generated outputs and lower logits to rarely observed ones.

During rollout, DLC constructs a calibrated sampling distribution by subtracting the memory logits from the policy logits as follows:

$$\tilde{f}(q, o) = f_\theta(q, o) - \mu \cdot f_\phi(q, o),\ \pi_{\text{sample}}(o \mid q) \propto \exp\left(\tilde{f}(q, o)\right).$$

where $\mu \geq 0$ controls the strength of memory-based exploration. The memory network intuitively acts as a learned frequency prior: frequently sampled modes are down-weighted, while under-explored modes are relatively promoted. Viewed through Theorem 3.2 and Theorem 3.3, DLC prevents dominant modes from repeatedly accumulating large empirical counts N_i, thereby reducing finite-sample over-sharpening and preserving exploration.

Table 1: Comparison of AVG@8 and PASS@8 performance across six benchmark datasets.

Model	Method	AVG@8 (%)							PASS@8 (%)						
		GSM8K	Math500	LightEval	AIME24	Minerva	O-Bench	Avg.	GSM8K	Math500	LightEval	AIME24	Minerva	O-Bench	Avg.
Qwen3-4B	GRPO	93.72	85.58	70.61	34.58	43.06	48.85	62.73	96.74	91.94	75.43	47.37	54.54	56.44	70.41
	RLOO	93.69	**86.44**	70.24	32.50	43.98	47.91	62.46	96.98	92.22	75.39	43.65	55.01	56.31	69.93
	Reinforce++	94.04	85.43	70.21	36.25	45.63	48.57	63.36	96.45	91.58	74.92	50.54	55.32	56.38	70.87
	DrGRPO	94.10	83.56	69.54	35.42	45.43	47.83	62.65	96.43	89.90	73.87	47.02	54.60	55.08	69.48
	IAC (Ours)	**94.63**	86.43	**72.15**	**37.08**	**47.01**	**48.87**	**64.36**	**97.38**	**92.33**	**77.30**	**52.28**	**57.74**	**56.59**	**72.27**
Qwen3-8B	GRPO	95.59	85.81	71.12	35.42	47.20	49.79	64.16	96.51	91.39	75.19	52.36	55.65	56.88	71.33
	RLOO	95.56	86.89	72.30	36.25	46.78	49.74	64.59	97.25	92.64	76.17	50.92	55.55	57.68	71.70
	Reinforce++	95.56	87.07	72.14	**40.68**	47.84	50.30	65.60	96.96	92.31	75.96	**55.24**	56.12	57.78	72.40
	DrGRPO	95.35	86.11	71.56	34.58	47.38	49.87	64.14	97.11	91.57	75.36	52.25	55.41	57.26	71.49
	IAC (Ours)	**95.72**	**87.67**	**73.85**	35.42	**50.60**	**50.49**	**65.63**	**97.56**	**94.54**	**78.69**	52.67	**60.60**	**58.48**	**73.76**
Qwen3-14B	GRPO	95.96	89.42	73.83	53.45	50.32	55.28	69.71	97.53	93.75	76.54	70.44	59.40	60.87	76.42
	RLOO	96.04	90.29	73.64	54.17	50.46	55.10	69.95	97.43	93.62	76.36	73.18	59.78	61.14	76.92
	Reinforce++	95.82	**90.88**	73.96	54.17	50.87	56.00	70.28	97.79	94.12	76.74	71.71	60.27	61.42	77.01
	DrGRPO	95.96	90.37	73.85	54.85	50.69	55.35	70.18	97.53	94.27	76.57	71.83	61.03	60.96	77.03
	IAC (Ours)	**96.36**	90.70	**74.52**	**56.67**	**52.34**	**56.32**	**71.15**	**98.11**	**94.37**	**77.28**	**78.07**	**62.64**	**62.37**	**78.81**

5 Empirical Evaluation

5.1 Experimental Setup

Models, datasets, and metrics. We evaluate our methods in mathematical reasoning with verifiable rewards. We use the DeepScaleR (Luo et al., 2025) corpus as our primary RLVR training set, consisting of approximately 40.3k problem–answer pairs with checkable labels. To assess generalization, we report performance on six diverse benchmarks: GSM8K, AIME 2024, MATH 500, LightEval, Minerva (Lewkowycz et al., 2022), and O-Bench (He et al., 2024). We conduct experiments with three backbone models, Qwen3-4B, Qwen3-8B and Qwen3-14B (Yang et al., 2025). As evaluation metrics, we report PASS@8 and AVG@8. PASS@8 measures the proportion of queries for which at least one of the eight responses is correct, and AVG@8 reflects the average proportion of correct responses per query. Appendix K provides evaluation results on different models and training datasets, including cross-family (LLaMA-3.1-8B) and code-generation (Qwen2.5-Coder-7B) experiments.

Baselines. We compare our methods against several state-of-the-art RL algorithms tailored for reasoning tasks, including GRPO (DeepSeek-AI, 2025a), DrGRPO (Liu et al., 2025), RLOO (Kool et al., 2019), and Reinforce++ with baseline (Hu et al., 2025); DAPO (Yu et al., 2025) is evaluated in the compatibility analysis (Appendix K). Furthermore, we also compare IAC against sample-filtering-based methods (Bae et al., 2025) in Appendix K. All baselines are run in the same setting with identical reward functions and sampling budgets for fair comparison. Moreover, since DLC introduces non-trivial computational overhead, our default configuration in large-scale experiments employs IAC alone.

Implementation details. All models, including the auxiliary memory network, are optimized with AdamW using a learning rate of 10^{-6}. We employ a global batch size of 64 and sample $G = 8$ responses per query. Training is run for 2500 steps, and we evaluate checkpoints every 50 steps, reporting the best performance for each method. We employ VERL's built-in recipes to standardize the implementation of baseline methods. For IAC, we use $\alpha = 1$ by default. Due to the page limit,

we leave sensitivity analysis (α), the comparison with sample-filtering baselines, and the entropy evolution analysis in Appendix K. All experiments are conducted using 32 H200 GPUs.

5.2 Evaluation Result

Comparison with state-of-the-art methods. Table 1 summarizes the AVG@8 and PASS@8 performance of different methods, respectively, across six mathematical reasoning benchmarks and three Qwen3 backbones. Overall, our method attains the best AVG@8 and PASS@8 in almost all settings across models and datasets, indicating that calibrated sharpening consistently improves both average quality and success probability. In contrast, different baselines occasionally excel on specific benchmarks. For example, Reinforce++ is competitive on AIME24 and MATH500, and RLOO sometimes attains strong averages. On Qwen3-14B, where all baselines are already strong, IAC still raises the overall AVG@8 from roughly 70.2 (Reinforce++ / DrGRPO) to 71.15, and PASS@8 from 77.03 (best baseline) to 78.81. In particular, improvements on harder, less saturated benchmarks (AIME24, Minerva, O-Bench) are more pronounced than on easier ones such as GSM8K, suggesting that our approach is especially beneficial when reasoning requires exploring multiple reasoning modes rather than amplifying a single dominant pattern. On the smaller backbones Qwen3-4B and Qwen3-8B, we observe a consistent pattern.

Compatibility with other methods. We further evaluate compatibility with entropy-regularization techniques (Appendix K). IAC alone already outperforms DAPO on most datasets, and DAPO+IAC yields additive gains (e.g., LightEval PASS@8 improves from 76.62% to 84.80%); adding DLC further improves all datasets, with the full DAPO+IAC+DLC configuration performing best. These results confirm that IAC and DLC synergize with existing state-of-the-art methods like DAPO.

5.3 Toward Large-scale Models

Benchmark	Baseline	Baseline + IAC
CodeFlowBench(Multi)	41.50	**42.40**
SysBench	74.51	**75.58**
MultiChallenge	38.46	**40.66**
MTBench101-GR	8.83	**8.97**
WildBenchV2	30.87	**31.86**

Table 2: Evaluation results on Ling-Flash.

We further evaluate IAC on Ling-Flash, a 104B-parameter linear-attention mixture-of-experts model. Specifically, we use an internal data mixture, which contains 5k instruction-following samples, 2.1k code samples, and 3083 math samples, and train the model for 300 steps. The baseline follows the internal default training recipe. We add IAC on top of this recipe while keeping the data and all other hyperparameters unchanged. To show the generalization performance, we include six benchmarks including CodeFlowBench(Multi) (Wang et al., 2025b)[5], SysBench (Qin et al., 2024), MultiChallenge (Deshpande et al., 2025), MTBench101-GR (Bai et al., 2024)[6], and WildBenchV2 (Lin et al., 2025). As shown in Table 2, IAC consistently improves performance across all evaluated benchmarks. The gains are especially clear on MultiChallenge, SysBench, WildBenchV2, and CodeFlowBench.

6 Conclusion

In this work, we formalized over-sharpening within the RLVR framework, identifying two drivers of distribution collapse: finite-sample bias, which uniformly suppresses unsampled correct modes, and semantic coupling, which propagates this collapse across related tasks through shared parameters. We introduced two plug-and-play calibrations, IAC and DLC, that stabilize the partition function and diversify the sampling process. Extensive evaluations across benchmarks and model scales show consistent performance gains over state-of-the-art baselines.

[5]The multi-turn subset of CodeFlowBench.

[6]The general reasoning subset of MTBench101.

LIMITATIONS

Several limitations of this work should be acknowledged. First, our theoretical analysis relies on assumptions including binary verifiable rewards. Second, the proposed methods introduce additional hyperparameters whose optimal values may vary across models and training setups, requiring tuning. Third, while the memory network in DLC effectively diversifies sampling, it introduces non-trivial computational overhead.

ETHICS STATEMENT

This work focuses on understanding and mitigating distribution collapse in reinforcement learning with verifiable rewards for large language models. All experiments are conducted on publicly available benchmarks. No human subjects, user studies, or collection of personally identifiable information are involved. The proposed methods aim to improve the reliability of LLM reasoning systems, which has positive societal implications for educational tools, scientific problem-solving, and assistive technologies.

REPRODUCIBILITY STATEMENT

We provide complete proofs of all theoretical results in Appendices A to D, G and H, the linearized-optimization background in Appendix E, and empirical validation of all assumptions in Appendix F. Algorithm 1 summarizes the full pipeline of our methods. All experimental settings, including models, training corpus, hyperparameters, and compute, are specified in Section 5; supplementary results across datasets and models, the sensitivity analysis over α, the comparison with filtering-based baselines, and the entropy evolution analysis are provided in Appendix K, and the controlled over-sharpening studies in Appendix I. We will release our full training and evaluation code upon acceptance.

AI USE STATEMENT

In this work, we used generative AI tools for creating and modifying parts of the experimental software code (e.g., training utilities, evaluation scripts, and plotting), which was verified and tested by the authors against the standard VERL-based training pipeline. We have not used generative AI tools for formulating the mathematical claims, assisting in the writing of the proofs, designing the research methodology or experiments, or interpreting the experimental results. Additionally, we used generative AI tools for drafting parts of the manuscript, editing the paper to improve readability, summarizing related literature, and formatting references. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 7319–7328. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.568. URL https://doi.org/10.18653/v1/2021.acl-long.568.

Kareem Amin, Sara Babakniya, Alex Bie, Weiwei Kong, Umar Syed, and Sergei Vassilvitskii. Escaping collapse: The strength of weak data for large language model training. *CoRR*, abs/2502.08924, 2025. doi: 10.48550/ARXIV.2502.08924. URL https://doi.org/10.48550/arXiv.2502.08924.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. *CoRR*, abs/2504.03380, 2025. doi: 10.48550/ARXIV.2504.03380. URL https://doi.org/10.48550/arXiv.2504.03380.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 7421–7454. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.401. URL https://doi.org/10.18653/v1/2024.acl-long.401.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *CoRR*, abs/2506.14758, 2025. doi: 10.48550/ARXIV.2506.14758. URL https://doi.org/10.48550/arXiv.2506.14758.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025. doi: 10.48550/ARXIV.2505.22617. URL https://doi.org/10.48550/arXiv.2505.22617.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025a. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556, 2025b. doi: 10.48550/ARXIV.2512.02556. URL https://doi.org/10.48550/arXiv.2512.02556.

Kaustubh Deshpande, Ved Sirdeshmukh, Johannes Baptist Mols, Lifeng Jin, Ed-Yeremai Hernandez-Cardona, Dean Lee, Jeremy Kritz, Willow E. Primack, Summer Yue, and Chen Xing. Multichallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier llms. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pp. 18632–18702. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.958/.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. *CoRR*, abs/2506.02355, 2025. doi: 10.48550/ARXIV.2506.02355. URL https://doi.org/10.48550/arXiv.2506.02355.

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. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global normalization. *arXiv preprint arXiv:2501.03252*, 2025.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko,

Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *CoRR*, abs/2510.14901, 2025. doi: 10.48550/ARXIV.2510.14901. URL https://doi.org/10.48550/arXiv.2510.14901.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

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. Solving quantitative reasoning problems with language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

Bill Yuchen Lin, Yuntian Deng, Khyathi Raghavi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=MKEHCx25xp.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *CoRR*, abs/2505.22660, 2025. doi: 10.48550/ARXIV.2505.22660. URL https://doi.org/10.48550/arXiv.2505.22660.

Yanzhao Qin, Tao Zhang, Yanjun Shen, Wenjing Luo, Haoze Sun, Yan Zhang, Yujing Qiao, Weipeng Chen, Zenan Zhou, Wentao Zhang, and Bin Cui. Sysbench: Can large language models follow system messages? *CoRR*, abs/2408.10943, 2024. doi: 10.48550/ARXIV.2408.10943. URL https://doi.org/10.48550/arXiv.2408.10943.

Yuda Song, Julia Kempe, and Rémi Munos. Outcome-based exploration for LLM reasoning. *CoRR*, abs/2509.06941, 2025. doi: 10.48550/ARXIV.2509.06941. URL https://doi.org/10.48550/arXiv.2509.06941.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. doi: 10.48550/ARXIV.2507.06261. URL https://doi.org/10.48550/arXiv.2507.06261.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *CoRR*, abs/2506.01939, 2025a. doi: 10.48550/ARXIV.2506.01939. URL https://doi.org/10.48550/arXiv.2506.01939.

Sizhe Wang, Zhengren Wang, Dongsheng Ma, Yongan Yu, Rui Ling, Zhiyu Li, Feiyu Xiong, and Wentao Zhang. Codeflowbench: A multi-turn, iterative benchmark for complex code generation. *CoRR*, abs/2504.21751, 2025b. doi: 10.48550/ARXIV.2504.21751. URL https://doi.org/10.48550/arXiv.2504.21751.

Junkang Wu, Kexin Huang, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Quantile advantage estimation for entropy-safe reasoning. *CoRR*, abs/2509.22611, 2025. doi: 10.48550/ARXIV.2509.22611. URL https://doi.org/10.48550/arXiv.2509.22611.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to LLM reasoning: from rejection sampling to reinforce. *CoRR*, abs/2504.11343, 2025. doi: 10.48550/ARXIV.2504.11343. URL https://doi.org/10.48550/arXiv.2504.11343.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/ARXIV.2503.14476. URL https://doi.org/10.48550/arXiv.2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837, 2025. doi: 10.48550/ARXIV.2504.13837. URL https://doi.org/10.48550/arXiv.2504.13837.

Haomin Zhuang, Yujun Zhou, Taicheng Guo, Yue Huang, Fangxu Liu, Kai Song, and Xiangliang Zhang. Exploring multi-temperature strategies for token- and rollout-level control in RLVR. *CoRR*, abs/2510.08892, 2025. doi: 10.48550/ARXIV.2510.08892. URL https://doi.org/10.48550/arXiv.2510.08892.

A Proof of Theorem 3.1

Theorem A.1. *The maximizer of Equation* (1)*, denoted as* π^**, is given by*

$$\pi^*(o|q) = \frac{1}{Z(q)}\,\pi_{ref}(o|q)\,\exp\!\left(\frac{A(q,o)}{\beta}\right), \tag{4}$$

where $Z(q)$ *is the partition function ensuring normalization.*

Proof. Because the expectation over q is linear and the policy can depend arbitrarily on q, the maximization decomposes across queries. For each fixed q, we omit it from notation for brevity and can independently maximize the per-query objective as follows:

$$J(\pi) = \int \pi(o)A(o)\,do - \beta \int \pi(o)\log\frac{\pi(o)}{\pi_{\text{ref}}(o)}\,do.$$

We introduce a Lagrange multiplier λ for the normalization constraint:

$$\mathcal{L}[\pi,\lambda] = \int \pi(o)A(o)\,do - \beta \int \pi(o)\log\frac{\pi(o)}{\pi_{\text{ref}}(o)}\,do - \lambda\left(\int \pi(o)\,do - 1\right).$$

Taking the functional derivative with respect to $\pi(o)$ (pointwise) yields:

$$\frac{\delta\mathcal{L}}{\delta\pi(o)} = A(o) - \beta\left(\log\frac{\pi(o)}{\pi_{\text{ref}}(o)} + 1\right) - \lambda.$$

Setting the derivative to zero for every o gives:

$$A(o) - \beta\left(\log\pi(o) - \log\pi_{\text{ref}}(o) + 1\right) - \lambda = 0.$$

We rearrange the above equation and obtain:

$$\log\pi(o) = \log\pi_{\text{ref}}(o) + \frac{A(o)}{\beta} - \left(1 + \frac{\lambda}{\beta}\right).$$

We then exponentiate both sides:

$$\pi(o) = \pi_{\text{ref}}(o)\,\exp\!\left(\frac{A(o)}{\beta}\right)\exp\!\left(-1 - \frac{\lambda}{\beta}\right).$$

The factor $\exp(-1-\lambda/\beta)$ is independent of o. Due to $\int \pi(o)\,do = 1$, we have:

$$\pi(o) = \frac{\pi_{\text{ref}}(o)\,\exp\!\left(\frac{A(o)}{\beta}\right)}{\int \pi_{\text{ref}}(o')\,\exp\!\left(\frac{A(o')}{\beta}\right)\,do'}.$$

Restoring the explicit dependence on q, we obtain Theorem 3.1. □

B Proof of Theorem 3.2

Proof. We begin by re-expressing the empirical objective $\hat{J}(\pi)$ at the mode level. Following standard KL-regularized reinforcement learning, we formulate the objective as the expected linear advantage minus the KL divergence relative to the reference policy. Given a query q, the objective is:

$$\hat{J}(\pi) = \sum_i \pi(o_i|q)\left(\frac{N_i}{G}A(q,o_i)\right) - \beta\sum_i \pi(o_i|q)\log\frac{\pi(o_i|q)}{\pi_{ref}(o_i|q)},$$

where we treat π as a functional optimization target over the discrete output space. To find the maximizer $\hat{\pi}$, we introduce a Lagrange multiplier λ for the normalization constraint $\sum_i \pi(o_i) = 1$. The Lagrangian $\mathcal{L}$ is defined as:

$$\mathcal{L}(\pi,\lambda) = \sum_i\left[\pi(o_i)\frac{N_iA_i}{G} - \beta\pi(o_i)\log\frac{\pi(o_i)}{\pi_{ref}(o_i)}\right] - \lambda\left(\sum_i \pi(o_i) - 1\right).$$

Taking the functional derivative with respect to each $\pi(o_i)$ and setting it to zero yields:

$$\frac{\partial \mathcal{L}}{\partial \pi(o_i)} = \frac{N_i A_i}{G} - \beta \left(\log \frac{\pi(o_i)}{\pi_{ref}(o_i)} + 1 \right) - \lambda = 0.$$

Rearranging the terms to isolate the log-probability ratio:

$$\log \frac{\pi(o_i)}{\pi_{ref}(o_i)} = \frac{N_i A_i}{\beta G} - \left(1 + \frac{\lambda}{\beta} \right).$$

Then, we have:

$$\log \hat{\pi}(o_i) = \log \pi_{ref}(o_i) + \frac{N_i A_i}{\beta G} + C,$$

where $C = -(1 + \lambda/\beta)$ is a normalization constant independent of o_i. Applying the constraint $\sum_i \hat{\pi}(o_i) = 1$ yields Theorem 3.2. □

C Distributional Update Geometry

Theorem C.1. *Consider a single gradient ascent step $\theta^+ = \theta + \eta \nabla_\theta \hat{J}(\theta)$ on the empirical objective with a learning rate of η. In the functional space of policies, let π_t denote the policy at step t. Under a first-order Taylor expansion, the updated policy π_{t+1} relates to the current policy π_t and the batch-optimal policy $\hat{\pi}$ as:*

$$\pi_{t+1}(o|q) \propto \pi_t(o|q)^{1-\eta\beta} \cdot \hat{\pi}(o|q)^{\eta\beta}. \tag{5}$$

Proof. The empirical objective $\hat{J}(\theta)$ is defined as:

$$\hat{J}(\theta) = \sum_i \frac{N_i}{G} A_i \log \pi_\theta(o_i) - \beta D_{KL}(\pi_\theta \| \pi_{\text{ref}}).$$

The batch-optimal policy $\hat{\pi}$ satisfies the stationary condition $\nabla_\pi \hat{J} = 0$, which yields $\hat{\pi}(o_i) = \frac{1}{Z'} \pi_{\text{ref}}(o_i) \exp(\frac{N_i A_i}{\beta G})$ according to Theorem 3.2. Taking the log:

$$\frac{N_i A_i}{\beta G} = \log \hat{\pi}(o_i) - \log \pi_{\text{ref}}(o_i) + \log Z'$$

The gradient of $\hat{J}$ with respect to the log-probability $w_i = \log \pi_\theta(o_i)$ is[7]:

$$\frac{\partial \hat{J}}{\partial w_i} = \frac{N_i A_i}{G} - \beta(\log \pi_\theta(o_i) - \log \pi_{\text{ref}}(o_i) + 1).$$

Combining the above two equations:

$$\frac{\partial \hat{J}}{\partial w_i} = \beta \left(\log \hat{\pi}(o_i) - \log \pi_{\text{ref}}(o_i) + \log Z' \right) - \beta \log \pi_\theta(o_i) + \beta \log \pi_{\text{ref}}(o_i) - \beta = \beta \left(\log \hat{\pi}(o_i) - \log \pi_\theta(o_i) \right) + \text{constant}.$$

In functional space, the update in log-space is:

$$\log \pi_{t+1}(o_i) = \log \pi_t(o_i) + \eta \frac{\partial \hat{J}}{\partial w_i} = \log \pi_t(o_i) + \eta\beta(\log \hat{\pi}(o_i) - \log \pi_t(o_i)) + C.$$

Rearranging the terms:

$$\log \pi_{t+1}(o_i) = (1 - \eta\beta) \log \pi_t(o_i) + \eta\beta \log \hat{\pi}(o_i) + C$$

Exponentiating both sides yields the geometric mean:

$$\pi_{t+1}(o_i) \propto \pi_t(o_i)^{1-\eta\beta} \cdot \hat{\pi}(o_i)^{\eta\beta}.$$

□

[7] In standard RLVR practice (e.g., GRPO, DAPO), advantages are normalized to be zero-mean within each batch for each query.

D Proof of Theorem 3.3

Proof. We first derive the first bound. By definition of $Z'(q)$ and partitioning modes into unsampled, correct, and incorrect sets, we have:

$$Z'(q) = \sum_{k \notin S^+ \cup S^-} \pi_{\text{ref}}(o_k \mid q) + \sum_{i \in S^+} \pi_{\text{ref}}(o_i \mid q)\, e^{\frac{N_i A_+}{\beta G}} + \sum_{j \in S^-} \pi_{\text{ref}}(o_j \mid q)\, e^{\frac{N_j A_-}{\beta G}}.$$

Using $e^x \geq 1 + x$, we obtain:

$$\begin{aligned} Z'(q) &\geq \sum_{k \notin S^+ \cup S^-} \pi_{\text{ref}}(o_k \mid q) + \sum_{i \in S^+} \pi_{\text{ref}}(o_i \mid q)\Big(1 + \frac{N_i A_+}{\beta G}\Big) + \sum_{j \in S^-} \pi_{\text{ref}}(o_j \mid q)\Big(1 + \frac{N_j A_-}{\beta G}\Big) \\ &= \sum_{\text{all } k} \pi_{\text{ref}}(o_k \mid q) + \frac{1}{\beta G}\Big[A_+ \sum_{i \in S^+} N_i\, \pi_{\text{ref}}(o_i \mid q) + A_- \sum_{j \in S^-} N_j\, \pi_{\text{ref}}(o_j \mid q)\Big] \\ &= 1 + \frac{1}{\beta G}\Big[A_+ \sum_{i \in S^+} N_i\, \pi_{\text{ref}}(o_i \mid q) - |A_-| \sum_{j \in S^-} N_j\, \pi_{\text{ref}}(o_j \mid q)\Big], \end{aligned}$$

which proves the first bound.

We now specialize to normalized binary advantages. Substitute $A_+ = (1-p^+)/\sigma$ and $A_- = -p^+/\sigma$ into the above bound:

$$Z'(q) \geq 1 + \frac{1}{\beta G \sigma}\Big[(1-p^+) \sum_{i \in S^+} N_i\, \pi_{\text{ref}}(o_i \mid q) - p^+ \sum_{j \in S^-} N_j\, \pi_{\text{ref}}(o_j \mid q)\Big].$$

Using $\sum_{i \in S^+} N_i = G p^+$ and $\sum_{j \in S^-} N_j = G(1-p^+)$, we can express the weighted averages:

$$\bar{\pi}^+ = \frac{\sum_{i \in S^+} N_i\, \pi_{\text{ref}}(o_i \mid q)}{\sum_{i \in S^+} N_i}, \qquad \bar{\pi}^- = \frac{\sum_{j \in S^-} N_j\, \pi_{\text{ref}}(o_j \mid q)}{\sum_{j \in S^-} N_j}.$$

Then we have:

$$\begin{aligned} Z'(q) &\geq 1 + \frac{1}{\beta\sigma}\Big[(1-p^+)p^+\, \bar{\pi}^+ - p^+(1-p^+)\, \bar{\pi}^-\Big] \\ &= 1 + \frac{p^+(1-p^+)}{\beta\sigma}\big(\bar{\pi}^+ - \bar{\pi}^-\big). \end{aligned}$$

By definition,

$$\pi^+_{\min} \leq \bar{\pi}^+, \qquad \bar{\pi}^- \leq \pi^-_{\max},$$

so

$$\bar{\pi}^+ - \bar{\pi}^- \;\geq\; \pi^+_{\min} - \pi^-_{\max} \;=\; \Delta_\pi.$$

Therefore, we see:

$$Z'(q) \;\geq\; 1 + \frac{p^+(1-p^+)}{\beta\sigma}\, \Delta_\pi.$$

If $\Delta_\pi > 0$, the right-hand side is greater than 1. For any unsampled mode o_k with $N_k = 0$, Theorem 3.2 gives

$$\hat{\pi}(o_k \mid q) = \frac{\pi_{\text{ref}}(o_k \mid q)}{Z'(q)} < \pi_{\text{ref}}(o_k \mid q).$$

Thus, all unsampled modes are uniformly suppressed. □

E Background for Section 3.3

The derivation presented here is largely adapted from the Neural Tangent Kernel framework. We now provide the derivation for the parameter update form $\Delta\theta$ and the induced logit shift Δf used in the main text. This derivation relies on the local properties of the neural network function under small parameter updates. For a small optimization step, the change in the model parameters $\Delta\theta$ is

small. Consequently, the change in the network output can be approximated by the first-order Taylor expansion around the current parameters θ:

$$f(x;\theta+\Delta\theta) \approx f(x;\theta) + \nabla_\theta f(x;\theta)^\top \Delta\theta.$$

Consider a training batch with target logit shifts given by the vector $\mathbf{y}$. To achieve this shift locally, the parameter update $\Delta\theta$ must satisfy the linearized constraint:

$$\mathbf{J}_q \Delta\theta = \mathbf{y},$$

where $\mathbf{J}_q$ is the Jacobian matrix of the batch samples. Since the number of parameters is typically much larger than the batch size, this system is under-determined. Standard gradient-based optimizers naturally update the parameters within the subspace spanned by the data gradients (the row space of $\mathbf{J}_q$). This corresponds to finding the solution with the minimum Euclidean norm $\|\Delta\theta\|_2$, representing the steepest direction to fit the local target:

$$\Delta\theta = \arg\min_\delta \|\delta\|_2^2 \quad \text{s.t.} \quad \mathbf{J}_q\delta = \mathbf{y}.$$

The closed-form solution to this problem is given by the Moore-Penrose pseudoinverse:

$$\Delta\theta = \mathbf{J}_q^\top (\mathbf{J}_q \mathbf{J}_q^\top)^{-1}\mathbf{y}.$$

Here, the matrix $\mathbf{K}_{qq} = \mathbf{J}_q\mathbf{J}_q^\top$ captures the local alignment between gradients of the batch samples. Once $\Delta\theta$ is determined, the change in the logit value for any other query-output pair (q', o') is governed by the same local linearization:

$$\Delta f(q', o') \approx \nabla_\theta f(q', o')^\top \Delta\theta.$$

Substituting the expression for $\Delta\theta$, we obtain:

$$\Delta f(q', o') = \nabla_\theta f(q', o')^\top \mathbf{J}_q^\top (\mathbf{J}_q\mathbf{J}_q^\top)^{-1}\mathbf{y} = \mathbf{k}_{q'}^\top \mathbf{K}_{qq}^{-1}\mathbf{y},$$

where $\mathbf{k}_{q'} = \mathbf{J}_q \nabla_\theta f(q', o')$ represents the vector of gradient dot products (kernel values) between the new query and the training batch. This confirms that within a small update step, the propagation of probability mass is dictated by gradient similarity.

F Assumption Validation

We empirically validate Assumptions 3.1 and 3.3 on Qwen3-0.6B. We first randomly extract 1000 pairs of questions (q, q') from GSM8K and, for each query, sample 8 responses. For (q, o) and (q', o'), we compute the inner product of their logit gradients $\nabla_\theta f(q, o)^\top \nabla_\theta f(q', o')$. Across all such pairs, the mean inner product is 1467.50 with a standard deviation of 363.97, indicating that gradients for semantically related outputs are strongly and consistently positively aligned, in line with Assumption 3.1.

We then estimate the empirical kernel matrix $\mathbf{K}_{qq} = \mathbf{J}_q\mathbf{J}_q^\top$ for q. As shown in Figure 3, the diagonal entries (self-alignment) are substantially larger than the off-diagonal entries (cross-alignment), corroborating the diagonal-dominance condition in Assumption 3.3.

G Proof of Theorem 3.4

Theorem G.1. *Let $\mathcal{A}_{sum} \triangleq \sum_{s=1}^G y_s = \frac{1}{\beta G}\sum_{s=1}^G N_s |A(q, o^{(s)})|$. The induced logit shift $\Delta f(q', o')$ satisfies the following bounds based on whether o' appears in the training batch $\{o^{(i)}, i = 1, \cdots, G\}$:*

- *If $o' \notin \{o^{(s)}\}_{s=1}^G$, the logit increase is upper-bounded by:*

$$\Delta f(q', o') \le \frac{\eta\rho_{\max}\mathcal{A}_{sum}}{\lambda_{\min} + (G-1)\rho_{\min}}.$$

- *If $o' = o^{(k)}$ for some $k \in \{1, \ldots, G\}$, the logit increase is lower-bounded by*

$$\Delta f(q', o') = \Delta f(q', o^{(k)}) \ge \frac{\eta N_{o'}(\lambda_{\min} - \rho_{\min})}{\lambda_{\max} - \rho_{\max}} \left(y_k - \frac{\rho_{\max}\mathcal{A}_{sum}}{\lambda_{\max} + (G-1)\rho_{\max}} \right).$$

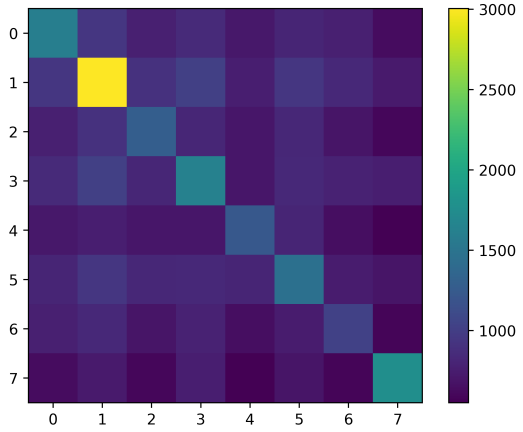


Figure 3: Empirical validation of diagonal dominance on Qwen3-0.6B using GSM8K. Empirical kernel matrix $\mathbf{K}_{qq}$, where diagonal entries (self-alignment) are larger than off-diagonal entries (cross-alignment).

Proof. We define the dual coefficient vector $\boldsymbol{\alpha} = \mathbf{K}_{qq}^{-1}\mathbf{y}$. We proceed by first bounding the entries of $\boldsymbol{\alpha}$.

Step 1: Bounds on $\boldsymbol{\alpha}$. Let $\mathbf{M}(\lambda, \rho) = (\lambda - \rho)\mathbf{I} + \rho\mathbf{1}\mathbf{1}^\top$, where diagonal entries are $\lambda > 0$ and off-diagonal entries are $\rho > 0$. Consider $\boldsymbol{\alpha}(\lambda, \rho) = \mathbf{M}(\lambda, \rho)^{-1}\mathbf{y}$. Applying the Sherman-Morrison formula[8], the inverse $\mathbf{M}(\lambda, \rho)^{-1}$ applied to $\mathbf{y}$ yields:

$$\boldsymbol{\alpha}(\lambda, \rho) = \mathbf{M}(\lambda, \rho)^{-1}\mathbf{y} = \frac{1}{\lambda - \rho}\left(\mathbf{I} - \frac{\rho}{\lambda + (G-1)\rho}\mathbf{1}\mathbf{1}^\top\right)\mathbf{y}.$$

The k-th coefficient of $\boldsymbol{\alpha}(\lambda, \rho)$ is given by:

$$\alpha_k(\lambda, \rho) = \frac{1}{\lambda - \rho}\left(y_k - \frac{\rho\mathcal{A}_{\text{sum}}}{\lambda + (G-1)\rho}\right).$$

Summing over all k, the interaction terms cancel out, simplifying to a dependence on the total mass:

$$\sum_s \alpha_s(\lambda, \rho) = \mathbf{1}^\top\mathbf{M}(\lambda, \rho)^{-1}\mathbf{y} = \frac{\mathcal{A}_{\text{sum}}}{\lambda + (G-1)\rho}.$$

Utilizing non-negativity of y_i, the spectral lower bound $\mathbf{K}_{qq} \succeq \mathbf{M}(\lambda_{\min}, \rho_{\min}) \succ 0$ and the order-reversing property of the matrix inverse, the sum of the coefficients is upper-bounded:

$$\sum_{s=1}^{G} \alpha_s = \mathbf{1}^\top\mathbf{K}_{qq}^{-1}\mathbf{y} \leq \mathbf{1}^\top\mathbf{M}(\lambda_{\min}, \rho_{\min})^{-1}\mathbf{y} = \frac{\mathcal{A}_{\text{sum}}}{\lambda_{\min} + (G-1)\rho_{\min}}.$$

Similarly, we have:

$$\sum_{s=1}^{G} \alpha_s \geq \mathbf{1}^\top\mathbf{M}(\lambda_{\max}, \rho_{\max})^{-1}\mathbf{y} = \frac{\mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}}.$$

For individual dual coefficients α_k, we have the following two-sided bounds:

$$\frac{1}{\lambda_{\max} - \rho_{\max}}\left(y_k - \frac{\rho_{\max}\mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}}\right) \leq \alpha_k \leq \frac{1}{\lambda_{\min} - \rho_{\min}}\left(y_k - \frac{\rho_{\min}\mathcal{A}_{\text{sum}}}{\lambda_{\min} + (G-1)\rho_{\min}}\right).$$

Step 2: Bounds on Logit Shift $\Delta f(q', o')$. The logit change is given by:

$$\Delta f(q', o') = \sum_s \alpha_s k((q', o'), (q, o^{(s)})).$$

[8] $(\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top\mathbf{A}^{-1}}{1 + \mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}$.

According to Assumption 3.2, we have $[\mathbf{k}_{q'}]_s = k((q', o'), (q, o^{(s)})) \approx \eta K((q, o'), (q, o^{(s)}))$.

Case 1: $o' \notin \{o^{(s)}\}_{s=1}^G$**.** Intuitively, for any distinct modes $o_a \neq o_b$, their kernel similarity is bounded by $\rho_{\max}$. Intuitively, the batch $\{o^{(s)}\}$ captures the densest region of the optimization landscape, i.e., the high-probability modes that share the most structural features and thus exhibit the highest gradient overlap. Consequently, any unseen mode $o' \notin \{o^{(s)}\}$ resides in the distribution's tail and is expected to have a lower gradient correlation with the batch samples than the samples have with each other, ensuring the bound holds. Then, we have:

$$[\mathbf{k}_{q'}]_s = k((q', o'), (q, o^{(s)})) \approx \eta K((q, o'), (q, o^{(s)})) \leq \eta\rho_{\max}.$$

Using bounds derived in Step 1, we obtain:

$$\Delta f(q', o') \leq \sum_{s=1}^{G} \alpha_s(\eta\rho_{\max}) = \eta\rho_{\max} \sum_{s=1}^{G} \alpha_s \leq \frac{\eta\rho_{\max}\mathcal{A}_{\text{sum}}}{\lambda_{\min} + (G-1)\rho_{\min}}.$$

Case 2: $o' \in \{o^{(s)}\}_{s=1}^G$**.** Let $S_{o'} = \{s \mid o^{(s)} = o'\}$ be the set of indices where the sample o' appears in the batch, with cardinality $|S_{o'}| = N_{o'}$. The logit shift accumulates contributions from all $N_{o'}$ occurrences (self-terms) and the remaining $G - N_{o'}$ samples (cross-terms):

$$\Delta f(q', o') = \sum_{s \in S_{o'}} \alpha_s [\mathbf{k}_{q'}]_s + \sum_{s \notin S_{o'}} \alpha_s [\mathbf{k}_{q'}]_s.$$

To derive a lower bound, we assume the stiffest kernel parameters ($\lambda_{\max}, \rho_{\max}$) for $\boldsymbol{\alpha}$ (minimizing gradient magnitude) and the loosest parameters ($\lambda_{\min}, \rho_{\min}$) for transfer. Substituting $[\mathbf{k}_{q'}]_k \geq \eta\lambda_{\min}$ and $[\mathbf{k}_{q'}]_j \geq \eta\rho_{\min}$:

$$\begin{aligned}
\Delta f(q', o') &\geq \eta\lambda_{\min} \sum_{s \in S_{o'}} \alpha_s + \eta\rho_{\min} \sum_{s \notin S_{o'}} \alpha_s \\
&= \eta\lambda_{\min} \sum_{s \in S_{o'}} \alpha_s + \eta\rho_{\min} \left(\sum_{all} \alpha_s - \sum_{s \in S_{o'}} \alpha_s \right) \\
&= \eta(\lambda_{\min} - \rho_{\min}) \sum_{s \in S_{o'}} \alpha_s + \eta\rho_{\min}\mathcal{A}_{\text{sum}}.
\end{aligned}$$

Next, we substitute the exact form of α_k derived in Step 1, using the stiff parameters $\lambda_{\max}$ and $\rho_{\max}$:

$$\alpha_k \geq \frac{1}{\lambda_{\max} - \rho_{\max}} \left(y_k - \frac{\rho_{\max}\mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}} \right).$$

Summing this over the $N_{o'}$ indices in $S_{o'}$:

$$\sum_{s \in S_{o'}} \alpha_s \geq \frac{1}{\lambda_{\max} - \rho_{\max}} \left(\sum_{s \in S_{o'}} y_s - N_{o'} \frac{\rho_{\max}\mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}} \right).$$

Substituting this back into the logit shift equation and dropping the non-negative term $\eta\rho_{\min}\mathcal{A}_{\text{sum}}$:

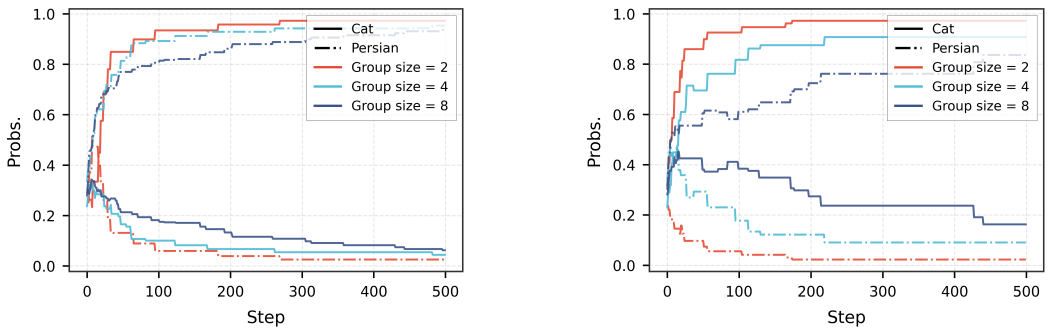
$$\Delta f(q', o') \geq \frac{\eta N_{o'}(\lambda_{\min} - \rho_{\min})}{\lambda_{\max} - \rho_{\max}} \left(y_k - \frac{\rho_{\max}\mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}} \right).$$

□

H Proof of Corollary 3.2

Corollary H.1. *Define the ratio $\mathcal{R}$ of the lower bound in Case 2 to the upper bound in Case 1:*

$$\mathcal{R} \triangleq \frac{\lambda_{\min} - \rho_{\min}}{\lambda_{\max} - \rho_{\max}} \cdot \left[\frac{N_{o'} y_k}{\mathcal{A}_{\textit{sum}}} \cdot \left(\frac{\lambda_{\min} + (G-1)\rho_{\min}}{\rho_{\max}} \right) - \frac{\lambda_{\min} + (G-1)\rho_{\min}}{\lambda_{\max} + (G-1)\rho_{\max}} \right].$$



(a) RLOO (b) Reinforce++ with baseline

Figure 4: The effectiveness of RLOO and Reinforce++ with baseline on over-sharpening.

Under the assumption of a well-conditioned kernel (where $\lambda_{\min} \approx \lambda_{\max} \approx \lambda$ and $\rho_{\min} \approx \rho_{\max} \approx \rho$), this ratio simplifies to:

$$\mathcal{R} \approx N_{o'} \left[\frac{y_k}{\mathcal{A}_{sum}} \left(\frac{\lambda}{\rho} + G - 1 \right) - 1 \right].$$

Proof. Dividing the lower bound of Case 2 by the upper bound of Case 1 in Theorem 3.4:

$$\mathcal{R} = \frac{\frac{\eta N_{o'}(\lambda_{\min} - \rho_{\min})}{\lambda_{\max} - \rho_{\max}} \left(y_k - \frac{\rho_{\max} \mathcal{A}_{\text{sum}}}{\lambda_{\max} + (G-1)\rho_{\max}} \right)}{\frac{\eta \rho_{\max} \mathcal{A}_{\text{sum}}}{\lambda_{\min} + (G-1)\rho_{\min}}}$$

$$= \frac{\lambda_{\min} - \rho_{\min}}{\lambda_{\max} - \rho_{\max}} \cdot \left[\frac{N_{o'} y_k}{\mathcal{A}_{\text{sum}}} \cdot \left(\frac{\lambda_{\min} + (G-1)\rho_{\min}}{\rho_{\max}} \right) - \frac{\lambda_{\min} + (G-1)\rho_{\min}}{\lambda_{\max} + (G-1)\rho_{\max}} \right].$$

Assuming uniform parameters λ and ρ, the stiffness damping factor $\frac{\lambda - \rho}{\lambda - \rho}$ becomes 1. The common term η cancels out. We are left with:

$$\mathcal{R} \approx \frac{N_{o'} \left(y_k - \frac{\rho \mathcal{A}_{\text{sum}}}{\lambda + (G-1)\rho} \right)}{\frac{\rho \mathcal{A}_{\text{sum}}}{\lambda + (G-1)\rho}} = N_{o'} \left[\frac{y_k}{\mathcal{A}_{\text{sum}}} \left(\frac{\lambda}{\rho} + G - 1 \right) - 1 \right].$$

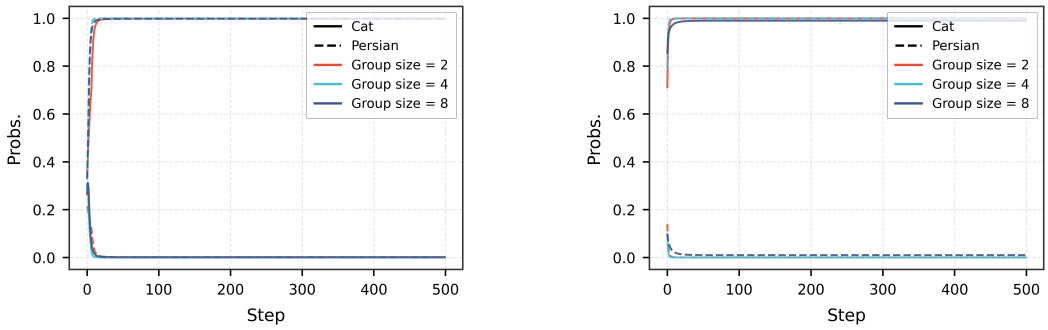
□

I Supplementary Experiment for Section 3.4

Advanced Advantage Estimators. To ensure that the phenomenon of over-sharpening is not an artifact of simplistic advantage estimation, we extend our analysis to include RLOO and Reinforce++ with baseline. As illustrated in Figure 4, while these methods can slightly delay the onset of collapse compared to raw rewards, the policy eventually collapses onto a single mode.

Optimizer effect. As shown in Figure 5, both Momentum optimizer and AdamW optimizer significantly accelerate the rate of collapse compared to standard SGD. We see that, for Momentum optimizer, the policy collapses almost instantaneously (within 20 steps) regardless of group size. This occurs because the momentum buffer accumulates gradients from early samples. Once sampling bias creates a slight preference for one mode (e.g., Cat), the history buffer sustains the update direction even if subsequent batches are more balanced. For AdamW optimizer, we observe a worse trend. By normalizing updates based on the second moment of the gradients, AdamW amplifies the effective learning rate, allowing the model to maximize the likelihood of the initially sampled modes more aggressively.

Multi-hop example. We construct a controlled Monster-Weapon-Material experiment on Qwen3-8B to empirically validate the semantic coupling mechanism. Each monster can be defeated by exactly 2



(a) MI (b) AdamW

Figure 5: Effect of different optimizers on over-sharpening.

Table 3: Monster-Weapon-Material experiment results. IAC mitigates semantic coupling-induced collapse.

Phase	Train Acc.	Test Acc.	Active Modes	Leakage Rate
SFT init	11%	9%	14.7	3%
Early RL (GRPO, 1 epoch)	51%	42%	4.5	14%
Late RL (GRPO, 5 epochs)	84%	23%	1.3	81%
Late RL (IAC, 5 epochs)	81%	**56%**	**3.2**	**31%**

weapons, and each weapon can be crafted from exactly 2 materials, yielding up to 4 valid materials per monster (50 monsters, 50 weapons, 50 materials, 5k QA pairs). We SFT the model for 5 epochs on single-hop QA pairs only (Monster-Weapon and Weapon-Material), then apply RLVR (10 epochs) exclusively on multi-hop queries ("What materials are needed to defeat [Monster]?"), withholding 20% of monsters as a held-out test set. Since the SFT model has never seen direct Monster→Material mappings, accuracy gains during RLVR reflect genuine compositional generalization.

We operationalize modes via two metrics: (1) *Active mode count*: number of distinct materials produced across 32 rollouts per monster; (2) *Cross-query leakage rate*: fraction of responses to held-out test monsters that name a material for training monsters. A rising leakage rate directly signifies semantic coupling-induced collapse propagating from training to test queries. The gradient cosine similarity between training and held-out queries is 0.64. As shown in Table 3, test accuracy first rises as the model discovers the compositional chain, then collapses as over-sharpening on training queries drives leakage. IAC mitigates this collapse, maintaining higher test accuracy and lower leakage.

Furthermore, partitioning test monsters by their average cosine similarity to training monsters into high (> 0.75), medium (0.50–0.75), and low (< 0.50) groups reveals monotonically increasing leakage rates of 91%, 76%, and 54%, respectively, confirming that η (operationalized as gradient cosine similarity) predicts the strength of semantic coupling.

J Supplementary Material for Section 4

Algorithm 1 summarizes the complete pipeline of our methods.

We validate the efficacy of our proposed calibration mechanisms using the toy setting described in Section 3.4. Figure 6 presents the evolution of probability mass. We see that inverse-success advantage calibration alone significantly retards the rate of collapse compared to the uncalibrated baseline. Coupling this with distribution-level calibration (memory network) enables the policy to reach a stable equilibrium where diversity is maintained, effectively arresting over-sharpening. Moreover, despite AdamW's tendency to accelerate collapse, our methods still can prevent the aggressive elimination of alternative modes.

Algorithm 1 Inverse-success Advantage Calibration + Distribution-Level Calibration

Require: policy π_θ, reference π_{ref}, optional memory m_ϕ, batch size B, group size G, calibration strength α, memory weight μ

while not converged **do**
  Sample a mini-batch of queries $\{q_b\}_{b=1}^{B}$ from $\mathcal{D}$;
  for each query q_b **do**
    if memory m_ϕ is used **then**
      Compute calibrated logits $\tilde{f}(q_b, \cdot) = f_\theta(q_b, \cdot) - \mu f_\phi(q_b, \cdot)$;
      Sample G trajectories $\{o_{b,s}\}_{s=1}^{G} \sim \text{softmax}(\tilde{f}(q_b, \cdot))$;
    else
      Sample G trajectories $\{o_{b,s}\}_{s=1}^{G} \sim \pi_\theta(\cdot \mid q_b)$;
    end if
    Compute verifiable rewards $r_{b,s}$ and advantages $A_{b,s}$;
    Identify positive indices $S_b^+ = \{s : A_{b,s} > 0\}$, negatives S_b^-;
    Compute success count $|S_b^+|$;
    for $s \in S_b^+$ **do** ▷ inverse-success calibration
      $\tilde{A}_{b,s} \leftarrow A_{b,s} \cdot (G - |S_b^+|)^\alpha$;
    end for
    for $s \in S_b^-$ **do** ▷ leave negatives unscaled
      $\tilde{A}_{b,s} \leftarrow A_{b,s}$;
    end for
  end for
  Update θ using calibrated advantages $\tilde{A}_{b,s}$ and KL regularization to π_{ref};
  if memory m_ϕ is used **then**
    Update m_ϕ on sampled $(q_b, o_{b,s})$ pairs to track empirical frequencies;
  end if
end while

K Supplementary Experiment for Section 5

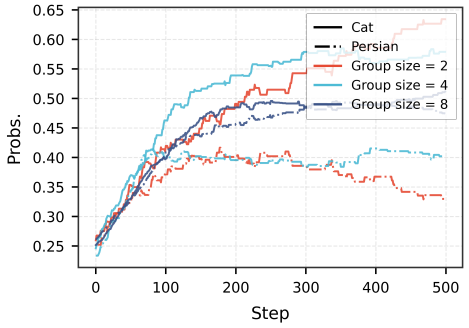
Table 4: Comparison of AVG@8 performance in DeepSeek-R1-Distill-Qwen-1.5B trained on DAPO-Math-17K.

Method	Math500	O-bench
GRPO	80.07	42.65
DAPO	82.25	43.64
IAC (Ours)	84.61	45.54

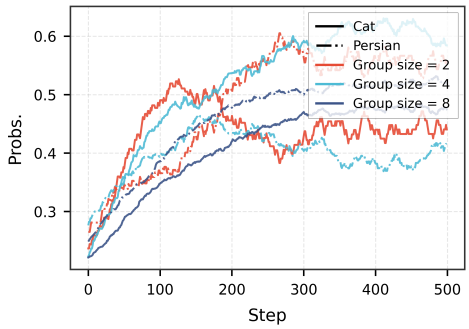
Table 5: Sensitivity of IAC to the calibration strength α. We report AVG@8 and PASS@8 on Math500 and O-Bench for Qwen3-8B when varying α in IAC. When $\alpha = 0$, IAC reduces to the vanilla GRPO objective. We also include a simple filtering baseline that discards the easiest and hardest 25% of training queries (by empirical success rate), retaining only the middle 50% of the difficulty spectrum.

Metric	Dataset	0	0.5	1	1.5	2	Filtering-based (Bae et al., 2025)
AVG@8	Math500	85.81	87.45	**87.67**	86.64	86.29	86.13
	O-Bench	49.79	50.47	50.49	**52.11**	50.17	50.21
PASS@8	Math500	91.39	93.79	**94.54**	94.37	93.85	92.06
	O-Bench	56.88	57.56	58.48	**59.66**	58.88	57.30

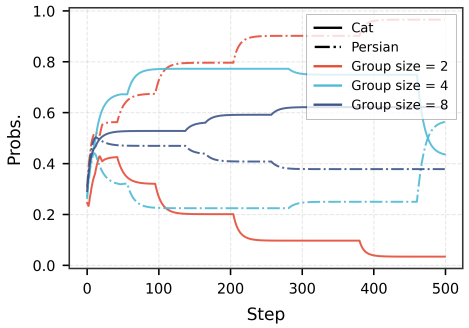
Performance comparison across different datasets and models. We further evaluate IAC on DAPO-Math-17K (Yu et al., 2025) using DeepSeek-R1-Distill-Qwen-1.5B to verify its generality. As shown in Table 4, IAC achieves the best performance, reaching 84.61% on Math500 and 45.54% on O-bench. This represents consistent improvements over GRPO (80.07% / 42.65%) and DAPO (82.25% / 43.64%), demonstrating that IAC can effectively enhance mathematical reasoning capabilities when applied to a different backbone model and training corpus.



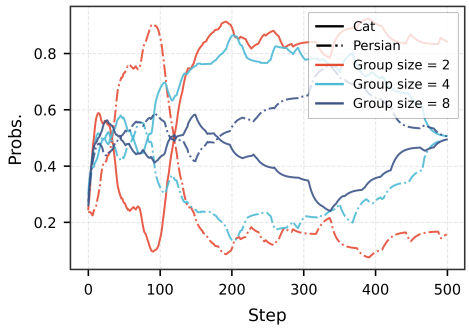
(a) Inverse-success Calibration in SGD



(b) Inverse-success Calibration + Dist. Calib. in SGD



(c) Inverse-success Calibration in AdamW



(d) Inverse-success Calibration + Dist. Calib. in AdamW

Figure 6: We apply our proposed mitigations to the Softmax example. The left column (a, c) employs only inverse-Success advantage calibration, showing a delayed rate of collapse. The right column (b, d) incorporates distribution-level calibration (memory network, $\mu = 0.5$), which arrests collapse and maintains diversity, even under the aggressive updates of AdamW.

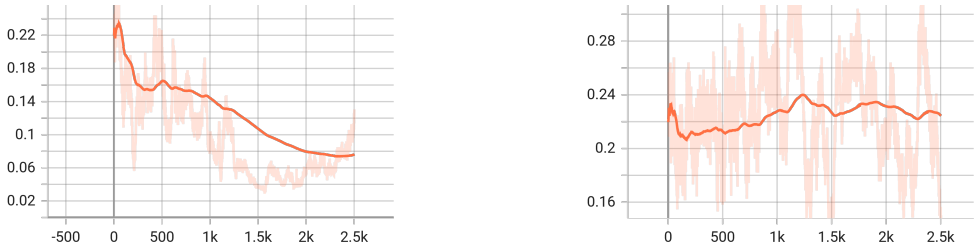
Cross-family and cross-domain generalization. We further evaluate IAC with LLaMA-3.1-8B and in the code-generation domain with Qwen2.5-Coder-7B, which is trained on a mixture of APPS, CodeContests, TACO, and Codeforces. As shown in Table 6, applying IAC improves the PASS@128 of LLaMA-3.1-8B from 76.79% to 79.38% on Math500 and from 45.91% to 49.43% on Minerva, indicating that IAC preserves the exploration capability required under large sampling budgets. As shown in Table 7, IAC outperforms GRPO and DAPO on both HumanEval and LiveCodeBench, demonstrating that mitigating over-sharpening generalizes beyond mathematical reasoning to other verifiable-reward domains.

Table 6: PASS@128 performance of LLaMA-3.1-8B before and after applying IAC.

Method	Math500	Minerva
Baseline	76.79	45.91
Baseline + IAC (Ours)	79.38	49.43

Table 7: AVG@8 performance on code generation with Qwen2.5-Coder-7B.

Method	HumanEval	LiveCodeBench
GRPO	88.3	34.1
DAPO	88.8	34.7
IAC (Ours)	**90.2**	**35.9**



(a) DAPO (b) DAPO+IAC

Figure 7: Evolution of policy entropy under DAPO and DAPO+IAC.

Gradient alignment between clipped and idealized updates. A natural concern is whether the batch-optimal attractor analysis extends to practical updates, since GRPO/DAPO-style objectives clip the importance ratio. To measure this gap directly, we compute the cosine similarity between the clipped (DAPO-style) and unclipped policy gradients on the same checkpoints and the same batched samples over the first 1,000 training steps, aggregated in windows of 100 steps. The similarity stays uniformly high (mean 0.975–0.985, standard deviation 0.006–0.011 across windows), indicating that clipping rarely redirects the update direction in practice and that the geometric-interpolation view in Theorem C.1 remains a faithful description of real optimization trajectories.

Sensitivity to the calibration strength α. Table 5 investigates how α in IAC affects downstream performance on Math500 and O-Bench. Recall that α controls how strongly we up-weight advantages from hard, low-success queries and down-weight those from easy, high-success ones. When $\alpha = 0$, our method reduces exactly to GRPO, and the results at this setting coincide with the GRPO baseline. As α increases from 0 to 1, we observe a consistent improvement on both datasets: AVG@8 and PASS@8 on Math500 peak at $\alpha = 1$, while O-Bench attains its best PASS@8 at a slightly larger $\alpha = 1.5$. Beyond this moderate regime, further increasing α leads to mild degradation. Intuitively, small α under-corrects sampling bias, allowing easy queries to keep dominating the update and driving over-sharpening. In contrast, large α over-corrects, overly suppressing signal from already-mastered queries and making the optimization less sample-efficient. As we can see, $\alpha = 1$ appears to be a good choice in practice.

Table 5 also compares IAC against a filtering strategy that discards the easiest 25% of training queries according to their empirical success rate (Bae et al., 2025). Across both Math500 and O-Bench, filtering yields only modest gains over GRPO and is consistently weaker than IAC. This gap highlights a key limitation of hard filtering: removing easy queries entirely reduces the effective training distribution and discards useful gradient information, particularly about how to maintain correctness on already-solved patterns. IAC, in contrast, keeps all queries in the optimization loop but rescales their influence on the update. Easy queries contribute a smaller—but non-zero—signal, preventing them from overwhelming the batch partition function $Z'(q)$, while hard queries are emphasized in a smooth, data-dependent manner. From a distributional perspective, filtering crudely prunes regions of the solution space, whereas IAC gently rebalances probability mass across difficulty levels. The latter is more effective at preserving useful behaviors on easy instances while still allocating sufficient capacity to explore and stabilize difficult, low-success regimes, which explains the consistent empirical advantage of IAC over filtering-based baselines.

Entropy evolution. Figure 7 tracks the evolution of the policy entropy when applying DAPO alone versus DAPO combined with IAC. Both methods start from similar initial entropy, but their trajectories diverge markedly over training. Under plain DAPO, entropy decreases monotonically and eventually settles around 0.08, signalling that the policy becomes almost deterministic. In contrast, for DAPO+IAC, it stabilizes at a much higher level, fluctuating between 0.20 and 0.24 throughout training. This indicates that the policy retains a substantial degree of stochasticity and continues to explore multiple semantic modes even late in optimization.

Compatibility with DAPO and DLC. We further evaluate the compatibility of the proposed methods using Qwen3-8B, with results summarized in Table 8. First, IAC alone outperforms DAPO on both AVG@8 and PASS@8 metrics across most datasets. Second, when IAC is integrated into DAPO (DAPO+IAC), we observe additive performance gains. For instance, on LightEval (PASS@8), the

combination improves performance from 76.62% (DAPO) to 84.80%. Third, we incorporate DLC with a memory network (Qwen3-1.7B) of $\mu = 0.5$ (DAPO+IAC+DLC). This full configuration yields the highest performance. These results confirm that IAC and DLC are non-redundant enhancements to the RLVR framework that effectively synergize with existing state-of-the-art methods like DAPO.

Table 8: Compatibility analysis using Qwen3-8B.

Metric	Method	Math500	LightEval	Minerva	O-Bench
AVG@8	DAPO	87.25	72.04	47.15	50.17
	IAC	87.67	73.85	50.60	50.49
	DAPO+IAC	87.89	74.48	51.07	50.58
	DAPO+IAC+DLC	**88.44**	**75.42**	**51.90**	**51.38**
PASS@8	DAPO	92.96	76.62	56.38	58.28
	IAC	94.54	78.69	60.60	58.48
	DAPO+IAC	94.74	84.80	61.97	58.88
	DAPO+IAC+DLC	**95.19**	**85.15**	**62.50**	**60.12**