# CALIBRATED BERNOULLI POLICY OPTIMIZATION FOR SINGLE-SAMPLE RLVR

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Group-based reinforcement learning with verifiable rewards (RLVR) has become a widely adopted training paradigm by stabilizing learning through repeated response sampling, but incurs substantial generation cost. We show that, under binary rewards, the GRPO advantage converges as group size grows to the unique positively oriented affine Bernoulli residual with zero conditional mean and unit conditional variance. This characterization motivates that a critic estimating prompt-level success probabilities can replace empirical group statistics, offering a principled route to single sample training. However, imperfect calibration distorts both residual centering and normalization scale. We propose Calibrated Bernoulli Policy Optimization (CBPO) to enable stable and efficient RLVR with only one rollout per prompt. CBPO combines analytical Bernoulli normalization with a prompt-level critic calibrated through squared ECE-style regularization and offline temperature scaling, both updated on evolving rollout. Our analysis bounds advantage mean and variance distortion under explicit calibration conditions. Extensive experiments demonstrate that CBPO matches the training stability and SOTA performance of multi-sample baselines, requiring only a single-sample computational budget.

## 1 INTRODUCTION

The pursuit of complex reasoning capabilities in Large Language Models (LLMs) has fundamentally transformed the traditional alignment paradigm. Moving beyond simple instruction tuning, Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as the core engine driving breakthroughs in mathematical and logical abilities (Shao et al., 2024; DeepSeek-AI et al., 2025; Kimi Team et al., 2025; Yu et al., 2025). In this framework, standard token-level policy optimization methods, such as PPO, struggle to address the credit assignment problem across reasoning chains spanning thousands of steps under sparse rewards (Schulman et al., 2017; 2015; Minsky, 1961; Lightman et al., 2024).

Accordingly, sequence-level optimization has emerged as a natural alternative, treating the complete reasoning chain as a single coherent action. Multi-sample methods such as Group Relative Policy Optimization (GRPO) achieve strong stability by generating multiple responses for each prompt and normalizing rewards within the group (Shao et al., 2024; Yu et al., 2025), but this repeated sampling substantially reduces training throughput. In contrast, single-sample sequence-level methods are more efficient, yet they often become unstable on long-tailed hard reasoning tasks (Ahmadian et al., 2024; Hu, 2025; Wang et al., 2026). We trace this gap to the statistics of binary verifiable rewards. We theoretically prove that, in the infinite sample limit, group normalization implicitly converges to Bernoulli variance normalization, which rescales reward residuals by their natural uncertainty. Without this scaling, the raw Bernoulli residual has difficulty-dependent variance that becomes small near probability boundaries. However, analytical variance normalization can be sensitive to critic probability errors, which affect the residual and its scale and can amplify advantages (Guo et al., 2017; Naeini et al., 2015).

Our contribution builds on the limiting GRPO residual, which is uniquely characterized within the affine class by conditional centering, unit conditional variance, and positive orientation under binary rewards. A learned critic can approximate it with one rollout, but now the critic controls both the



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.

baseline and the normalization denominator. This coupled role explains why inaccurate probabilities can distort the advantage even when the critic provides a useful ranking.

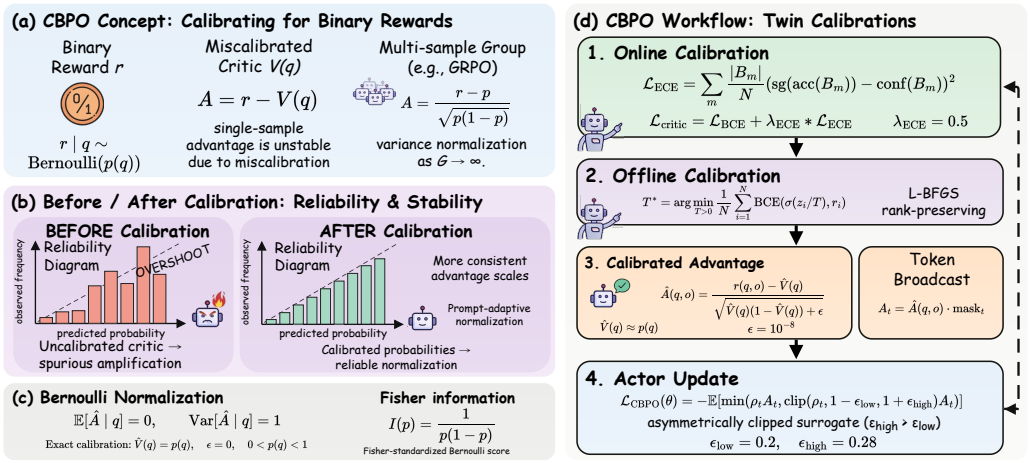


Figure 1: CBPO: motivation and training workflow. (a) Binary rewards yield the Bernoulli normalization target approached by large-group GRPO. (b) Schematic reliability diagrams illustrate how calibration supports more consistent advantage scales. (c) Exact calibration recovers zero mean and unit variance for interior probabilities with $\epsilon = 0$, with a Fisher-standardized Bernoulli score interpretation. (d) Online ECE regularization and offline temperature scaling calibrate the critic. Both calibration stages repeat as training proceeds.

We propose Calibrated Bernoulli Policy Optimization (CBPO), which combines this analytical normalization with an ECE-style critic penalty and rank-preserving temperature scaling (Naeini et al., 2015; Guo et al., 2017). Both calibration components are updated on evolving rollout data. Our analysis bounds advantage distortion when calibration error is controlled away from probability boundaries; it does not assert that these mechanisms guarantee those conditions during training. The Fisher connection is a supporting interpretation of the Bernoulli reward residual, rather than equivalence to natural-gradient optimization in the full policy parameter space (Amari, 1998; Kakade, 2001).

Our contributions are:

- We characterize the binary reward limit of GRPO as the unit variance affine Bernoulli residual, providing an analytical target for single-sample training.
- We propose CBPO, which estimates this target with one rollout per prompt through a calibrated critic. It also combines Bernoulli variance normalization with ECE-style regularization and rank-preserving temperature scaling, both updated on evolving rollout data.
- We establish how critic miscalibration jointly distorts advantage centering and variance, and derive explicit bounds under controlled calibration error away from probability boundaries. Experiments confirm that CBPO matches the stability of multi-sample optimization with single-sample overhead.

## 2 Preliminaries

### 2.1 Reinforcement Learning with Verifiable Rewards

We consider an LLM policy $\pi_\theta$ optimized against a binary verifiable reward $r(o, q) \in \{0, 1\}$, a standard outcome-supervision setting in recent RLVR systems (Shao et al., 2024; DeepSeek-AI et al., 2025; Yu et al., 2025), where $o$ denotes the complete response and $q$ the prompt. We define the prompt-level success probability:

$$p(q) \triangleq \mathbb{P}_{\pi_\theta}(r = 1 \mid q) \tag{1}$$

Since the reward is binary, the outcome for any given prompt can be naturally modeled as a Bernoulli random variable (Cover & Thomas, 2006):

$$r \mid q \sim \text{Bernoulli}(p(q)) \tag{2}$$

with mean $\mathbb{E}[r \mid q] = p(q)$ and variance $\text{Var}[r \mid q] = p(q)(1 - p(q))$. This Bernoulli structure underlies the variance normalization analysis developed in subsequent sections.

### 2.2 Sequence-Level Actor-Critic

In this approach, the critic $V_\phi$ operates exclusively on prompt tokens, producing a single scalar logit $z = f_\phi(q)$ mapped to probability $\hat{p} = \sigma(z)$ via the sigmoid function. This follows the actor-critic tradition of using a learned baseline to reduce variance, but applies it at the sequence level for long-horizon reasoning (Schulman et al., 2015; 2017; Wang et al., 2026). This method replaces group statistics with the learned critic $V_\phi(q)$ as a query-specific baseline, yielding the advantage estimate:

$$\hat{A}_{\text{AC}} = r - V_\phi(q). \tag{3}$$

This eliminates multi-sampling but removes the implicit normalization inherent in GRPO. The training objective is binary cross-entropy against the observed outcomes:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{N} \big[ r_i \log \sigma(z_i) + (1 - r_i) \log(1 - \sigma(z_i)) \big]. \tag{4}$$

### 2.3 Statistical Foundations of Bernoulli Normalization

For a fixed prompt and rollout policy, the raw residual $r - p$ has zero mean but variance $p(1-p)$. Its scale therefore depends on prompt difficulty and vanishes near deterministic outcomes. Our analysis identifies the corresponding standardized target for interior $0 < p < 1$:

$$A^*(r, p) = \frac{r - p}{\sqrt{p(1-p)}}, \qquad \mathbb{E}[A^* \mid p] = 0, \qquad \text{Var}(A^* \mid p) = 1. \tag{5}$$

For an affine residual $a(p)r + b(p)$, these two moment constraints give $b = -ap$ and $a^2 p(1-p) = 1$; positive orientation $a > 0$ uniquely selects $A^*$. This characterization concerns the scalar reward residual, not optimal variance of the full policy gradient. Deterministic rewards at $p \in \{0, 1\}$ cannot satisfy both constraints (Appendix G.2).

This target also explains group normalization. For $G$ independent Bernoulli rewards, the sample mean, unbiased variance, and normalized group residual satisfy (Shao et al., 2024)

$$\mu_G = \frac{1}{G} \sum_{j=1}^{G} r_j, \qquad \sigma_G^2 = \frac{G}{G-1} \mu_G(1 - \mu_G), \qquad \frac{r_i - \mu_G}{\sigma_G} \xrightarrow[G \to \infty]{\text{a.s.}} A^*(r_i, p). \tag{6}$$

Appendix G.1 gives the derivation and finite-group qualifications. The ideal residual also corresponds to the score of a Bernoulli reward distribution normalized by the square root of its Fisher information (Appendix G.7). Together, these results establish a normalization target. The remaining challenge is to approximate it reliably using one rollout and a learned success probability.

## 3 Methodology

CBPO approximates the Bernoulli normalization target in Section 2.3 with one rollout per prompt. Figure 1 connects this statistical motivation to the training workflow: a calibrated prompt critic supplies both the reward baseline and its normalization scale. Online ECE regularization and offline temperature scaling refine these probabilities, enabling prompt-dependent advantage estimation from a single binary outcome. The resulting scalar advantage is broadcast to response tokens for an asymmetrically clipped policy update. Appendix G provides the theoretical derivations.

### 3.1 From the Statistical Target to a Calibrated Critic

Let $s = \hat{V}(q)$ be a fixed critic probability before the actor update and $D(s) = s(1-s) + \epsilon$. The practical residual is $\hat{A} = (r - s)/\sqrt{D(s)}$. At a fixed prompt,

$$\mathbb{E}[\hat{A} \mid q] = \frac{p(q) - s}{\sqrt{D(s)}}, \qquad \mathrm{Var}[\hat{A} \mid q] = \frac{p(q)(1 - p(q))}{D(s)}. \tag{7}$$

As illustrated schematically in Figure 1(b), probability errors affect both advantage centering and scale, with potentially large amplification near probability boundaries despite a fixed $\epsilon > 0$. Calibration therefore improves both the baseline and uncertainty estimate; accurate ranking alone is insufficient, since monotone transformations preserve rankings but can substantially alter the normalization scale.

To state a calibration bound, let $p_s = \Pr(r = 1 \mid \hat{V} = s)$ denote accuracy conditional on the predicted score. If $|p_s - s| \le \delta$ and $s \in [\eta, 1 - \eta]$, where $0 < \eta < 1/2$, then

$$|\mathbb{E}[\hat{A} \mid s]| \le \frac{\delta}{\sqrt{\eta(1-\eta) + \epsilon}}, \qquad |\mathrm{Var}[\hat{A} \mid s] - 1| \le \frac{\delta + \epsilon}{\eta(1-\eta) + \epsilon}. \tag{8}$$

These bounds show how controlling calibration error limits distortion in advantage centering and variance, motivating the calibration procedure below. Derivations are provided in Appendices G.5 and G.4.

### 3.2 Dual Critic Calibration

Binary cross-entropy is a proper probability-scoring objective, but finite data, model misspecification, optimization error, and policy drift can still leave critic predictions miscalibrated. CBPO therefore combines two complementary mechanisms (Figure 1(d)): online ECE regularization encourages agreement between bin confidence and success frequency, while offline temperature scaling adjusts the global probability scale without changing the ranking.

#### 3.2.1 Online ECE Regularization

For a given training batch of $N$ prompts, the critic predictions $\hat{p}_i = \sigma(z_i)$ are partitioned into $M$ equal width bins $\{B_m\}$, and a squared ECE-style penalty encourages agreement between bin-level accuracy and confidence:

$$\mathcal{L}_{\mathrm{ECE}} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left(\mathrm{sg}(\mathrm{acc}(B_m)) - \mathrm{conf}(B_m)\right)^2. \tag{9}$$

where $\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$ is the average predicted probability, and $\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} r_i$ is the true empirical success rate within the bin. The total critic objective is

$$\mathcal{L}_{\mathrm{critic}} = \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{ECE}} \cdot \mathcal{L}_{\mathrm{ECE}}, \tag{10}$$

where $\lambda_{\mathrm{ECE}}$ controls the regularization strength, set to $0.5$ by default. Empty bins contribute zero. For fixed hard-bin memberships, gradients act through bin confidences; the loss is not globally smooth across bin boundaries. This squared training penalty should be distinguished from an absolute-error ECE diagnostic.

#### 3.2.2 Offline Temperature Scaling

After each critic training phase, we perform offline temperature scaling by refitting a positive scalar temperature on the collected rollouts:

$$T^* = \arg\min_{T > 0} \frac{1}{N} \sum_{i=1}^{N} \mathrm{BCE}\big(\sigma(z_i/T), r_i\big), \qquad \hat{V}(q) = \sigma(f_\phi(q)/T^*). \tag{11}$$

We use L-BFGS for this lightweight scalar fit. Positive temperature scaling preserves ranking while adjusting the global logit scale. It is repeatedly refitted, rather than applied only once before training, and introduces no additional response sampling. ECE regularization and repeated refitting adapt to evolving data, but neither guarantees perfect calibration under policy drift. Appendix G.9 explains the restricted calibration family and optimization interpretation.

### 3.3 Calibrated Bernoulli Policy Optimization

#### 3.3.1 Variance-Normalized Advantage

Using the critic probability $\hat{V}(q)$ after the calibration procedure, we define the sequence-level advantage as:

$$\hat{A}(q,o) = \frac{r(q,o) - \hat{V}(q)}{\sqrt{\hat{V}(q)(1-\hat{V}(q)) + \epsilon}}. \tag{12}$$

This scalar advantage is then broadcast to all valid response tokens to guide token-level optimization: $A_t = \hat{A}(q,o) \cdot \mathrm{mask}_t$.

The numerator centers the observed outcome, while the denominator uses the analytical variance of a binary reward. With $\hat{V}(q) = p(q)$, the advantage has zero conditional mean and variance $p(q)(1-p(q))/[p(q)(1-p(q)) + \epsilon] \leq 1$. This normalization reduces variation in advantage scale across prompt difficulty levels. It also enables advantage estimation from one rollout per prompt, avoiding the generation cost of repeated group sampling. Appendix G.6 gives the derivation.

#### 3.3.2 Clipped Surrogate Objective

CBPO optimizes the actor network using an asymmetrically clipped surrogate objective:

$$\mathcal{L}_{\mathrm{CBPO}}(\theta) = -\mathbb{E}_{q,\, o\sim\pi_{\theta_{\mathrm{old}}}}\left[\frac{1}{|o|}\sum_{t=1}^{|o|}\min(\rho_t A_t, \mathrm{clip}(\rho_t, 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})A_t)\right], \tag{13}$$

where $\rho_t = \pi_\theta(y_t \mid q, y_{<t})/\pi_{\theta_{\mathrm{old}}}(y_t \mid q, y_{<t})$ is the token-level importance ratio. The asymmetric bounds, $\epsilon_{\mathrm{high}} > \epsilon_{\mathrm{low}}$, allow a wider increase in token probability.

CBPO integrates calibrated Bernoulli advantages into the clipped policy objective, adapting the learning signal to prompt difficulty while moderating policy updates. This design enables policy optimization from a single binary outcome per prompt, reducing the rollout cost of group-based methods. Algorithm 1 provides the complete procedure.

## 4 Experiments

### 4.1 Experiment Setup

**Implementation Details.** CBPO samples one response for each of 256 prompts per training step using vLLM (Kwon et al., 2023). We train for 500 steps and evaluate every 5 steps. Full optimization, calibration, sequence-length, and hardware settings are provided in Appendix B.

**Training.** To fully evaluate CBPO, we conduct experiments on different LLMs and reasoning paradigms: Qwen2.5-7B-Instruct and Qwen3-8B (w/o thinking) for standard-length mathematical reasoning, and Qwen3-8B (w. thinking) for long-chain reasoning up to 20k tokens (Qwen et al., 2024; Yang et al., 2025). Following prior work, all models are trained on the **DeepScaleR** dataset (Luo et al., 2025). During rollout, we use a decoding temperature of 0.6 and top-p of 0.95.

**Baselines.** We compare against strong reinforcement learning baselines widely used in LLM reasoning: standard PPO, GRPO, GSPO, RLOO, DAPO, and SPPO (Schulman et al., 2017; Shao et al., 2024; Zheng et al., 2025; Ahmadian et al., 2024; Yu et al., 2025; Wang et al., 2026). The comparison uses outcome-only binary rewards. The reported compute comparison uses eight rollouts per prompt for GRPO, GSPO, RLOO, and DAPO, and one for PPO, SPPO, and CBPO. Appendix B summarizes the experimental settings.

**Evaluation.** We evaluate models on six highly challenging mathematical reasoning benchmarks: AIME24, AIME25, AIME26, AMC23, MATH500, and Olympiad. We cite MATH for the MATH500 subset and OlympiadBench for the Olympiad-style benchmark (Hendrycks et al., 2021; He et al., 2024). We use vLLM (Kwon et al., 2023) with a temperature of $0.6$ and top-p of $0.95$

for inference. We report Avg@64 and Pass@32 for AIME and AMC, and Avg@16 and Pass@8 for MATH and Olympiad. Appendix C defines the Pass and Avg estimators, sample counts, and benchmark aggregation, including the hard subset.

### 4.2 Main Results

Table 1 reports the reasoning performance of CBPO and competing RL algorithms across standard and long-chain settings.

| Method | AIME24 | | AIME25 | | AIME26 | | AMC | | MATH | | Olympiad | | AVG | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg@64 | Pass@32 | Avg@64 | Pass@32 | Avg@64 | Pass@32 | Avg@64 | Pass@32 | Avg@16 | Pass@8 | Avg@16 | Pass@8 | Avg | Pass |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | | | | | |
| Base | 9.0 | 34.9 | 8.9 | 34.6 | 7.1 | 26.7 | 47.1 | 87.0 | 56.6 | 62.4 | 28.2 | 42.1 | 26.1 | 48.0 |
| PPO | 12.5 | 39.3 | 9.4 | 40.0 | 8.5 | 31.7 | 50.8 | 92.5 | 58.8 | 68.1 | 28.9 | 48.3 | 28.1 | 53.3 |
| GRPO | 12.7 | 40.5 | 10.0 | **44.3** | 9.4 | 35.8 | 52.1 | **95.2** | 59.8 | 68.4 | 29.7 | 44.5 | 29.0 | 54.8 |
| GSPO | 11.9 | 35.5 | 9.0 | 38.4 | 7.9 | 27.1 | **53.9** | 93.5 | 57.9 | 66.5 | 28.8 | 43.8 | 28.2 | 50.8 |
| RLOO | 12.7 | **42.2** | 9.4 | 41.8 | **10.3** | 29.9 | 50.4 | 90.7 | 60.0 | 64.9 | 30.3 | 49.8 | 28.8 | 53.2 |
| DAPO | 10.7 | 36.5 | 8.9 | 41.0 | 8.9 | 33.5 | 47.8 | 88.6 | **60.9** | 67.5 | 28.7 | 46.3 | 27.6 | 52.2 |
| SPPO | 11.9 | 38.8 | 8.5 | 36.7 | 8.1 | 32.4 | 52.0 | 91.7 | 58.2 | 63.3 | 28.5 | 47.4 | 27.9 | 51.7 |
| CBPO | **13.0** | 41.4 | **11.0** | 42.6 | 9.6 | **36.8** | **53.9** | 95.0 | 59.8 | **69.4** | **31.5** | 49.2 | **29.8** | **55.7** |
| **Qwen3-8B without thinking** | | | | | | | | | | | | | | |
| Base | 24.1 | 66.7 | 21.3 | 43.3 | 15.1 | 56.7 | 65.5 | 92.5 | 61.8 | 70.6 | 35.3 | 51.7 | 37.2 | 63.6 |
| PPO | 36.8 | 71.5 | 28.8 | 62.9 | 25.7 | 65.2 | 78.9 | 95.8 | 65.8 | 74.2 | 37.7 | 55.0 | 45.6 | 70.8 |
| GRPO | 38.5 | 76.7 | 29.3 | 61.8 | 26.5 | 69.0 | 77.1 | 95.0 | 67.3 | 75.9 | 38.4 | 59.4 | 46.2 | 73.0 |
| GSPO | 36.7 | 73.0 | 26.6 | 56.7 | 22.9 | 62.9 | 77.9 | 96.5 | **68.1** | 73.1 | 39.3 | 58.2 | 45.3 | 70.1 |
| RLOO | 38.9 | 76.4 | 29.1 | 64.2 | 25.2 | 69.8 | **81.1** | **100.0** | 65.5 | 73.0 | 41.9 | 60.8 | 46.9 | 74.0 |
| DAPO | 41.3 | **80.6** | 31.0 | **68.1** | 27.7 | 71.9 | 80.8 | 97.1 | 66.7 | 75.1 | 38.6 | 56.6 | 47.7 | 74.9 |
| SPPO | 40.8 | 78.1 | 29.8 | 62.5 | 24.6 | 67.3 | 79.8 | 96.3 | 65.0 | 72.5 | 40.3 | 57.4 | 46.7 | 72.4 |
| CBPO | **41.7** | 80.0 | **33.2** | 67.4 | **28.5** | **73.6** | 80.2 | 99.3 | 67.6 | **76.6** | **42.9** | **62.0** | **49.0** | **76.5** |
| **Qwen3-8B with thinking** | | | | | | | | | | | | | | |
| Base | 61.5 | 76.5 | 46.8 | 63.1 | 56.7 | 79.3 | 89.5 | 100.0 | 70.2 | 80.5 | 44.8 | 61.4 | 61.6 | 76.8 |
| PPO | 66.0 | 83.3 | 56.8 | 82.7 | 59.9 | 81.5 | 91.7 | 100.0 | 73.8 | 84.7 | 47.3 | 64.7 | 65.9 | 82.8 |
| GRPO | 67.7 | 84.2 | 57.8 | 83.3 | **62.6** | 85.0 | 91.0 | 100.0 | 73.4 | **89.7** | 47.2 | 67.1 | 66.6 | 84.9 |
| GSPO | 70.5 | 86.5 | 49.5 | 73.1 | 58.4 | 83.1 | 92.7 | 100.0 | 71.3 | 85.5 | 46.4 | 67.7 | 64.8 | 82.6 |
| RLOO | 68.7 | 84.8 | 55.2 | 78.3 | 60.4 | 85.8 | 92.8 | 100.0 | 72.0 | 87.2 | 47.2 | 68.9 | 66.1 | 84.2 |
| DAPO | 67.3 | 83.4 | 54.7 | 78.8 | 60.0 | 81.5 | 91.0 | 100.0 | 71.7 | 86.3 | 45.8 | 65.5 | 65.1 | 82.6 |
| SPPO | 69.4 | 85.9 | 55.6 | 79.8 | 60.9 | 80.9 | 90.7 | 100.0 | 72.9 | 88.8 | 47.6 | **70.3** | 66.2 | 84.3 |
| CBPO-8k | 69.0 | 83.1 | **58.2** | **84.8** | 61.9 | 83.4 | 92.7 | 100.0 | 72.6 | 88.0 | 46.6 | 66.2 | 66.8 | 84.3 |
| CBPO-20k | **71.8** | **89.2** | 56.8 | 82.7 | 61.2 | **86.5** | **93.2** | 100.0 | **74.4** | 89.5 | **48.5** | 69.6 | **67.7** | **86.2** |

Table 1: Main results across three model settings. All values are percentages. Avg denotes expected accuracy over repeated sampling, and Pass denotes pass rate. The best result in each panel and metric is bolded.

**Aggregate accuracy and longer reasoning.** CBPO has the highest aggregate Avg/Pass in all three panels of Table 1: 29.8/55.7 for Qwen2.5-7B, 49.0/76.5 for Qwen3-8B without thinking, and 67.7/86.2 for CBPO-20k with thinking. Strong baselines remain best on some individual metrics. For long-chain reasoning, CBPO-8k reaches 66.8/84.3 and CBPO-20k improves the aggregate further, supporting the extension of CBPO to longer reasoning horizons. On Qwen3-8B without thinking, CBPO exceeds the strongest baseline, DAPO, by 1.3 Avg and 1.6 Pass points. It also achieves the highest Olympiad Avg in all three settings (31.5, 42.9, and 48.5), extending these gains to challenging competition problems.

**Robustness across seeds.** Table 2 evaluates PPO, GRPO, SPPO, and CBPO across three independent seeds on Qwen3-8B without thinking. CBPO achieves the highest mean on all four metrics, with Avg $48.50 \pm 0.55$ versus $47.29 \pm 0.97$ for GRPO. Its mean gains over the strongest baseline range from 1.21 to 1.46 percentage points.

| Method | Avg ↑ | Pass ↑ | Hard Avg ↑ | Hard Pass ↑ |
|---|---|---|---|---|
| PPO | $45.55 \pm 0.66$ | $70.79 \pm 1.16$ | $32.25 \pm 0.83$ | $63.59 \pm 1.26$ |
| 95% CI | $[43.92, 47.18]$ | $[67.92, 73.66]$ | $[30.20, 34.30]$ | $[60.47, 66.71]$ |
| GRPO | $47.29 \pm 0.97$ | $74.06 \pm 0.96$ | $34.51 \pm 1.19$ | $68.05 \pm 1.19$ |
| 95% CI | $[44.87, 49.71]$ | $[71.68, 76.43]$ | $[31.56, 37.46]$ | $[65.09, 71.00]$ |
| SPPO | $47.27 \pm 0.57$ | $73.30 \pm 0.85$ | $34.48 \pm 0.64$ | $67.21 \pm 0.86$ |
| 95% CI | $[45.87, 48.68]$ | $[71.19, 75.41]$ | $[32.89, 36.08]$ | $[65.07, 69.34]$ |
| **CBPO** | $\mathbf{48.50 \pm 0.55}$ | $\mathbf{75.52 \pm 0.88}$ | $\mathbf{35.83 \pm 0.74}$ | $\mathbf{69.45 \pm 1.16}$ |
| **95% CI** | $[\mathbf{47.13}, \mathbf{49.86}]$ | $[\mathbf{73.33}, \mathbf{77.71}]$ | $[\mathbf{34.00}, \mathbf{37.65}]$ | $[\mathbf{66.58}, \mathbf{72.33}]$ |

Table 2: Three-seed results on Qwen3-8B without thinking: mean $\pm$ SD (percentage points), with 95% Student-$t$ confidence intervals below.

### 4.3 Ablation Analysis

Table 3 reports component ablations on Qwen3-8B without thinking. Bernoulli normalization increases Avg from 0.4673 to 0.4759 and Hard Avg from 0.3390 to 0.3501; ECE-style regularization gives the largest standalone Avg gain (0.4775), and temperature scaling improves Avg to 0.4742. The full combination reaches 0.4901 Avg and 0.7075 Hard Pass. These results support complementary roles for normalization and calibration, while the standalone normalization gain shows that imperfect calibration need not cause immediate failure in every run.

| Method | Norm. | ECE | Temp. | Avg | Pass | Hard Avg | Hard Pass |
|---|---|---|---|---|---|---|---|
| SPPO / None | × | × | × | 0.4673 | 0.7235 | 0.3390 | 0.6633 |
| + Bernoulli Norm. | ✓ | × | × | 0.4759 | 0.7369 | 0.3501 | 0.6808 |
| + ECE Loss | × | ✓ | × | 0.4775 | 0.7394 | 0.3515 | 0.6834 |
| + Temp. Scaling | × | × | ✓ | 0.4742 | 0.7338 | 0.3476 | 0.6769 |
| CBPO full | ✓ | ✓ | ✓ | **0.4901** | **0.7649** | **0.3658** | **0.7075** |

Table 3: Ablation study on Qwen3-8B without thinking. Hard metrics average AIME24, AIME25, AIME26, and Olympiad.

Each component also improves Pass and both Hard metrics over SPPO, with the full model giving the strongest combined result. Figure 6 compares one-factor sweeps of the ECE weight, calibration bin count, and temperature-refitting interval. The default $(\lambda_{\mathrm{ECE}}, M, K) = (0.5, 10, 1)$ achieves the highest Avg and Pass in all three sweeps, supporting the combination of moderate regularization and frequent temperature refitting. Appendix E reports all metrics.

### 4.4 Calibration and Training Dynamics

**Long-horizon reasoning and generation stability.** Figure 2 shows that CBPO sustains extended responses during training. In the thinking setting, its mean response length remains around 7,500–8,000 tokens after the initial adjustment, without the pronounced shortening observed for GSPO or the abrupt drop seen in DAPO. In both non-thinking settings, CBPO also develops longer responses as training progresses. Together with the accuracy gains in Table 1, these trends support the preservation of long reasoning behavior and stable generation lengths in the thinking setting under single-sample training.




Figure 2: Response-length dynamics across three model settings.

**Critic calibration.** Figure 3 shows predictions closer to the diagonal under dual calibration. The binwise penalty encourages probability–frequency agreement; temperature refitting adjusts the global scale while preserving rankings. These static evaluations support their complementary roles, while held-out checkpoints are still needed to assess calibration under policy drift.

**Policy optimization and update cost.** Figure 4 shows CBPO's training reward rising early and remaining above the other displayed trajectories for most of training. Its step time stays close to SPPO and PPO and below the remaining baselines. These curves describe observed training behavior; reward definitions and PG-loss aggregation differ across methods and do not provide a common policy-quality ranking.

Appendix F reports calibration dynamics across four CBPO parameter settings (Figure 7).

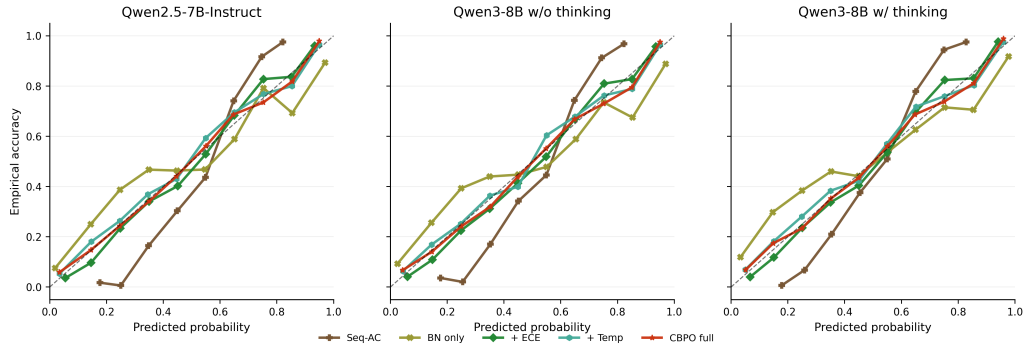


Figure 3: Critic reliability across three model settings. Dual calibration moves predictions toward the diagonal relative to sequence-level actor-critic and normalization alone. These static diagnostics do not establish uniform online calibration.

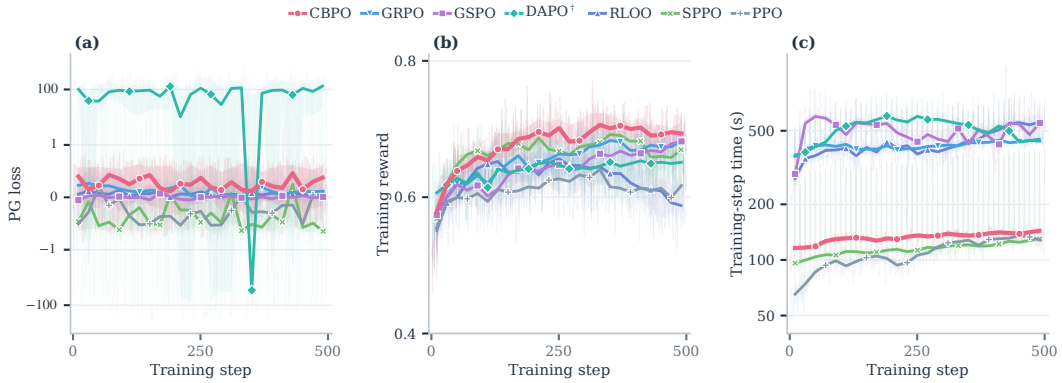


Figure 4: Training dynamics of CBPO and six baselines: (a) policy-gradient loss, (b) training reward, and (c) training-step time. Panels (a) and (c) use symmetric-log and logarithmic scales, respectively. Reward curves are method-specific training signals, not held-out accuracy; PG-loss magnitudes depend on the underlying objective and aggregation.

### 4.5 Compute Efficiency and Equal-Budget Performance

**Computational efficiency.** Table 4 reports that CBPO uses one rollout per prompt and 174.75 GPU-hours, versus 449.14–584.45 GPU-hours for eight-rollout methods, reducing cost by 61.1–70.1%. It achieves 2.89 prompts/s and a step time of 140.65 s, close to 136.51 s for SPPO and 132.72 s for PPO. Calibration adds 0.1158 s per step, accounting for 0.0821% of training-step time and 0.9645% of critic-update time (Table 10).

| Method | Rollouts/prompt | Rollout (s/step) | Step (s) | Evaluation (s) | Prompts/s | Wall-clock (h) | GPU-hours |
|---|---|---|---|---|---|---|---|
| GRPO | 8 | 184.97 | 431.78 | 157.68 | 1.38 | 60.03 | 480.23 |
| DAPO | 8 | 239.06 | 525.52 | 163.86 | 1.07 | 73.06 | 584.45 |
| RLOO | 8 | 193.09 | 403.85 | 154.95 | 1.32 | 56.14 | 449.14 |
| GSPO | 8 | 227.81 | 508.71 | 169.24 | 1.12 | 70.73 | 565.86 |
| PPO | 1 | 84.13 | 132.72 | 152.52 | 3.04 | 18.47 | 147.77 |
| SPPO | 1 | 87.98 | 136.51 | 161.21 | 2.91 | 21.22 | 169.76 |
| CBPO | 1 | 88.63 | 140.65 | 162.56 | 2.89 | 21.84 | 174.75 |

Table 4: Reported computational costs. Prompt throughput is computed as 256 prompts divided by rollout time.

**Joint accuracy and coverage.** At a fixed budget of 100 GPU-hours, Figure 5 shows CBPO leading on both the full benchmark set and the Hard subset. It achieves 45.64 Avg, 72.83 Pass, 32.98 Hard Avg, and 66.15 Hard Pass, exceeding SPPO by 1.52, 2.82, 1.81, and 3.01 percentage points, respectively. Appendix D provides the full numerical results.

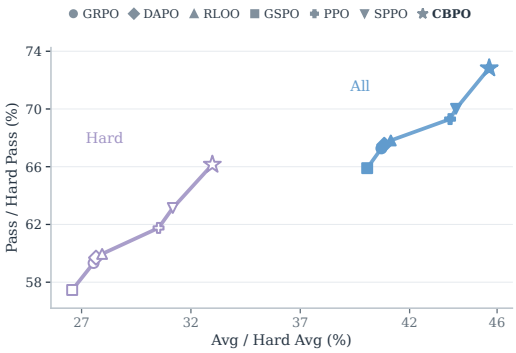


Figure 5: All and Hard scores at 100 GPU-hours. Filled/open markers denote Avg–Pass / Hard Avg–Hard Pass (Appendix D).

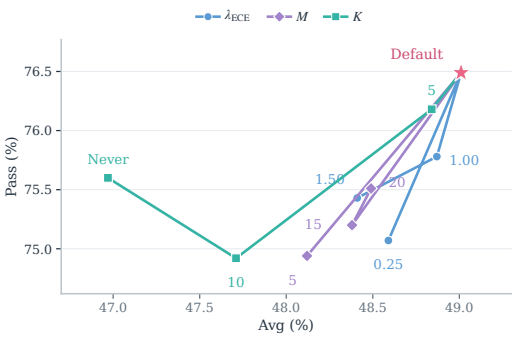


Figure 6: Sensitivity to $\lambda_{\mathrm{ECE}}$, $M$, and $K$. Labels give parameter values; the star marks the default $(0.5, 10, 1)$. Never denotes no refitting.

## 5 Related Work

**Reinforcement Learning with Verifiable Rewards.** RLVR improves LLM reasoning through verifiable outcome rewards, supporting mathematics, coding, theorem proving, and tool use beyond preference based RLHF (Christiano et al., 2017; Cobbe et al., 2021; Jimenez et al., 2024). DeepSeek-Math, DeepSeek-R1, Kimi k1.5, and DAPO demonstrate its potential for long chain reasoning and gains beyond supervised fine tuning (Shao et al., 2024; DeepSeek-AI et al., 2025; Kimi Team et al., 2025; Yu et al., 2025). GRPO and its variants stabilize learning using relative advantages from multiple responses per prompt. CBPO reduces this sampling cost by estimating Bernoulli reward moments from calibrated prompt-level success probabilities with one rollout per prompt.

**Credit Assignment in Long Chain Reasoning.** Delayed outcome rewards complicate credit assignment in long reasoning chains (Minsky, 1961). Token-level PPO relies on value estimates to distribute these sparse rewards across reasoning tokens (Schulman et al., 2017; 2015). Process supervision instead provides stepwise feedback (Lightman et al., 2024). Math-Shepherd (Wang et al., 2024) learns process rewards from automatically generated step labels, while OmegaPRM (Luo et al., 2024) uses divide-and-conquer search to collect process supervision. VinePPO (Kazemnejad et al., 2025) refines credit assignment using Monte Carlo value estimates at intermediate reasoning steps. CBPO uses one rollout per prompt with outcome-only rewards, treating each response as one action and normalizing its residual by the calibrated Bernoulli standard deviation.

**Sequence Level Policy Optimization.** Recent work has increasingly shifted from token level RL toward sequence level policy optimization for LLM reasoning. RLOO (Ahmadian et al., 2024) and REINFORCE++ (Hu, 2025) revisit simple policy gradient methods with sequence level rewards; GRPO (Shao et al., 2024) avoids learned critics through multi sample group baselines; GSPO (Zheng et al., 2025) applies sequence level importance ratios and clipping; and VAPO (Yue et al., 2025) studies value based long CoT reinforcement learning under sparse rewards and heterogeneous sequence lengths. SPPO (Wang et al., 2026) formulates reasoning as a sequence level contextual bandit and uses a scalar prompt level critic to replace empirical group baselines. CBPO extends this formulation with a calibrated success-probability critic that supplies both the baseline and Bernoulli variance for single-sample advantage normalization.

## 6 Conclusion

CBPO connects single-rollout RLVR to the standardized Bernoulli residual implicit in large-group normalization. Its central design follows from a calibration barrier: a critic controls both residual centering and scale. Squared ECE-style regularization and repeated temperature fitting improve the empirical approximation, while the theory remains conditional on calibration quality and interior probabilities. The original reasoning evaluations and added compute-normalized results support a practical efficiency advantage in the tested binary-reward setting; broader reward types and sustained calibration under policy drift require further evidence.

## Reproducibility Statement

We provide the complete algorithm in Appendix A, hyperparameters and training configurations in Appendix B, and evaluation protocols in Appendix C. The supplementary materials include the accompanying code and data to facilitate reproduction of our experiments and analyses.

## AI Use Statement

Large language models were used to assist with manuscript drafting, language polishing, and translation, improving clarity, readability, and logical flow. They also assisted in preparing LaTeX code and plotting scripts for figures and tables, and in checking the mathematical rigor and logical consistency of the equations.

Scientific decisions concerning the research objectives, methodology, experimental design, and conclusions remained with the authors. AI tools provided assistance under author direction and did not independently determine these decisions.

The authors take full responsibility for the manuscript, including all AI-assisted content, and for ensuring its accuracy, originality, and compliance with applicable AI-use policies and research ethics standards.

## References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. 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)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30*, pp. 4299–4307, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

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

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. ISBN 978-0-471-24195-9. URL http://www.elementsofinformationtheory.com/.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/arXiv.2501.12948. URL https://arxiv.org/abs/2501.12948.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://arxiv.org/abs/2103.03874.

Jian Hu. REINFORCE++: A simple and efficient approach for aligning large language models. *CoRR*, abs/2501.03262, 2025. doi: 10.48550/arXiv.2501.03262. URL https://arxiv.org/abs/2501.03262.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/4b86abe48d358ecf194c56c69108433e-Abstract.html.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron C. Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in RL training of llms. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, 2025. URL https://openreview.net/forum?id=Myx2kJFzAn.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *CoRR*, abs/2501.12599, 2025. doi: 10.48550/arXiv.2501.12599. URL https://arxiv.org/abs/2501.12599.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024. doi: 10.48550/arXiv.2406.06592. URL https://arxiv.org/abs/2406.06592.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL, 2025. URL https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961. doi: 10.1109/JRPROC.1961.287775. URL https://doi.org/10.1109/JRPROC.1961.287775.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2901–2907, 2015. URL https://ojs.aaai.org/index.php/AAAI/article/view/9602.

Qwen, 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, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *CoRR*, abs/1506.02438, 2015. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. 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. 9426–9439. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.510. URL https://doi.org/10.18653/v1/2024.acl-long.510.

Tianyi Wang, Yixia Li, Long Li, Yibiao Chen, Shaohan Huang, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. SPPO: Sequence-level PPO for long-horizon reasoning tasks. *CoRR*, abs/2604.08865, 2026. doi: 10.48550/arXiv.2604.08865. URL https://arxiv.org/abs/2604.08865.

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, Jianxin Yang, Jiaxi Yang, Jing Zhou, 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://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/arXiv.2503.14476. URL https://arxiv.org/abs/2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *CoRR*, abs/2504.05118, 2025. doi: 10.48550/arXiv.2504.05118. URL https://arxiv.org/abs/2504.05118.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *CoRR*, abs/2507.18071, 2025. doi: 10.48550/arXiv.2507.18071. URL https://arxiv.org/abs/2507.18071.

## A Complete Algorithm

Algorithm 1 summarizes the full training procedure of CBPO.

**Algorithm 1** Calibrated Bernoulli Policy Optimization

**Input:** Policy $\pi_\theta$, critic $V_\phi$, temperature $T$, ECE weight $\lambda_{\mathrm{ECE}}$, stability constant $\epsilon$, clip bounds $\epsilon_{\mathrm{low}}, \epsilon_{\mathrm{high}}$

**for** each training iteration **do**
  Sample prompts $\{q_i\}_{i=1}^{B}$ and generate one response $o_i \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid q_i)$
  Compute binary rewards $r_i = r(q_i, o_i) \in \{0, 1\}$ and store old token log probabilities
  Compute critic logits $z_i = f_\phi(q_i)$ and calibrated probabilities $\hat{V}_i = \sigma(z_i/T)$
  Compute and freeze $\hat{A}_i = (r_i - \hat{V}_i)/\sqrt{\hat{V}_i(1-\hat{V}_i) + \epsilon}$
  Broadcast $\hat{A}_i$ to tokens as $A_{i,t} = \hat{A}_i \cdot \mathrm{mask}_{i,t}$
  **for** $k = 1, \ldots, K_{\mathrm{critic}}$ **do**
    Recompute $z_i = f_\phi(q_i)$; partition $\sigma(z_i)$ into $M$ bins, omitting empty bins
    Compute $\mathcal{L}_{\mathrm{ECE}} = \sum_{m=1}^{M} \frac{|B_m|}{B} \big(\mathrm{sg}(\mathrm{acc}(B_m)) - \mathrm{conf}(B_m)\big)^2$
    Update $V_\phi$ using $\mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{ECE}}\mathcal{L}_{\mathrm{ECE}}$
  **end for**
  Recompute post-update logits; every $K_T$ iterations (default $K_T = 1$), refit $T > 0$ with L-BFGS using Eq. 11; retain $T$ otherwise
  **for** $k = 1, \ldots, K_{\mathrm{actor}}$ **do**
    Compute $\rho_{i,t} = \pi_\theta(y_{i,t} \mid q_i, y_{i,<t}) / \pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid q_i, y_{i,<t})$
    Update $\pi_\theta$ with the clipped CBPO objective using $A_{i,t}$
  **end for**
**end for**

*Note:* $\mathrm{acc}(B_m) = |B_m|^{-1} \sum_{i \in B_m} r_i$, $\mathrm{conf}(B_m) = |B_m|^{-1} \sum_{i \in B_m} \sigma(z_i)$, and $\mathrm{sg}(\cdot)$ denotes stop-gradient.
The newly fitted critic and temperature are used at the next iteration; current actor advantages stay fixed.

**Output:** Updated policy $\pi_\theta$, critic $V_\phi$, and temperature $T$

## B Baseline Implementation Details and Reproducibility

The comparisons use the same binary outcome-reward setting and benchmark metrics. Table 5 summarizes the settings recorded for the evaluated methods. Method-specific update rules follow the cited PPO, GRPO, GSPO, RLOO, DAPO, and SPPO papers.

**CBPO implementation.** For each CBPO training step, we used a batch of 256 prompts and sampled one response per prompt. CBPO used actor learning rate $1 \times 10^{-6}$, critic learning rate $1 \times 10^{-5}$, PPO mini-batch size 32, and actor and critic micro-batch size 4 per GPU. Rollouts were generated

using vLLM (Kwon et al., 2023). For Qwen2.5-7B and Qwen3-8B without thinking, the maximum prompt and response lengths were 2048 and 8192, respectively, and training used $8\times$A800 GPUs with 80GB memory (Qwen et al., 2024; Yang et al., 2025). For Qwen3-8B with thinking, the maximum response length was extended to 20k tokens and training used $8\times$H800 GPUs with 80GB memory (Yang et al., 2025). We used asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. For calibration, we set $\lambda_{\text{ECE}} = 0.5$, used 10 ECE bins, enabled temperature scaling, and set the Bernoulli normalization constant to $\epsilon = 10^{-8}$ (Naeini et al., 2015; Guo et al., 2017). Training lasted 500 steps, with validation before training and evaluation every 5 steps.

| Setting | Recorded value or scope |
|---|---|
| Training data | DeepScaleR |
| Models | Qwen2.5-7B-Instruct; Qwen3-8B, with and without thinking |
| CBPO prompt batch | 256 prompts per training step |
| Rollouts per prompt | GRPO/DAPO/RLOO/GSPO: 8; PPO/SPPO/CBPO: 1 in the compute comparison |
| CBPO learning rates | Actor $10^{-6}$; critic $10^{-5}$ |
| CBPO update batches | Mini-batch 32; actor/critic micro-batch 4 per GPU |
| Training / evaluation decoding | Temperature 0.6 / 0.6; top-$p = 0.95$ |
| Maximum prompt length | 2048 tokens |
| Maximum response length | 8192 tokens for standard runs; 20k for the long-thinking run; the original table also reports CBPO-8k |
| Training hardware | $8\times$A800 (80GB) for standard runs; $8\times$H800 (80GB) for long thinking |
| Training schedule | 500 steps; validation before training and evaluation every 5 steps |
| CBPO clipping | $\epsilon_{\text{low}} = 0.2$; $\epsilon_{\text{high}} = 0.28$ |
| CBPO calibration | $\lambda_{\text{ECE}} = 0.5$; 10 equal-width bins; positive temperature refit after each critic-training phase |
| Normalization stabilizer | $\epsilon = 10^{-8}$ |
| Evaluation samples | AIME/AMC: $n = 64$, Pass@32; MATH/Olympiad: $n = 16$, Pass@8 |
| Multi-seed comparison | Three seeds for PPO, GRPO, SPPO, CBPO on Qwen3-8B without thinking |

Table 5: Experimental settings; method-specific entries are identified explicitly.

**Training–evaluation overlap.** Our dataset check identified no overlapping benchmark instances between the evaluation sets and the DeepScaleR training data. The reported training composition includes AIME 1984–2023, pre-2023 AMC problems, Omni-MATH, and STILL; the evaluation uses AIME24–26, AMC23, MATH500, and Olympiad. AIME and AMC evaluation problems are drawn from years excluded from their respective training subsets.

**Timing scope.** Table 4 preserves the measured timing entries. Its throughput is rollout throughput: for CBPO, $256/88.63 = 2.89$ prompts/s, whereas dividing by the full step time gives approximately 1.82 prompts/s. GPU-hours agree with eight times wall-clock hours up to rounding. Mean step time and total elapsed time are different reported quantities and should not be used interchangeably.

## C Evaluation Metrics and Aggregation

For each query $q$, we sample $n$ responses and let $c_q$ denote the number of correct ones. Following the unbiased pass@$k$ estimator used in the OpenAI evaluation protocol (Chen et al., 2021), we compute

$$\text{Pass@}k = \mathbb{E}_{q\sim\mathcal{Q}}\left[1 - \frac{\binom{n-c_q}{k}}{\binom{n}{k}}\right], \tag{14}$$

$$\text{Avg@}n = \mathbb{E}_{q\sim\mathcal{Q}}\left[\frac{c_q}{n}\right]. \tag{15}$$

Equations 14 and 15 define our two evaluation metrics: $\text{Pass@}k$ estimates the probability of obtaining at least one correct solution among $k$ selected samples, while $\text{Avg@}n$ measures average per-sample accuracy. We estimate Pass@32 from $n = 64$ samples for AIME and AMC, and Pass@8

from $n = 16$ samples for MATH and Olympiad, which uses the full set of $n$ samples when estimating success among $k$ draws. Avg and Pass aggregate the six benchmarks with equal weight; Hard Avg and Hard Pass average AIME24, AIME25, AIME26, and Olympiad.

## D EQUAL-BUDGET METRIC COMPARISONS

Table 6 reports the full numerical results at 100 GPU-hours underlying Figure 5. Avg and Pass aggregate six benchmarks; the Hard metrics cover AIME24, AIME25, AIME26, and Olympiad. CBPO achieves the highest scores on all four metrics, exceeding SPPO by 1.52 Avg, 2.82 Pass, 1.81 Hard Avg, and 3.01 Hard Pass points. The gains on both benchmark groups support improved accuracy and solution coverage within the same training budget.

| Method | Avg ↑ | Pass ↑ | Hard Avg ↑ | Hard Pass ↑ |
|---|---|---|---|---|
| GRPO | 40.71 | 67.27 | 27.55 | 59.33 |
| DAPO | 40.84 | 67.52 | 27.66 | 59.70 |
| RLOO | 41.14 | 67.82 | 27.94 | 59.96 |
| GSPO | 40.06 | 65.90 | 26.57 | 57.46 |
| PPO | 43.85 | 69.30 | 30.52 | 61.76 |
| SPPO | 44.12 | 70.01 | 31.17 | 63.14 |
| **CBPO** | **45.64** | **72.83** | **32.98** | **66.15** |

Table 6: Performance (%) at 100 GPU-hours. Hard metrics average AIME24, AIME25, AIME26, and Olympiad.

## E CALIBRATION SENSITIVITY AND OVERHEAD

We vary the ECE weight $\lambda_{\mathrm{ECE}}$, bin count $M$, and refitting interval $K$ one at a time, holding the other parameters at $0.5$, 10, and 1. The default configuration achieves the highest aggregate Avg and Pass and the lowest reported Eval ECE and $D_{\mathrm{var}}$ across these sweeps.

Tables 7–9 report the sensitivity results, with task scores expressed as fractions. The variance diagnostic is

$$D_{\mathrm{var}} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \widehat{\mathrm{Var}}(\hat{A}_i : i \in B_m) - 1 \right|. \tag{16}$$

It measures empirical deviation from unit advantage variance within bins; it is not a gradient-variance guarantee. With nonzero $\epsilon$, even exactly estimated near-boundary probabilities need not yield variance one.

| $\lambda_{\mathrm{ECE}}$ | Avg | Pass | Hard Avg | Hard Pass | Eval ECE ↓ | $D_{\mathrm{var}}$ ↓ |
|---|---|---|---|---|---|---|
| 0.25 | 0.4859 | 0.7507 | 0.3533 | 0.6842 | 0.29 | 0.29 |
| **0.50** | **0.4901** | **0.7649** | **0.3658** | **0.7075** | **0.12** | **0.15** |
| 1.00 | 0.4887 | 0.7578 | 0.3594 | 0.6949 | 0.31 | 0.20 |
| 1.50 | 0.4841 | 0.7543 | 0.3590 | 0.6970 | 0.44 | 0.33 |

Table 7: Reported sensitivity to the ECE regularization weight, holding $M = 10$ and refit interval $K = 1$. Task-performance entries are fractions; Eval ECE and $D_{\mathrm{var}}$ retain their reported scales. Bold denotes the default configuration.

| Bins ($M$) | Avg | Pass | Hard Avg | Hard Pass | Eval ECE ↓ | $D_{\mathrm{var}}$ ↓ |
|---|---|---|---|---|---|---|
| 5 | 0.4812 | 0.7494 | 0.3525 | 0.6994 | 0.39 | 0.36 |
| **10** | **0.4901** | **0.7649** | **0.3658** | **0.7075** | **0.12** | **0.15** |
| 15 | 0.4838 | 0.7520 | 0.3600 | 0.6952 | 0.23 | 0.21 |
| 20 | 0.4849 | 0.7551 | 0.3598 | 0.6981 | 0.28 | 0.27 |

Table 8: Reported sensitivity to the number of calibration bins, holding $\lambda_{\mathrm{ECE}} = 0.5$ and $K = 1$. Task-performance entries are fractions; diagnostic scales are unchanged.

Increasing $\lambda_{\mathrm{ECE}}$ from 0.5 to 1.5 reduces Avg from 0.4901 to 0.4841 and increases the reported Eval ECE from 0.12 to 0.44. Changing the bin count to 5, 15, or 20 also reduces aggregate Avg/Pass relative to the default. Disabling temperature refitting reduces Avg from 0.4901 to 0.4697 and raises $D_{\mathrm{var}}$ from 0.15 to 0.46. Less frequent refits lower aggregate performance, although the hard metrics and calibration diagnostics need not vary monotonically with the interval. The sweeps are local to the evaluated setting and are not a complete hyperparameter interaction or multi-seed sensitivity study.

| Refit interval ($K$) | Avg | Pass | Hard Avg | Hard Pass | Eval ECE ↓ | $D_{\text{var}}$ ↓ |
|---|---|---|---|---|---|---|
| **1** | **0.4901** | **0.7649** | **0.3658** | **0.7075** | **0.12** | **0.15** |
| 5 | 0.4884 | 0.7618 | 0.3538 | 0.6945 | 0.30 | 0.29 |
| 10 | 0.4771 | 0.7492 | 0.3620 | 0.7018 | 0.19 | 0.22 |
| Never, $T = 1$ | 0.4697 | 0.7560 | 0.3547 | 0.6959 | 0.34 | 0.46 |

Table 9: Reported sensitivity to the temperature-refitting interval, holding $\lambda_{\text{ECE}} = 0.5$ and $M = 10$. $K$ counts training iterations. Task-performance entries are fractions; diagnostic scales are unchanged.

### E.1 Incremental Calibration Cost

| Method | Total (s) | Temperature (s) | ECE (s) | Of train step | Of critic update |
|---|---|---|---|---|---|
| CBPO | 0.1158 | 0.1001 | 0.0157 | 0.0821% | 0.9645% |

Table 10: Reported calibration overhead per training step. Fractions and timings are transcribed at the reported precision.

The two calibration components take 0.1158 s per step, approximately 0.08% of training time. These scalar calibration operations reuse collected rollouts and require no additional response generation. The full critic cost remains part of CBPO's end-to-end training cost in Table 4.

## F Calibration Dynamics Across Configurations

**Configurations.** Figure 7 compares four CBPO configurations with one rollout per prompt. For Qwen2.5-7B, C1/C2 use 1.5B/7B critics, actor mini-batches 32/64, and micro-batches 4/8, with $(\lambda_{\text{ECE}}, M) = (0.5, 10)$. For Qwen3-8B, both configurations use an 8B critic, with $(\lambda_{\text{ECE}}, M) = (1, 15)$ for C1 and $(1.5, 20)$ for C2. The training runs span 250 and 500 steps, respectively; these runs use training and evaluation temperatures of 0.6.

**Calibration across parameter settings.** Figure 7 shows decreasing critic objectives across all four configurations. Comparing the first and last 20 training observations, the squared calibration penalty decreases by 29.2–36.0% and the BCE term by 29.1–46.7% within each run. Fitted temperatures settle near one after the initial adjustment. These consistent trends support effective critic fitting and adaptive probability scaling across the tested model and calibration settings.

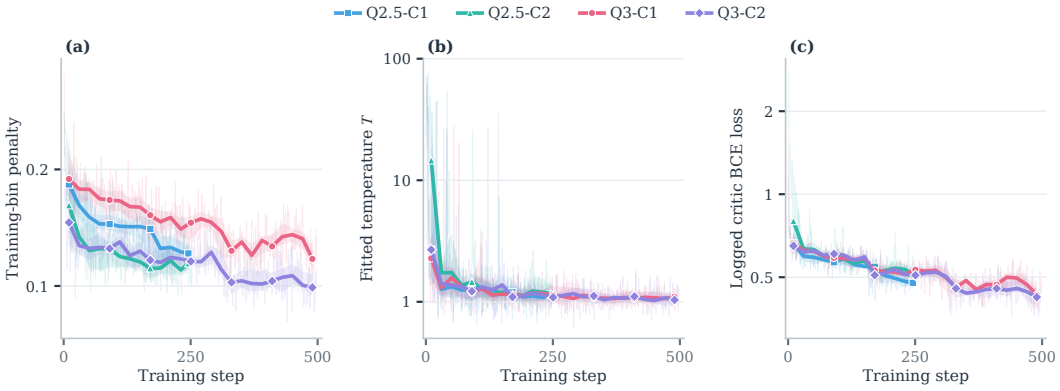


Figure 7: Critic calibration across four CBPO configurations: (a) squared training-bin penalty, (b) fitted temperature, and (c) critic BCE with the training implementation's micro-batch scaling. Lines and bands show medians and interquartile ranges within 20-step windows for one run per configuration. Qwen2.5 and Qwen3 training runs span 250 and 500 steps, respectively; panels (b,c) use logarithmic scales.

**Advantage and gradient dynamics.** Figure 8 compares minimum advantages and actor gradient norms before clipping. Qwen2.5 C1/C2 contain three/one steps with advantages below $-100$, and Qwen3 C1/C2 contain one/zero. The five extreme minima are approximately $-9{,}986$. Qwen3 C2 avoids these extreme events in the 500-step training run and maintains gradient norms close to one, providing additional support for its favorable training behavior among the evaluated settings.

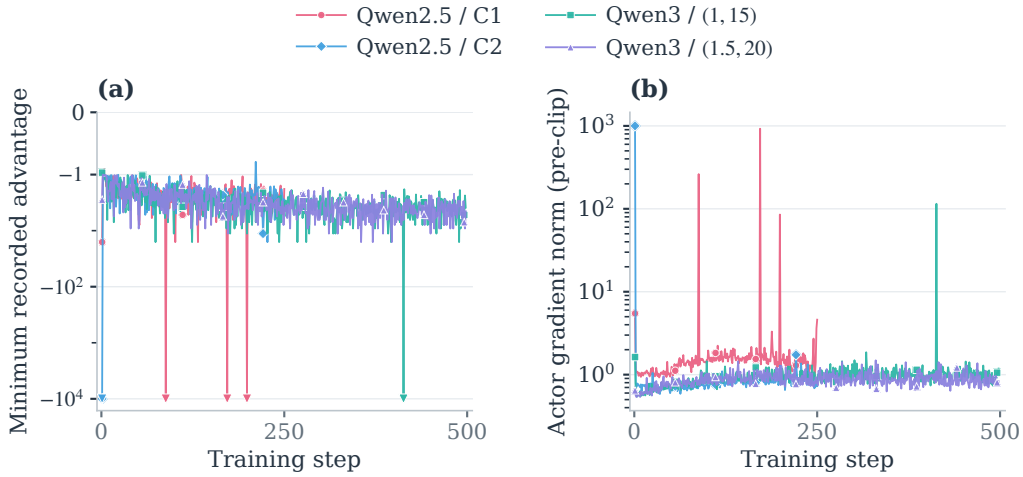


Figure 8: Minimum advantages and actor gradient norms before clipping for the four CBPO configurations. All observations are shown without smoothing. The advantage and gradient panels use symmetric-log and logarithmic scales, respectively.

### F.1 Calibration Overhead

Figure 9 compares calibration cost for Qwen3 C1/C2. The ECE penalty and temperature refitting together take 115.8/118.6 ms per step, respectively. For the 500-step runs, their recorded cumulative costs account for 0.0824%/0.0828% of training time. Both configurations thus support repeated probability calibration at a small incremental cost while reusing the existing rollout batch.

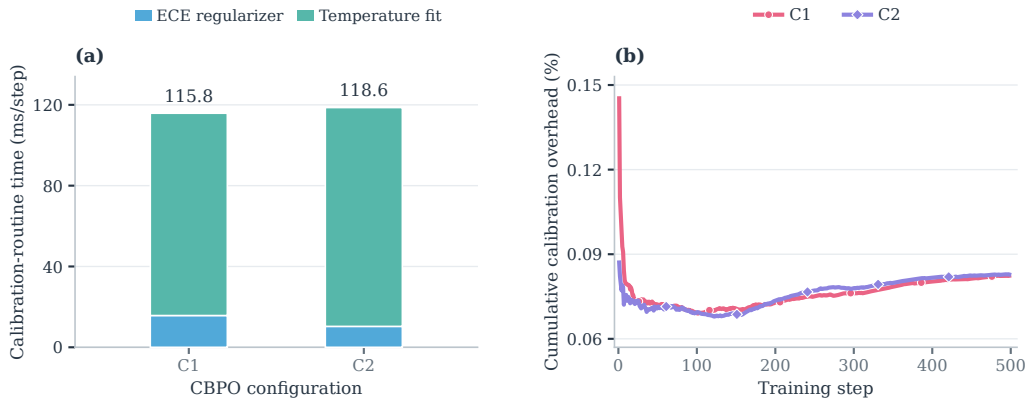


Figure 9: Calibration overhead for Qwen3 C1/C2: (a) mean time per step for the ECE penalty and temperature refitting; (b) cumulative calibration time as a percentage of cumulative training-step time during the 500-step runs.

# G THEORETICAL DERIVATIONS

We analyze binary outcome rewards under a fixed rollout policy. For a prompt $q$, let $p(q) = \Pr(r = 1 \mid q)$ and $r \mid q \sim \text{Bernoulli}(p(q))$. Unless stated otherwise, $0 < p(q) < 1$, the critic is fixed before the analyzed response is sampled, and probabilities and advantages are treated as constants in the actor update. The results characterize reward residuals; they do not establish convergence or stability of the full clipped policy optimizer.

## G.1 GRPO CONVERGES TO BERNOULLI VARIANCE NORMALIZATION

Let $r_1, r_2, \ldots$ be independent Bernoulli$(p)$ outcomes for a fixed prompt. For $G \geq 2$, define the empirical mean and unbiased sample variance by

$$\mu_G = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma_G^2 = \frac{1}{G-1}\sum_{j=1}^{G}(r_j - \mu_G)^2. \tag{17}$$

Since $r_j^2 = r_j$, expansion of the sum gives

$$\sum_{j=1}^{G}(r_j - \mu_G)^2 = G\mu_G - 2G\mu_G^2 + G\mu_G^2 = G\mu_G(1-\mu_G), \qquad \sigma_G^2 = \frac{G}{G-1}\mu_G(1-\mu_G). \tag{18}$$

By the strong law of large numbers, $\mu_G \to p$ almost surely and therefore $\sigma_G \to \sqrt{p(1-p)} > 0$. For any fixed sample index $i$,

$$\frac{r_i - \mu_G}{\sigma_G} \xrightarrow[G\to\infty]{\text{a.s.}} A^*(r_i, p) := \frac{r_i - p}{\sqrt{p(1-p)}}. \tag{19}$$

At finite $G$, groups with identical rewards have zero empirical variance and require a numerical convention or stabilizer. For interior $p$, the denominator is almost surely positive for all sufficiently large $G$. Using divisor $G$ instead of $G-1$ changes the finite-sample correction, but not this limit.

This limit identifies the population normalization targeted by the group estimator. It does not make a single realized CBPO update identical to a finite-group GRPO update: the latter still averages multiple sampled policy scores, while CBPO uses a learned probability estimate and one sampled response.

## G.2 UNIQUENESS OF THE STANDARDIZED AFFINE BERNOULLI RESIDUAL

Consider an affine residual $A = a(p)r + b(p)$, with $a(p) > 0$ so that a successful outcome receives the larger advantage. Conditional centering implies

$$0 = \mathbb{E}[A \mid p] = a(p)p + b(p), \qquad b(p) = -a(p)p. \tag{20}$$

Unit conditional variance further requires

$$1 = \text{Var}(A \mid p) = a(p)^2 p(1-p). \tag{21}$$

Consequently,

$$a(p) = \frac{1}{\sqrt{p(1-p)}}, \qquad b(p) = -\frac{p}{\sqrt{p(1-p)}}, \qquad A = A^*(r, p). \tag{22}$$

Without the positive-orientation condition, the two solutions are $A^*$ and $-A^*$. The result is a uniqueness statement within this affine, centered, unit-variance class. It does not establish optimal policy-gradient variance, maximal effective sample size, or optimal sample complexity. At $p = 0$ or 1, the reward is deterministic and a centered affine function of it cannot have unit variance.

## G.3 RAW RESIDUAL ADVANTAGES CREATE DIFFICULTY-DEPENDENT SIGNAL SCALE

With an exact promptwise probability estimate, the unnormalized residual is $A_{\text{raw}} = r - p$. Its conditional moments are

$$\mathbb{E}[A_{\text{raw}} \mid q] = 0, \qquad \text{Var}(A_{\text{raw}} \mid q) = p(q)(1-p(q)). \tag{23}$$

The variance is largest at $p = 1/2$ and approaches zero at the boundaries. For a hard prompt with $p \ll 1$, a failure yields $A_{\text{raw}} = -p$ with probability $1 - p$, whereas a rare success yields $1 - p$ with probability $p$. Thus the residual has a difficulty-dependent scale, and informative successes can be scarce in a finite rollout budget. Normalization standardizes this residual scale under exact probabilities; it does not increase the probability of sampling a success. Nor does scalar residual variance alone determine the magnitude or signal-to-noise ratio of the policy gradient, which also depends on $\nabla_\theta \log \pi_\theta(o \mid q)$.

### G.4 Miscalibration Causes Spurious Variance Amplification

For a fixed prompt, write $s = \hat{V}(q)$ and $D(s) = s(1-s)+\epsilon$, where $\epsilon > 0$ is the numerical stabilizer. The practical residual is

$$\hat{A} = \frac{r - s}{\sqrt{D(s)}}. \tag{24}$$

Because $s$ is fixed conditional on $q$,

$$\mathbb{E}[\hat{A} \mid q] = \frac{p(q) - s}{\sqrt{D(s)}}, \qquad \mathrm{Var}(\hat{A} \mid q) = \frac{p(q)(1 - p(q))}{D(s)}. \tag{25}$$

Equivalently, defining $e(q) = s - p(q)$ gives a conditional centering error $-e(q)/\sqrt{D(s)}$. A near-boundary critic prediction can therefore amplify both centering error and variance when the true probability remains interior. For fixed positive $\epsilon$, the variance is finite and at most $1/(4\epsilon)$; divergence as $s \to 0$ or 1 refers to the unstabilized $\epsilon = 0$ expression. A small stabilizer alone can still permit large advantages and is not a guarantee of stable optimization.

The critic consequently controls both the numerator and the scale. The term *centering error* here describes the scalar residual relative to its ideal zero mean. It should not be read as a proof of policy-gradient bias: an action-independent baseline still has zero expected score-function contribution in an unclipped on-policy estimator. Prompt-dependent normalization changes the weighting of prompt contributions, and clipping introduces additional distinctions from that estimator.

### G.5 Approximate Score Calibration Bounds

To connect the analysis to calibration diagnostics, now condition on a predicted score rather than on an individual prompt. Let $p_s = \Pr(r = 1 \mid \hat{V} = s)$. Since the reward is binary, its conditional distribution at score $s$ is Bernoulli$(p_s)$, even when multiple prompts share that score. Hence

$$\mathbb{E}[\hat{A} \mid s] = \frac{p_s - s}{\sqrt{D(s)}}, \qquad \mathrm{Var}(\hat{A} \mid s) = \frac{p_s(1 - p_s)}{D(s)}. \tag{26}$$

Subtracting one from the variance and factoring the numerator gives

$$\mathrm{Var}(\hat{A} \mid s) - 1 = \frac{p_s(1 - p_s) - s(1 - s) - \epsilon}{D(s)} = \frac{(p_s - s)(1 - p_s - s) - \epsilon}{D(s)}. \tag{27}$$

Suppose $|p_s - s| \le \delta$ for the scores under consideration. Because $p_s, s \in [0, 1]$, we have $|1 - p_s - s| \le 1$. Applying this inequality and the triangle inequality to the preceding expressions yields

$$\left|\mathbb{E}[\hat{A} \mid s]\right| \le \frac{\delta}{\sqrt{D(s)}}, \qquad \left|\mathrm{Var}(\hat{A} \mid s) - 1\right| \le \frac{\delta + \epsilon}{D(s)}. \tag{28}$$

If, additionally, $s \in [\eta, 1 - \eta]$ for $0 < \eta \le 1/2$, then $D(s) \ge \eta(1 - \eta) + \epsilon$, giving the uniform bounds

$$\left|\mathbb{E}[\hat{A} \mid s]\right| \le \frac{\delta}{\sqrt{\eta(1 - \eta) + \epsilon}}, \qquad \left|\mathrm{Var}(\hat{A} \mid s) - 1\right| \le \frac{\delta + \epsilon}{\eta(1 - \eta) + \epsilon}. \tag{29}$$

These bounds are conditional on uniform control of calibration error and the stated score range. ECE regularization and temperature fitting do not by themselves establish those assumptions throughout online policy drift. Moreover, score calibration $p_s = s$ is weaker than exact promptwise estimation $\hat{V}(q) = p(q)$. For example, a constant score of $1/2$ can be calibrated over an equally weighted mixture of prompts with success probabilities $0.1$ and $0.9$. Thus good binned reliability does not alone prove that the ideal GRPO-limit residual is recovered for every prompt.

### G.6 Calibrated Bernoulli Normalization Gives Unit Conditional Variance

Under the stronger idealization of exact promptwise probability estimation, $s = p(q)$, the moments in Equation 25 become

$$\mathbb{E}[\hat{A} \mid q] = 0, \qquad \mathrm{Var}(\hat{A} \mid q) = \frac{p(q)(1-p(q))}{p(q)(1-p(q)) + \epsilon} \leq 1. \tag{30}$$

With $\epsilon = 0$ and $0 < p(q) < 1$, this is exactly one, recovering the unique standardized residual. With positive $\epsilon$, the variance is below one and tends to zero as the true probability approaches either deterministic boundary. Under score calibration alone, the corresponding statement holds conditional on $s$, with $p_s = s$, rather than necessarily for each prompt.

### G.7 Connection to the Fisher-Standardized Bernoulli Score

For $r \sim \mathrm{Bernoulli}(p)$, the log likelihood and its score are

$$\ell(p; r) = r \log p + (1-r)\log(1-p), \qquad \frac{\partial \ell}{\partial p} = \frac{r-p}{p(1-p)}. \tag{31}$$

The score has zero mean, and its second moment gives the Fisher information:

$$\mathcal{I}(p) = \mathbb{E}\left[\left(\frac{\partial \ell}{\partial p}\right)^2\right] = \frac{\mathbb{E}[(r-p)^2]}{p^2(1-p)^2} = \frac{1}{p(1-p)}. \tag{32}$$

Therefore the ideal residual has the equivalent forms

$$A^*(r, p) = \frac{r-p}{\sqrt{p(1-p)}} = (r-p)\sqrt{\mathcal{I}(p)} = \mathcal{I}(p)^{-1/2}\frac{\partial \ell(p; r)}{\partial p}. \tag{33}$$

It is thus a Fisher-standardized score in the one-dimensional Bernoulli reward coordinate. Fisher information also underlies natural-gradient methods (Amari, 1998; Kakade, 2001), but Equation 33 does not identify CBPO with an inverse-Fisher update in the full policy parameter space. In particular, it does not imply equal policy-gradient magnitudes across prompts or maximal effective sample size.

### G.8 ECE Regularization and Critic Calibration

A predictor is score calibrated when $\Pr(r = 1 \mid \hat{V} = s) = s$ almost surely over its score distribution (Naeini et al., 2015; Guo et al., 2017). For a batch of $N$ outcomes, partition the unscaled critic probabilities $\hat{p}_i = \sigma(z_i)$ into $M$ equal-width bins. For each nonempty bin,

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|}\sum_{i \in B_m} r_i, \qquad \mathrm{conf}(B_m) = \frac{1}{|B_m|}\sum_{i \in B_m} \hat{p}_i. \tag{34}$$

The squared ECE-style penalty and complete critic loss are

$$\mathcal{L}_{\mathrm{ECE}} = \sum_{m:|B_m|>0} \frac{|B_m|}{N}\left(\mathrm{sg}(\mathrm{acc}(B_m)) - \mathrm{conf}(B_m)\right)^2, \qquad \mathcal{L}_{\mathrm{critic}} = \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{ECE}}\mathcal{L}_{\mathrm{ECE}}. \tag{35}$$

Empty bins contribute zero. For fixed bin assignments, the penalty provides gradients through mean confidence. It encourages bin-level frequency matching and is distinct from the usual absolute-gap ECE diagnostic.

BCE is itself a proper scoring rule: its population minimum over unrestricted predictions occurs at the true conditional probabilities. Finite samples, restricted models, imperfect optimization, and a changing rollout policy can nevertheless leave calibration error. The additional penalty explicitly encourages reliability at the bin level. Neither zero empirical binned error nor minimizing this regularizer implies exact promptwise probabilities or a uniform bound on the score calibration error in Equation 29.

### G.9 Temperature Scaling Is Rank Preserving and Adjusts Global Logit Scale

For fixed logits $z = f_\phi(q)$ and $T > 0$, define $\hat{V}_T(q) = \sigma(z/T)$. If $z_1 > z_2$, then $z_1/T > z_2/T$, and the strict monotonicity of the sigmoid implies $\sigma(z_1/T) > \sigma(z_2/T)$. A positive temperature therefore preserves ranking (Guo et al., 2017).

After a critic training phase, CBPO fits the temperature using binary negative log-likelihood on the collected rollout outcomes:

$$T^* \in \arg\min_{T>0} \mathcal{L}_{\mathrm{NLL}}(T), \qquad \mathcal{L}_{\mathrm{NLL}}(T) = -\frac{1}{N}\sum_{i=1}^{N} \left[ r_i \log V_i(T) + (1 - r_i) \log(1 - V_i(T)) \right], \tag{36}$$

where $V_i(T) = \sigma(z_i/T)$. L-BFGS provides a lightweight numerical fit. For fixed logits, inverse temperature $\tau = 1/T$ yields

$$\mathcal{L}_{\mathrm{NLL}}(\tau) = \frac{1}{N}\sum_i \big[\log(1 + e^{\tau z_i}) - r_i \tau z_i\big], \qquad \frac{d^2\mathcal{L}_{\mathrm{NLL}}}{d\tau^2} = \frac{1}{N}\sum_i z_i^2 \sigma(\tau z_i)(1 - \sigma(\tau z_i)) \geq 0. \tag{37}$$

This convexity is in inverse temperature for fixed logits; it is not a general convexity or finite global-optimum guarantee for optimization in $T$.

Conditioning on fixed logits and true probabilities $p_i$, the expected loss decomposes as

$$\mathbb{E}[\mathcal{L}_{\mathrm{NLL}}(T)] = \frac{1}{N}\sum_i \left[ H(p_i) + D_{\mathrm{KL}}(\mathrm{Bern}(p_i) \,\|\, \mathrm{Bern}(V_i(T))) \right]. \tag{38}$$

The entropy term $H(p_i)$ is independent of $T$. Minimizing expected NLL therefore minimizes the average KL divergence within the restricted one-temperature family. That family need not contain the true probabilities, and empirical fitting need not achieve perfect calibration or improve every held-out calibration metric. Temperatures above one soften logits toward $1/2$, while temperatures below one sharpen them. $T = 1$ simply leaves the logits unchanged; it is not evidence of calibration.

In CBPO, ECE regularization is applied during critic updates and temperature is repeatedly refitted on recent current-policy rollouts. The advantages already computed for the current actor update remain fixed; the updated critic and temperature are used in subsequent advantage estimation. This repeated adaptation can respond to policy drift, but does not remove the need to measure calibration over training.

### G.10 Theoretical Extensions Beyond Binary Rewards

The experiments and implemented CBPO method concern binary verifiable rewards. The following moment identities describe possible extensions, not evaluated variants. For any scalar reward $R \mid q$ with finite conditional mean $\mu(q)$ and variance $v(q)$, define

$$A_{\mathrm{gen}} = \frac{R - \mu(q)}{\sqrt{v(q) + \epsilon}}. \tag{39}$$

Under exact moments,

$$\mathbb{E}[A_{\mathrm{gen}} \mid q] = 0, \qquad \mathrm{Var}(A_{\mathrm{gen}} \mid q) = \frac{v(q)}{v(q) + \epsilon}. \tag{40}$$

The Bernoulli case is particularly economical because its mean determines its variance. General rewards require reliable estimation of both quantities.

**Finite-valued rewards.** For $R = c_Y$, where $Y$ has categorical probabilities $\mathbf{p}$, let $\mathbf{c} = (c_1, \ldots, c_K)^\top$. Then

$$\mu = \sum_k p_k c_k, \qquad v = \sum_k p_k (c_k - \mu)^2 = \mathbf{c}^\top F \mathbf{c}, \qquad F = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top. \tag{41}$$

Here $F$ is the categorical Fisher matrix in logit coordinates. Since the logit score is $\mathbf{e}_Y - \mathbf{p}$, $R - \mu = \mathbf{c}^\top(\mathbf{e}_Y - \mathbf{p})$. With $v > 0$ and $\epsilon = 0$, $A_{\mathrm{gen}}$ is the standardized projection of this score along the reward direction. The Bernoulli construction is the special case $K = 2$ with reward values zero and one.

**Regular one-parameter natural exponential families.** Suppose $f(R;\alpha) = h(R)\exp\{\alpha R - \psi(\alpha)\}$ with regular support, finite positive variance, and sufficient statistic $R$. Then $\mu = \psi'(\alpha)$ and $v = \psi''(\alpha)$, so

$$\frac{\partial \log f}{\partial \mu} = \frac{R-\mu}{v}, \qquad \mathcal{I}(\mu) = \frac{1}{v}, \qquad A_{\text{gen}} = (R-\mu)\sqrt{\mathcal{I}(\mu)} \quad (\epsilon = 0). \tag{42}$$

This extends the standardized-score interpretation under the stated family assumptions. An empirical extension would require an appropriate distributional critic and validation of its conditional moment estimates. Continuous, partial-credit, and preference-based rewards remain outside the current experimental claims.

### G.11 Summary of Theoretical Roles

The GRPO limit identifies a Bernoulli standardization target, and the affine uniqueness argument specifies it once centering, unit variance, and positive orientation are required. The miscalibration identities explain why replacing group statistics by a critic affects both centering and scale. Approximate score calibration yields bounded distortions under explicit error and boundary conditions. ECE regularization and repeated temperature fitting are practical ways to encourage calibration; their empirical effectiveness is distinct from these conditional statements. The Fisher connection characterizes the reward residual rather than the full policy optimizer.

## H Artifact Licenses and Terms of Use

All models, datasets, and software packages used in this work are publicly available research artifacts. We follow their respective licenses and terms of use. We do not redistribute the original datasets or model checkpoints. Any released code or metadata associated with this work is intended solely for research purposes and should be used in accordance with the licenses and terms of the underlying artifacts.

## I Limitations

The implemented and evaluated method uses binary outcome rewards. Coding with partial credit, continuous tool-use scores, and preference-based rewards are outside the validated scope. Appendix G.10 gives moment-normalization and categorical/exponential-family interpretations as theoretical extensions, not implemented results. Calibration errors can be amplified near probability boundaries; low aggregate ECE does not certify a uniform error bound. Archived CBPO runs contain occasional extreme negative advantages, so empirical calibration should not be treated as a guarantee of bounded training tails (Appendix F). Three-seed results cover only Qwen3-8B without thinking and have overlapping confidence intervals. CBPO does not localize faulty reasoning steps, and the current outcome-only comparisons do not establish superiority over process-supervised methods. The reported efficiency measurements are specific to the measured training setup.