

# 000 001 002 003 004 005 006 007 008 009 010 011 DREAM: DECOUPLED REINFORCEMENT LEARNING 012 WITH REWARD MEASUREMENT FOR LARGE LAN- 013 GUAGE MODEL TEST-TIME TRAINING 014 015 016 017 018 019 020 021 022 023 024 025 026 027

028 **Anonymous authors**  
029  
030 Paper under double-blind review  
031

## 032 ABSTRACT 033

034 This paper studies the problem of large language model (LLM) test-time training,  
035 which aims to enhance the reasoning ability of LLMs via unlabeled test data. Re-  
036 cent works usually utilize majority voting to infer the labels of samples to guide  
037 the reinforcement learning process, which could be inaccurate and biased with  
038 potential error accumulation. Towards this end, we propose a novel approach  
039 named Decoupled Reinforcement Learning with RMeasurement (DREAM)  
040 for LLM test-time training. The core of our proposed DREAM is to decouple the  
041 reward estimation from reinforcement learning with enhanced calibration. In par-  
042 ticular, our DREAM trains an LLM-based calibration model which takes both  
043 questions and answers as input, and outputs the calibration scores. To mitigate  
044 overconfident results, the judge model is trained by simulating on an independent  
045 reference dataset with positive and negative pairs. The reference-based calibra-  
046 tion scores would be incorporated into voting-based reward estimation to reduce  
047 the potential biases, which enhance reliable test-time training. Extensive experi-  
048 ments on benchmark datasets validate the superiority of the proposed DREAM in  
049 comparison with competing baselines.  
050

## 051 1 INTRODUCTION 052

053 Large language models (LLMs) (Zhao et al., 2024; Chang et al., 2024; Kasneci et al., 2023; Naveed et al., 2025) have become the backbone of modern natural language processing (NLP) (Khurana et al., 2023; Feuerriegel et al., 2025; Frenda et al., 2025), powering a wide range of downstream applications (Thirunavukarasu et al., 2023; Kaddour et al., 2023; Nie et al., 2024; Guo et al., 2024; Yan et al., 2025). Beyond their scale and architecture, the effectiveness of LLMs crucially depends on their training paradigm (Minaee et al., 2024; Wang et al., 2025; Dong et al., 2025), which can be broadly divided into *pre-training* (Ma et al., 2024) and *post-training* (Zosa et al., 2025; Kumar et al., 2025). While pre-training builds general-purpose capabilities from large-scale corpora, post-training emphasizes aligning LLMs with human preferences and adapting them to task-specific requirements, making it a critical stage for improving reasonability and reliability.

054 Among post-training paradigms for large language models (Lai et al., 2025; Lv et al., 2025; Tie et al., 2025), test-time reinforcement learning (Odena et al., 2017; Qu et al., 2025) has emerged as a promising strategy to enhance reasoning and generalization on complex tasks (Ge et al., 2023; Han et al., 2024) and unseen knowledge (Farquhar et al., 2023; Li et al., 2025). Unlike conventional reinforcement learning that requires labeled data, TTRL (Zuo et al., 2025) leverages unlabeled test inputs and majority voting based rewards to refine policies online without explicit supervision. Early approaches exploit consistency-based signals, such as self-consistency across multiple reasoning paths, to generate pseudo-rewards for stable adaptation (Wang et al., 2022; Ghosh et al., 2024), while subsequent work integrates calibration and uncertainty modeling to mitigate overconfident or biased feedback (Stangel et al., 2025; Park et al., 2025). Extensions of TTRL combine it with test-time training (TTT) (Sun et al., 2020; Osowiechi et al., 2024), introducing auxiliary self-supervised objectives to improve robustness under distribution shifts (Zhang et al., 2025d), and task-specific designs, such as leveraging region consistency in GUI grounding (Du et al., 2025), further demonstrate its adaptability. Collectively, these advances establish test-time reinforcement learning as a

flexible and rapidly evolving framework that unifies self-supervision, calibrated reward estimation, and reinforcement learning to enhance LLMs’ reasoning, robustness, and task adaptability.

Despite the advantages of test-time reinforcement learning, several challenges remain when handling unlabeled data. For instance, TTTRL (Zuo et al., 2025) derives rewards by aggregating multiple outputs via majority voting. However, such voting may yield inaccurate rewards that deviate from the ground-truth labels, thereby introducing negative feedback into the subsequent reinforcement learning process. These observations raise a critical question: **① How can reinforcement learning be enhanced with reliable rewards in the absence of explicit supervision?** More importantly, reward generation and policy updates are inherently coupled within the reinforcement learning paradigm. Consequently, inaccurate rewards can lead to biased policy optimization, which in turn compounds generation errors and creates a vicious cycle during the post-training process. This concern motivates a second key question: **② Can reward estimation be decoupled from policy optimization to mitigate bias accumulation in post-training?**

To address these challenges, we propose a simple yet effective framework DREAM for enhancing adaptation on unlabeled test data by decoupling reward estimation from reinforcement learning in LLM test-time training. Specifically, DREAM constructs an independent labeled reference dataset to train an auxiliary judge model, which is employed to estimate the reliability of model outputs. Built upon this, we design a hybrid reward calibration module that incorporates reliability scores predicted by the frozen judge model to produce final rewards. Subsequently, these calibrated rewards are leveraged to perform unbiased test-time reinforcement learning, thereby promoting generalizability when handling diverse inputs in unsupervised scenarios. To demonstrate the reliability and effectiveness of DREAM, we further conduct extensive experiments on benchmark datasets, showing outstanding advantages over state-of-the-art baseline approaches.

To better elucidate the contributions of DREAM, we summarize them as follows:

- ① **Novel Perspective.** To the best of our knowledge, we are the first to decouple reward estimation from reinforcement learning for unlabeled test data during the test-time training phase.
- ② **Unified Framework.** To enhance reinforcement learning without labeled data, we introduce an auxiliary module trained on independent reference data to estimate output reliability, which is subsequently leveraged to calibrate the final reward for policy optimization.
- ③ **Empirical Validation.** We demonstrate the effectiveness of DREAM through extensive experiments on benchmark datasets against several baselines, showing its superiority empirically.

## 2 RELATED WORK

### 2.1 LLM POST-TRAINING

While *pre-training* provides a general foundation by learning from large-scale corpora, *post-training* aligns LLMs with specific knowledge, enhancing reasoning capabilities and refining outputs through target feedback (Kumar et al., 2025). Furthermore, classic paradigms for *post-training* include: (i) **Test-time scaling (TTS)**, which improves inference performance by allocating additional computational resources during reasoning (e.g. sequential scaling (Madaan et al., 2023; Brinkmann & Bizer, 2025) and parallel sequential (Korikov et al., 2025; Zhou et al., 2025)); (ii) **Fine-tuning**, which adapts the pretrained model to specific downstream domains to enhance task alignment Zhang et al. (2024); and (iii) **Reinforcement learning (RL)**, which optimizes models using reward signals derived from human preferences or other evaluative feedback (Laleh & Ahmadabadi, 2024; Yu et al., 2025; Hu et al., 2025). **Test-time scaling** improves inference performance without updating model parameters but incurs higher computational cost and may generalize poorly (Zhang et al., 2025c). **Fine-tuning** (Pletenev et al., 2025) enables strong task specialization but risks overfitting and requires high-quality labeled data. In contrast, **reinforcement learning** provides a flexible framework for post-training optimization, leveraging reward signals to enhance task alignment even under limited supervision. This flexibility motivates our focus on RL in the work.

### 2.2 REINFORCEMENT LEARNING IN LLMs

Reinforcement learning (RL) (Chen et al., 2025b; Dang & Ngo, 2025) has become a mainstream post-training paradigm for aligning large language models (LLMs) with human preferences (Zhang et al., 2025a; Yu et al., 2025; Hariharan, 2025), thereby enhancing their reasoning ability and adap-

108  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117  
 118  
 119  
 120  
 121  
 122  
 123  
 124  
 125  
 126  
 127  
 128  
 129  
 130  
 131  
 132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152  
 153  
 154  
 155  
 156  
 157  
 158  
 159  
 160  
 161

tive generalization during inference. For instance, Proximal Policy Optimization (PPO) (Schulman et al., 2017) performs policy updates through multiple epochs of stochastic gradient ascent and serves as a cornerstone of Reinforcement Learning with Human Feedback (RLHF) (Bai et al., 2022; Chaudhari et al., 2024). Beyond on-policy methods (Mroueh et al., 2025), several off-policy approaches (e.g., Implicit Language Q-Learning (ILQL) (Snell et al., 2023) and VerifierQ (Qi et al., 2024)) exploit offline datasets or auxiliary verifier signals to improve efficiency and reasoning reliability. More importantly, DeepSeek-R1 (Guo et al., 2025) introduces Group Relative Policy Optimization (GRPO) (Zhang et al., 2025b; Sane, 2025), which extends LLMs’ reasoning capabilities through outcome-based relative rewards. Despite these advances, most RL-based post-training (Sun & van der Schaar, 2025) frameworks rely on reliable supervised signals from humans or AI models, limiting their versatility when such labels are unavailable (Wang et al., 2024; Chen et al., 2025a;b). To address this challenge, Test-time Reinforcement Learning (TTRL) (Zuo et al., 2025) employs majority voting to derive consistency-based rewards, enabling reinforcement learning directly on unlabeled test data. However, coupling output generation and reward evaluation often leads to over-confident predictions and biased reinforcement learning. This motivates the need to decouple reward estimation from policy optimization for more robust test-time training.

### 3 THE PROPOSED DREAM

**Problem Definition.** In this work, we address the problem of test-time training for large language models (LLMs) using unlabeled test data. Formally, let  $\mathcal{D}_U = \{x_i\}_{i=1}^{N_U}$  denote the set of test-time inputs without ground-truth labels, where  $x_i \in \mathcal{X}$  belongs to the input space. Let  $\pi_\theta(\cdot | x)$  represents the base policy model, which could generate candidate outputs  $\{y_i\}$  according to  $y_i \sim \pi_\theta(y | x)$  for a given input  $x$ . Since the true labels are unavailable, our objective is to design a reliable reward estimation module  $R(\cdot)$  and to leverage the resulting calibration scores to optimize the policy model  $\pi_\theta$  within a reinforcement learning framework.

#### 3.1 DREAM: DECOUPLING REWARD ESTIMATION AND REINFORCEMENT LEARNING

To provide reliable rewards for test-time training with unlabeled data, our DREAM framework decouples reward estimation from reinforcement learning to generate enhanced calibration scores. Specifically, DREAM first generates multiple outputs from the policy model via repeated sampling, which are then employed to construct rewards for the unlabeled data through majority voting. To mitigate overly confident reward signals, we additionally train a judge model using a simulated independent reference dataset with supervised labels, which identifies whether the generated outputs are reliable. The calibration scores produced by the judge model are subsequently leveraged to further refine the rewards, thereby enabling more reliable test-time training. A detailed overview of the DREAM pipeline is presented in Figure 1.

#### 3.2 CALIBRATION MODEL OPTIMIZATION WITH REFERENCE DATA

For unlabeled data, directly incorporating rewards into reinforcement learning (Zuo et al., 2025) may lead to biased and overconfident outcome during test-time training. To address this challenge, we introduce an additional model to calibrate the rewards, thereby providing more reliable evaluations based on an established reference dataset. More specifically, we construct a reference dataset  $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{N_L}$ , where  $x_i$  denotes a question and  $y_i$  represents its corresponding answer. This dataset is then leveraged to independently train a judge model. For each input  $x_i$ , we sample  $K$  candidate outputs from the policy model  $\pi_\theta(\cdot | x_i)$ , forming a candidate set denotes as:

$$\mathcal{O}_i = \{\hat{y}_{i,k} \mid \hat{y}_{i,k} \stackrel{\text{i.i.d.}}{\sim} \pi_\theta(\cdot | x_i), 1 \leq k \leq K\}, \quad (1)$$

where  $\hat{y}_{i,k}$  are independently and identically distributed (i.i.d.) samples draw from the policy model  $\pi(\cdot | x_i)$ . These outputs are then passed through a canonical answer extraction function  $C$ , which generates binary labels indicating whether each sampled output matches the ground-truth answer:

$$\mathcal{S}_i = \{s_{i,k}\}_{k=1}^K = \{\mathbb{I}(C(\hat{y}_{i,k}) = y_i) \mid \hat{y}_{i,k} \in \mathcal{O}_i\}, \quad (2)$$

where  $s_{i,k}$  is the binary label representing the consistency with the ground-truth label, and  $\mathbb{I}$  is the indicator function. Built upon these formulation, the judge model  $F_\phi$  is trained to learn a mapping

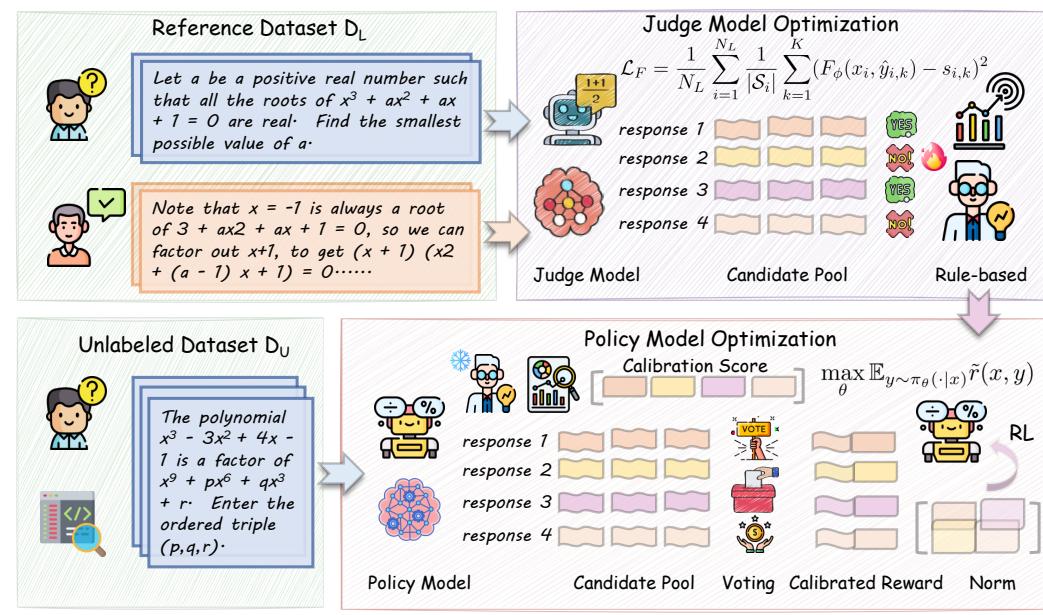


Figure 1: Overview of DREAM. The framework is composed of two optimization modules: judge model optimization and policy model optimization.

from  $(x_i, \mathcal{O}_i)$  to  $\mathcal{S}_i$ , i.e.,  $F_\phi : (x_i, \hat{y}_{i,k}) \rightarrow s_{i,k}$ . The parameters of  $F_\phi$  are then optimized with a mean squared error (MSE) regression loss:

$$\mathcal{L}_F = \frac{1}{N_L} \sum_{i=1}^{N_L} \frac{1}{|\mathcal{S}_i|} \sum_{k=1}^K (F_\phi(x_i, \hat{y}_{i,k}) - s_{i,k})^2, \quad (3)$$

where  $F_\phi(x_i, \hat{y}_{i,k})$  denotes the calibration score predicted by the judge model, and  $s_{i,k}$  is the binary alignment score. By incorporating these additional signals, the judge model gains the ability to assess the reliability of generated outputs, thereby serving as an auxiliary module to refine reward estimation for unlabeled data during test-time training.

### 3.3 REWARD ENHANCEMENT THROUGH CALIBRATION

Generally, the policy model  $\pi_\theta$  is optimized using a supervised reward function to guide reinforcement learning. However, in the absence of supervisory signals for unlabeled data, direct reward-based optimization becomes unreliable and overconfident. To tackle this, we propose a hybrid reward enhancement module that incorporates calibration via a previously trained judge model, enabling more effective utilization of rewards under unsupervised conditions.

**Voting-Based Consistency Reward.** To generate rewards that enhance reinforcement learning, we first construct a candidate set by repeatedly sampling from the policy model  $\pi_\theta$ . For the unlabeled dataset  $\mathcal{D}_U = \{x_i\}_{i=1}^{N_U}$ , an answer set  $\mathcal{O}_i = \{\hat{y}_{i,k}\}_{k=1}^K$  is obtained for each input  $x_i$  following Eq. 1. Subsequently, we apply majority voting (Zhou et al., 2024; Zhao et al., 2025) to evaluate the consistency within the answer set, formulated as:

$$r_{i,k}^v = \mathbb{I}(\hat{y}_{i,k} = \tilde{y}_i), \quad \text{where } \tilde{y}_i = \text{MV}(\mathcal{O}_i) \quad \text{and} \quad 1 \leq k \leq K. \quad (4)$$

Here  $K$  denotes the number of sampled outputs, and  $\text{MV}(\mathcal{O}_i)$  represents the consensus output determined by the majority voting aggregation strategy. The indicator  $r_{i,k}^v$  thus defines a consistency reward for each output in the candidate set  $\mathcal{O}_i$ . This approach allows us to derive intrinsic rewards in the absence of supervised signals, providing guidance for the optimization of the policy model.

**Calibration for Reward Reliability.** Recalling that we have trained a judge model  $F_\phi$  to assess the confidence of each output individually. By integrating this confidence assessment component, we can decouple pseudo-label generation from quality evaluation on the unlabeled dataset. This

paradigm provides more reliable rewards to enhance the policy model through reinforcement learning during test-time training. Accordingly, the final reward for each output is obtained by combining the voting-based reward with the calibration score from the judge model:

$$r_{i,k} = r_{i,k}^v + s_{i,k}, \quad \text{where } 1 \leq i \leq N_U, \quad 1 \leq k \leq K. \quad (5)$$

This approach decouples pseudo-labels from confidence estimation, promoting correct answers while penalizing incorrect ones, thereby mitigating overconfidence in erroneous outputs.

### 3.4 UNBIASED TEST-TIME REINFORCEMENT LEARNING

To enable unbiased test-time reinforcement learning, DREAM incorporates rewards that combine voting-based consistency with calibration scores for enhanced reliability.

**Group-wise Reward Normalization.** Group Relative Policy Optimization (GRPO) (Shao et al., 2024) optimizes policies by comparing the relative strengths of different responses within a group, which reduces both memory usage and computational overhead. Motivated by this, we normalize rewards using a candidate set  $\mathcal{O}_i$ . Specifically, given an output  $\hat{y}_{i,k} \in \mathcal{O}_i$  with its associated reward  $r_{i,k}$ , the normalized score is computed as:

$$\tilde{r}_{i,k} = \frac{r_{i,k} - \mu_i}{\sigma_i}, \quad \text{where } \mu_i = \frac{1}{K} \sum_{k=1}^K r_{i,k} \quad \text{and} \quad \sigma_i = \sqrt{\frac{1}{K} \sum_{k=1}^K (r_{i,k} - \mu_i)^2}. \quad (6)$$

This reward normalization provides a more stable and comparable signal, which can effectively guide the optimization of the policy model.

**Policy Model Optimization.** Here, we leverage the previously generated rewards to optimize the policy model using unlabeled data during the test-time training process. The reinforcement learning objective is to maximize the expected reward, formulated as:

$$\max_{\theta} \mathbb{E}_{y \sim \pi_{\theta}(\cdot|x)} \tilde{r}(x, y), \quad (7)$$

where  $\tilde{r}(x, y)$  represents the calibrated reward according to Eq. 6. What's more, the parameters  $\theta$  are then updated via gradient ascent:

$$\theta \leftarrow \theta + \gamma \cdot \mathbb{E}_{y \sim \pi_{\theta}(\cdot|x)} \tilde{r}(x, y), \quad (8)$$

in which  $\gamma$  represents the learning rate. This update strategy provides a robust reward estimation by incorporating auxiliary calibration from an independent judge model, thereby enhancing the effective utilization of unlabeled data during test-time training. In conclusion, DREAM demonstrates significantly improved generalizability when handling diverse inputs under unsupervised scenarios.

### 3.5 TRAINING PROCEDURE OF DREAM

To enable better test-time training with unlabeled data, DREAM carefully decouples reward estimation from reinforcement learning, allowing the model to leverage unlabeled inputs without introducing overly confident or biased updates. The process proceeds in two interlinked stages, each designed to ensure reliable reward signals and stable policy optimization. (i) **Judge model optimization.** In the first stage, DREAM constructs an auxiliary judge model that serves as a reliable reward estimator. To provide supervision for the judge model, we generate a set of additional math questions with known ground-truth answers. For each question, the policy model is queried multiple times (e.g.,  $K = 4$ ) to produce diverse candidate solutions, capturing the inherent stochasticity and variability of the model outputs. Each candidate output is then evaluated against the corresponding ground-truth answer using rule-based verification. Outputs that correctly solve the question are grouped into positive pairs, while incorrect outputs form negative pairs. The labeled pairs are subsequently used to train the judge model using a supervised learning objective, allowing it to effectively assign reliability scores to candidate outputs. (ii) **Policy model optimization.** For a given math question, we first sample multiple outputs from the policy model to construct a candidate set  $\mathcal{O}_i$ , which captures the range of plausible solutions that the policy may generate. Each candidate in  $\mathcal{O}_i$  is then evaluated by the judge model, which produces a calibration score reflecting its estimated quality. These scores are combined with a voting-based reward scheme, which are then refined by the calibration scores. Finally, these calibrated rewards are used to perform reinforcement learning updates

270  
271**Algorithm 1:** Test-time Training Procedure of DREAM

---

```

272 Input: Reference dataset  $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{N_L}$  and policy model  $\pi_\theta$ 
273 1 for iteration  $i \leftarrow 1$  to  $N_L$  do // Judge model optimization
274 2 | Generate a candidate set  $\mathcal{O}_i$  for each input  $x_i$  using  $\pi_\theta$  according to Eq. 1
275 3 | Compute the matching score between  $\hat{y}_{i,k} \in \mathcal{O}_i$  and the ground-truth  $y_i$  by Eq. 2
276 4 | Optimize the judge model  $F_\phi$  using the MSE loss defined in Eq. 3
277 5 end
278 Output: Trained judge model  $F_\phi$  // Calibration for reward reliability
279 Input: Unlabeled test dataset  $\mathcal{D}_U = \{x_i\}_{i=1}^{N_U}$  and the frozen judge model  $F_\phi$ 
280 6 for iteration  $i \leftarrow 1$  to  $N_U$  do // Policy model optimization
281 7 | Generate a candidate set  $\mathcal{O}_i$  for each input  $x_i$  using  $\pi_\theta$  according to Eq. 1
282 8 | Estimate the calibration score  $s_{i,k}$  for each candidate  $\hat{y}_{i,k}$  // Predict with judge model
283 9 | Compute the voting-based reward  $r_{i,k}^v$  within  $\mathcal{O}_i$  using Eq. 4
284 10 | Derive the calibrated reward  $r_{i,k}$  according to Eq. 5
285 11 | Normalize the group-wise reward  $\tilde{r}_{i,k}$  using Eq. 6
286 12 | Optimize  $\pi_\theta$  by Eq. 7 and Eq. 8.
287 13 end
288 Output: Trained policy model  $\pi_\theta$  // Optimize the policy with calibrated rewards
289
290
```

---

291  
292  
293

on the policy model, allowing it to improve its performance iteratively while accounting for the judge’s assessment. The entire process, including both the judge and policy model optimizations, is formally summarized in Algorithm 1.

294

## 4 EXPERIMENT

295

### 4.1 EXPERIMENTAL SETUP

296

**Backbone Models.** To verify the generalizability of our reward score calibration, we conduct experiments on a broad range of backbone models, encompassing both base and instruct models at different scales. Specifically, we select Qwen2.5-Math-1.5B (Yang et al., 2024), Qwen2.5-Math-7B (Yang et al., 2024), Qwen2.5-7B (Qwen et al., 2025), LLaMA3.1-8B-Instruct (Dubey et al., 2024), and DeepSeek-R1-7B-Instruct (Guo et al., 2025) as backbone models.

303

**Evaluation Setup.** We evaluate DREAM on three challenging mathematical reasoning benchmarks: AIME 2024, AMC, and MATH-500. For better evaluation, we adopt avg@k, maj@k, and pass@k to evaluate the experimental results. Following the practice of (Zuo et al., 2025), we generate responses with non-zero temperature sampling with a temperature of 0.6 and a top-p value of 0.95 unless specified. The maximum generation length is set to 3,072 tokens. We prompt the model to perform step-by-step reasoning and to present the final answer enclosed within “\boxed{}”.

304

**Baselines.** Our primary baseline for comparison is TTRL (Zuo et al., 2025), the state-of-the-art method for test-time reinforcement learning on unlabeled data. This direct comparison allows us to isolate and measure the performance gains attributable to our decoupled reward calibration mechanism. Additionally, we report the performance of the original backbone models without any test-time training to establish a performance floor and quantify the absolute improvement.

305

**Implementation Details.** We implement DREAM by applying GRPO (Shao et al., 2024) to the policy model, guided by our calibrated reward signals. For the policy model’s hyperparameters, we use a cosine learning rate schedule with a peak value of  $5 \times 10^{-7}$  and employ the AdamW optimizer. During the rollout phase, we sample 64 responses per prompt, which are then used for majority voting and scoring by the judge model. Subsequently, we downsample to 32 responses per prompt for the training batch. The maximum generation length is set to 3,072 tokens. The number of training episodes is adapted to the dataset size: 80 for AIME 2024, 30 for AMC, and 10 for MATH-500. All experiments were conducted on a cluster of 8 NVIDIA H20 GPUs.

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

**Reward Calibration Model.** A core component of our methodology is the judge model for reward calibration. As the proprietary datasets used for the pre-training and supervised fine-tuning (SFT) of

Table 1: Performance comparison with TTRL on various backbone models. We report avg@16 on AIME 2024, AMC, and MATH-500 datasets.

Backbone	Model	AIME 2024	AMC	MATH-500	Avg.
<i>Math Base Models</i>	Qwen2.5-Math-1.5B	7.5	29.5	33.8	23.6
	w/ TTRL	12.1 $\uparrow$ 4.6	45.9 $\uparrow$ 16.4	68.2 $\uparrow$ 34.4	42.1 $\uparrow$ 18.5
	w/ DREAM	<b>16.7</b> $\uparrow$ 9.2	<b>46.8</b> $\uparrow$ 17.3	<b>72.9</b> $\uparrow$ 39.1	<b>45.5</b> $\uparrow$ 21.9
	Qwen2.5-Math-7B	13.5	36.5	46.5	32.2
	w/ TTRL	<b>40.8</b> $\uparrow$ 27.3	67.0 $\uparrow$ 30.5	83.4 $\uparrow$ 36.9	63.7 $\uparrow$ 31.5
	w/ DREAM	40.0 $\uparrow$ 26.5	<b>67.4</b> $\uparrow$ 30.9	<b>84.5</b> $\uparrow$ 38.0	<b>64.0</b> $\uparrow$ 31.8
<i>Vanilla Base Model</i>	Qwen2.5-7B	7.3	32.0	60.9	33.4
	w/ TTRL	23.3 $\uparrow$ 16.0	51.8 $\uparrow$ 19.8	<b>81.1</b> $\uparrow$ 20.2	52.1 $\uparrow$ 18.7
	w/ DREAM	<b>23.3</b> $\uparrow$ 16.0	<b>55.4</b> $\uparrow$ 23.4	80.1 $\uparrow$ 19.2	<b>52.9</b> $\uparrow$ 19.5
<i>Instruct Model</i>	LLaMA3.1-8B-Instruct	4.8	21.5	48.4	24.9
	w/ TTRL	10.0 $\uparrow$ 5.2	32.3 $\uparrow$ 10.8	<b>61.4</b> $\uparrow$ 13.0	34.6 $\uparrow$ 9.7
	w/ DREAM	<b>13.3</b> $\uparrow$ 8.5	<b>34.8</b> $\uparrow$ 13.3	59.6 $\uparrow$ 11.2	<b>35.9</b> $\uparrow$ 11.0
	DeepSeek-R1-7B-Instruct	1.7	16.0	43.1	20.3
	w/ TTRL	1.0 $\downarrow$ 0.7	19.3 $\uparrow$ 3.3	50.8 $\uparrow$ 7.7	23.7 $\uparrow$ 3.4
	w/ DREAM	<b>1.7</b> $\uparrow$ 0.0	<b>20.8</b> $\uparrow$ 4.8	<b>51.5</b> $\uparrow$ 8.4	<b>24.7</b> $\uparrow$ 4.4

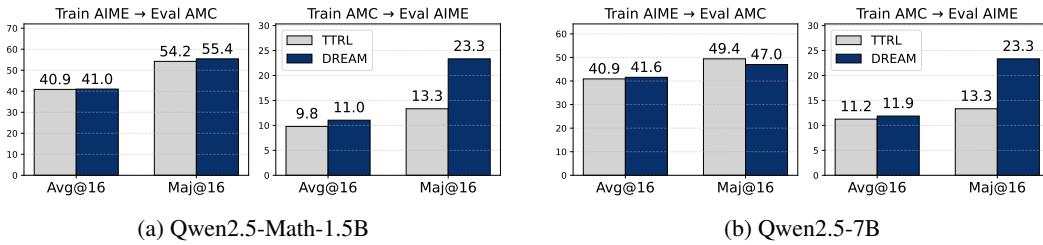


Figure 2: Performance comparison of our DREAM and TTRL on out-of-distribution testing data. We select Qwen2.5-Math-1.5B as the backbone model and report avg@16.

the backbone models are often not accessible, we simulate a representative labeled dataset to train our reward calibration model. We utilize the training split of the MATH dataset for this purpose. For each question in the MATH training set (Hendrycks et al., 2024), we generate four candidate responses from the target policy model using a sampling temperature of 0.6 and a top-p of 0.9. These responses are then assigned a binary score based on a rule-based comparison against the ground-truth answer. This process yields a dataset of  $\langle \text{question}, \text{response}, \text{score} \rangle$  tuples for training the judge model as a regression task. Further details on the judge model’s training data distribution, architecture, and optimization are provided in Appendix A.

## 4.2 EXPERIMENT RESULTS AND ANALYSIS

**Reward calibration generally leads to better test-time RL training.** As shown in Table 1, DREAM mitigates the accumulation of reward noise caused by overconfidence during training by calibrating the voting-based reward with a decoupled reliability estimation, thereby demonstrating a positive average performance gain over the majority voting baseline on most of the scenarios. The benefits of our calibration mechanism are most pronounced for weaker models, where the initial reward signal from majority voting is inherently noisier. For example, DREAM achieves a significant improvement of 4.6 on AIME 2024 for the smaller Qwen2.5-Math-1.5B, as its judge model effectively filters erroneous positive rewards and mitigates overconfident optimization on flawed outputs. However, the impact of calibration can be nuanced. We observed a marginal performance decrease with DREAM on Qwen2.5-Math-7B for the AIME dataset. This could be attributed to our implementation choice for computational efficiency, where a single reward calibration model based on Qwen2.5-Math-1.5B was used across all experiments. This approach may be suboptimal when applied to larger, more capable models.

**Reward calibration generalizes well beyond the target task.** We evaluate the models trained with DREAM on out-of-distribution (OOD) test data to assess their generalization capabilities. Specif-

378  
 379  
 380  
 381  
 382  
 383  
 384  
 385  
 386  
 387  
 388  
 389  
 390  
 391  
 392  
 393  
 394  
 395  
 396  
 397  
 398  
 399  
 400  
 401  
 402  
 403  
 404  
 405  
 406  
 407  
 408  
 409  
 410  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447  
 448  
 449  
 450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462  
 463  
 464  
 465  
 466  
 467  
 468  
 469  
 470  
 471  
 472  
 473  
 474  
 475  
 476  
 477  
 478  
 479  
 480  
 481  
 482  
 483  
 484  
 485  
 486  
 487  
 488  
 489  
 490  
 491  
 492  
 493  
 494  
 495  
 496  
 497  
 498  
 499  
 500  
 501  
 502  
 503  
 504  
 505  
 506  
 507  
 508  
 509  
 510  
 511  
 512  
 513  
 514  
 515  
 516  
 517  
 518  
 519  
 520  
 521  
 522  
 523  
 524  
 525  
 526  
 527  
 528  
 529  
 530  
 531  
 532  
 533  
 534  
 535  
 536  
 537  
 538  
 539  
 540  
 541  
 542  
 543  
 544  
 545  
 546  
 547  
 548  
 549  
 550  
 551  
 552  
 553  
 554  
 555  
 556  
 557  
 558  
 559  
 560  
 561  
 562  
 563  
 564  
 565  
 566  
 567  
 568  
 569  
 570  
 571  
 572  
 573  
 574  
 575  
 576  
 577  
 578  
 579  
 580  
 581  
 582  
 583  
 584  
 585  
 586  
 587  
 588  
 589  
 590  
 591  
 592  
 593  
 594  
 595  
 596  
 597  
 598  
 599  
 600  
 601  
 602  
 603  
 604  
 605  
 606  
 607  
 608  
 609  
 610  
 611  
 612  
 613  
 614  
 615  
 616  
 617  
 618  
 619  
 620  
 621  
 622  
 623  
 624  
 625  
 626  
 627  
 628  
 629  
 630  
 631  
 632  
 633  
 634  
 635  
 636  
 637  
 638  
 639  
 640  
 641  
 642  
 643  
 644  
 645  
 646  
 647  
 648  
 649  
 650  
 651  
 652  
 653  
 654  
 655  
 656  
 657  
 658  
 659  
 660  
 661  
 662  
 663  
 664  
 665  
 666  
 667  
 668  
 669  
 670  
 671  
 672  
 673  
 674  
 675  
 676  
 677  
 678  
 679  
 680  
 681  
 682  
 683  
 684  
 685  
 686  
 687  
 688  
 689  
 690  
 691  
 692  
 693  
 694  
 695  
 696  
 697  
 698  
 699  
 700  
 701  
 702  
 703  
 704  
 705  
 706  
 707  
 708  
 709  
 710  
 711  
 712  
 713  
 714  
 715  
 716  
 717  
 718  
 719  
 720  
 721  
 722  
 723  
 724  
 725  
 726  
 727  
 728  
 729  
 730  
 731  
 732  
 733  
 734  
 735  
 736  
 737  
 738  
 739  
 740  
 741  
 742  
 743  
 744  
 745  
 746  
 747  
 748  
 749  
 750  
 751  
 752  
 753  
 754  
 755  
 756  
 757  
 758  
 759  
 760  
 761  
 762  
 763  
 764  
 765  
 766  
 767  
 768  
 769  
 770  
 771  
 772  
 773  
 774  
 775  
 776  
 777  
 778  
 779  
 780  
 781  
 782  
 783  
 784  
 785  
 786  
 787  
 788  
 789  
 790  
 791  
 792  
 793  
 794  
 795  
 796  
 797  
 798  
 799  
 800  
 801  
 802  
 803  
 804  
 805  
 806  
 807  
 808  
 809  
 810  
 811  
 812  
 813  
 814  
 815  
 816  
 817  
 818  
 819  
 820  
 821  
 822  
 823  
 824  
 825  
 826  
 827  
 828  
 829  
 830  
 831  
 832  
 833  
 834  
 835  
 836  
 837  
 838  
 839  
 840  
 841  
 842  
 843  
 844  
 845  
 846  
 847  
 848  
 849  
 850  
 851  
 852  
 853  
 854  
 855  
 856  
 857  
 858  
 859  
 860  
 861  
 862  
 863  
 864  
 865  
 866  
 867  
 868  
 869  
 870  
 871  
 872  
 873  
 874  
 875  
 876  
 877  
 878  
 879  
 880  
 881  
 882  
 883  
 884  
 885  
 886  
 887  
 888  
 889  
 890  
 891  
 892  
 893  
 894  
 895  
 896  
 897  
 898  
 899  
 900  
 901  
 902  
 903  
 904  
 905  
 906  
 907  
 908  
 909  
 910  
 911  
 912  
 913  
 914  
 915  
 916  
 917  
 918  
 919  
 920  
 921  
 922  
 923  
 924  
 925  
 926  
 927  
 928  
 929  
 930  
 931  
 932  
 933  
 934  
 935  
 936  
 937  
 938  
 939  
 940  
 941  
 942  
 943  
 944  
 945  
 946  
 947  
 948  
 949  
 950  
 951  
 952  
 953  
 954  
 955  
 956  
 957  
 958  
 959  
 960  
 961  
 962  
 963  
 964  
 965  
 966  
 967  
 968  
 969  
 970  
 971  
 972  
 973  
 974  
 975  
 976  
 977  
 978  
 979  
 980  
 981  
 982  
 983  
 984  
 985  
 986  
 987  
 988  
 989  
 990  
 991  
 992  
 993  
 994  
 995  
 996  
 997  
 998  
 999  
 1000  
 1001  
 1002  
 1003  
 1004  
 1005  
 1006  
 1007  
 1008  
 1009  
 1010  
 1011  
 1012  
 1013  
 1014  
 1015  
 1016  
 1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079  
 1080  
 1081  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092  
 1093  
 1094  
 1095  
 1096  
 1097  
 1098  
 1099  
 1100  
 1101  
 1102  
 1103  
 1104  
 1105  
 1106  
 1107  
 1108  
 1109  
 1110  
 1111  
 1112  
 1113  
 1114  
 1115  
 1116  
 1117  
 1118  
 1119  
 1120  
 1121  
 1122  
 1123  
 1124  
 1125  
 1126  
 1127  
 1128  
 1129  
 1130  
 1131  
 1132  
 1133  
 1134  
 1135  
 1136  
 1137  
 1138  
 1139  
 1140  
 1141  
 1142  
 1143  
 1144  
 1145  
 1146  
 1147  
 1148  
 1149  
 1150  
 1151  
 1152  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167  
 1168  
 1169  
 1170  
 1171  
 1172  
 1173  
 1174  
 1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187  
 1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238  
 1239  
 1240  
 1241  
 1242  
 1243  
 1244  
 1245  
 1246  
 1247  
 1248  
 1249  
 1250  
 1251  
 1252  
 1253  
 1254  
 1255  
 1256  
 1257  
 1258  
 1259  
 1260  
 1261  
 1262  
 1263  
 1264  
 1265  
 1266  
 1267  
 1268  
 1269  
 1270  
 1271  
 1272  
 1273  
 1274  
 1275  
 1276  
 1277  
 1278  
 1279  
 1280  
 1281  
 1282  
 1283  
 1284  
 1285  
 1286  
 1287  
 1288  
 1289  
 1290  
 1291  
 1292  
 1293  
 1294  
 1295  
 1296  
 1297  
 1298  
 1299  
 1300  
 1301  
 1302  
 1303  
 1304  
 1305  
 1306  
 1307  
 1308  
 1309  
 1310  
 1311  
 1312  
 1313  
 1314  
 1315  
 1316  
 1317  
 1318  
 1319  
 1320  
 1321  
 1322  
 1323  
 1324  
 1325  
 1326  
 1327  
 1328  
 1329  
 1330  
 1331  
 1332  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362  
 1363  
 1364  
 1365  
 1366  
 1367  
 1368  
 1369  
 1370  
 1371  
 1372  
 1373  
 1374  
 1375  
 1376  
 1377  
 1378  
 1379  
 1380  
 1381  
 1382  
 1383  
 1384  
 1385  
 1386  
 1387  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396  
 1397  
 1398  
 1399  
 1400  
 1401  
 1402  
 1403  
 1404  
 1405  
 1406  
 1407  
 1408  
 1409  
 1410  
 1411  
 1412  
 1413  
 1414  
 1415  
 1416  
 1417  
 1418  
 1419  
 1420  
 1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428  
 1429  
 1430  
 1431  
 1432  
 1433  
 1434  
 1435  
 1436  
 1437  
 1438  
 1439  
 1440  
 1441  
 1442  
 1443  
 1444  
 1445  
 1446  
 1447  
 1448  
 1449  
 1450  
 1451  
 1452  
 1453  
 1454  
 1455  
 1456  
 1457  
 1458  
 1459  
 1460  
 1461  
 1462  
 1463  
 1464  
 1465  
 1466  
 1467  
 1468  
 1469  
 1470  
 1471  
 1472  
 1473  
 1474  
 1475  
 1476  
 1477  
 1478  
 1479  
 1480  
 1481  
 1482  
 1483  
 1484  
 1485  
 1486  
 1487  
 1488  
 1489  
 1490  
 1491  
 1492  
 1493  
 1494  
 1495  
 1496  
 1497  
 1498  
 1499  
 1500  
 1501  
 1502  
 1503  
 1504  
 1505  
 1506  
 1507  
 1508  
 1509  
 1510  
 1511  
 1512  
 1513  
 1514  
 1515  
 1516  
 1517  
 1518  
 1519  
 1520  
 1521  
 1522  
 1523  
 1524  
 1525  
 1526  
 1527  
 1528  
 1529  
 1530  
 1531  
 1532  
 1533  
 1534  
 1535  
 1536  
 1537  
 1538  
 1539  
 1540  
 1541  
 1542  
 1543  
 1544  
 1545  
 1546  
 1547  
 1548  
 1549  
 1550  
 1551  
 1552  
 1553  
 1554  
 1555  
 1556  
 1557  
 1558  
 1559  
 1560  
 1561  
 1562  
 1563  
 1564  
 1565  
 1566  
 1567  
 1568  
 1569  
 1570  
 1571  
 1572  
 1573  
 1574  
 1575  
 1576  
 1577  
 1578  
 1579  
 1580  
 1581  
 1582  
 1583  
 1584  
 1585  
 1586  
 1587  
 1588  
 1589  
 1590  
 1591  
 1592  
 1593  
 1594  
 1595  
 1596  
 1597  
 1598  
 1599  
 1600  
 1601  
 1602  
 1603  
 1604  
 1605  
 1606  
 1607  
 1608  
 1609  
 1610  
 1611  
 1612  
 1613  
 1614  
 1615  
 1616  
 1617  
 1618  
 1619  
 1620  
 1621  
 1622  
 1623  
 1624  
 1625  
 1626  
 1627  
 1628  
 1629  
 1630  
 1631  
 1632  
 1633  
 1634  
 1635  
 1636  
 1637  
 1638  
 1639  
 1640  
 1641  
 1642  
 1643  
 1644  
 1645  
 1646  
 1647  
 1648  
 1649  
 1650  
 1651  
 1652  
 1653  
 1654  
 1655  
 1656  
 1657  
 1658  
 1659  
 1660  
 1661  
 1662  
 1663  
 1664  
 1665  
 1666  
 1667  
 1668  
 1669  
 1670  
 1671  
 1672  
 1673  
 1674  
 1675  
 1676  
 1677  
 1678  
 1679  
 1680  
 1681  
 1682  
 1683  
 1684  
 1685  
 1686  
 1687  
 1688  
 1689  
 1690  
 1691  
 1692  
 1693  
 1694  
 1695  
 1696  
 1697  
 1698  
 1699  
 1700  
 1701  
 1702  
 1703  
 1704  
 1705  
 1706  
 1707  
 1708  
 1709  
 1710  
 1711  
 1712  
 1713  
 1714  
 1715  
 1716  
 1717  
 1718  
 1719  
 1720  
 1721  
 1722  
 1723  
 1724  
 1725  
 1726  
 1727  
 1728  
 1729  
 1730  
 1731  
 1732  
 1733  
 1734  
 1735  
 1736  
 1737  
 1738  
 1739  
 1740  
 1741  
 1742  
 1743  
 1744  
 1745  
 1746  
 1747  
 1748  
 1749  
 1750  
 1751  
 1752  
 1753  
 1754  
 1755  
 1756  
 1757  
 1758  
 1759  
 1760  
 1761  
 1762  
 1763  
 1764  
 1765  
 1766  
 1767  
 1768  
 1769  
 1770  
 1771  
 1772  
 1773  
 1774  
 1775  
 1776  
 1777  
 1778  
 1779  
 1780  
 1781  
 1782  
 1783  
 1784  
 1785  
 1786  
 1787  
 1788  
 1789  
 1790  
 1791  
 1792  
 1793  
 1794  
 1795  
 1796  
 1797  
 1798  
 1799  
 1800  
 1801  
 1802  
 1803  
 1804  
 1805  
 1806  
 1807  
 1808  
 1809  
 1810  
 1811  
 1812  
 1813  
 1814  
 1815  
 1816  
 1817  
 1818  
 1819  
 1820  
 1821  
 1822  
 1823  
 1824  
 1825  
 1826  
 1827  
 1828  
 1829  
 1830  
 1831  
 1832  
 1833  
 1834  
 1835  
 1836  
 1837  
 1838  
 1839  
 1840  
 1841  
 1842  
 1843  
 1844  
 1845  
 1846  
 1847  
 1848  
 1849  
 1850  
 1851  
 1852  
 1853  
 1854  
 1855  
 1856  
 1857  
 1858  
 1859  
 1860  
 1861  
 1862  
 1863  
 1864  
 1865  
 1866  
 1867  
 1868  
 1869  
 1870

432 a preliminary exploration of this approach, where we replaced the internal model with Qwen2.5-  
 433 Math-PRM-7B.

434 Here, instead of using process-  
 435 supervised rewards, we compute  
 436 the reward directly at the entire response  
 437 level. We performed tests similar to  
 438 those in Table 2 and found that the  
 439 quality of the reliability scores obtained  
 440 this way is viable. The comparison  
 441 between the internal and external cali-  
 442 bration models is presented in Table 4.  
 443 As shown, their performance is largely  
 444 on par across three models and two  
 445 datasets. Our model was trained on data consisting of only  $\sim 7k$  queries with 4 responses sampled  
 446 for each, whereas Qwen2.5-Math-PRM-7B was trained on  $\sim 500k$  queries with 6-8 responses each.  
 447 This indicates that for our task, fine-tuning the model itself on a small-scale dataset can achieve  
 448 performance comparable to that of a general-purpose math reward model trained on a much larger  
 449 dataset. In the future, we will further explore the use of more powerful external calibration models,  
 450 such as Qwen2.5-Math-PRM-72B or Qwen2.5-Math-RM-72B.

451 **The effectiveness of test-time RL.** To demonstrate the efficacy of test-time RL, we compare the  
 452 performance of Qwen2.5-Math-1.5B against its instruction-tuned version and similarly-sized, RL-  
 453 trained models, using the pass@1 score with greedy decoding. The results in Table 5 indicate that  
 454 test-time RL with the voting-based TTTRL already surpasses the heavily instruction-tuned Qwen2.5-  
 455 Math-1.5B-Instruct. Building on this, our proposed DREAM further improves performance by  
 456 decoupling reliability estimation from reward calibration, achieving results comparable to models  
 457 trained with large-scale RL. Importantly, large-scale RL methods rely on labeled datasets, whereas  
 458 such labels are unavailable in our setting. Under these constraints, DREAM delivers competitive  
 459 performance with only scarce labels and relatively small-scale RL during test-time training. These  
 460 findings highlight both the efficiency and practicality of test-time adaptation, underscoring the broad  
 461 applicability of DREAM comprehensively.

462 Table 5: Pass@1 comparison on AIME 2024 and AMC against similar models trained on large-scale  
 463 labeled data. All results are based on greedy decoding, with baselines sourced from Dr. GRPO.

Model	AIME 2024	AMC	MATH-500	Avg.
Qwen2.5-Math-1.5B	20.0	32.5	36.2	29.6
w/ TTTRL	16.7	48.2	72.4	45.8
w/ DREAM	20.0	50.6	73.8	48.1
Qwen2.5-Math-1.5B-Instruct	10.0	48.2	74.2	44.1
DeepSeek-R1-Distill-1.5B-@3k	2.5	21.7	52.2	25.5
DeepSeek-R1-Distill-1.5B-@8k	20.0	49.4	77.4	48.9
Oat-Zero-1.5B	20.0	53.0	74.2	49.1

## 5 CONCLUSION

477 In this work, we introduce DREAM, a novel reward calibration framework that decouples reward  
 478 estimation from reinforcement learning to address the challenge of unreliable rewards on unlabeled  
 479 data. To disentangle reward estimation from reinforcement learning, DREAM introduces an aux-  
 480 iliary judge model trained to assess the consistency between generated responses and ground-truth  
 481 answers. By incorporating the judge model to calibrate voting-based rewards, DREAM enables  
 482 more accurate policy optimization without explicit supervision. Extensive experiments on multiple  
 483 mathematical reasoning benchmarks and diverse LLM backbones demonstrate the generalizability  
 484 of our approach. We believe DREAM opens up promising directions for advancing test-time scaling  
 485 of LLMs, particularly in scenarios where labeled data is scarce or unavailable.

486 REFERENCES  
487

488 Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn  
489 Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless  
490 assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*,  
491 2022.

492 Alexander Brinkmann and Christian Bizer. Self-refinement strategies for llm-based product attribute  
493 value extraction. *arXiv preprint arXiv:2501.01237*, 2025.

494

495 Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan  
496 Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM*  
497 *transactions on intelligent systems and technology*, 15(3):1–45, 2024.

498 Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan,  
499 Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. Rlhf deciphered: A criti-  
500 cal analysis of reinforcement learning from human feedback for llms. *ACM Computing Surveys*,  
501 2024.

502

503 Jack Chen, Fazhong Liu, Naruto Liu, Yuhang Luo, Erqu Qin, Harry Zheng, Tian Dong, Haojin Zhu,  
504 Yan Meng, and Xiao Wang. Step-wise adaptive integration of supervised fine-tuning and rein-  
505 forcement learning for task-specific llms. *arXiv preprint arXiv:2505.13026*, 2025a.

506 Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan,  
507 Wen Zhang, Huajun Chen, Fan Yang, et al. Learning to reason with search for llms via reinforce-  
508 ment learning. *arXiv preprint arXiv:2503.19470*, 2025b.

509

510 Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small llms: What works  
511 and what doesn't. *arXiv preprint arXiv:2503.16219*, 2025.

512

513 Haotian Dong, Jingyan Jiang, Rongwei Lu, Jiajun Luo, Jiajun Song, Bowen Li, Ying Shen, and  
514 Zhi Wang. Beyond a single ai cluster: A survey of decentralized llm training. *arXiv preprint*  
515 *arXiv:2503.11023*, 2025.

516

517 Yong Du, Yuchen Yan, Fei Tang, Zhengxi Lu, Chang Zong, Weiming Lu, Shengpei Jiang, and  
518 Yongliang Shen. Test-time reinforcement learning for gui grounding via region consistency. *arXiv*  
519 *preprint arXiv:2508.05615*, 2025.

520

521 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha  
522 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.  
523 *arXiv e-prints*, pp. arXiv–2407, 2024.

524

525 Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik,  
526 and Rohin Shah. Challenges with unsupervised llm knowledge discovery. *arXiv preprint*  
527 *arXiv:2312.10029*, 2023.

528

529 Stefan Feuerriegel, Abdurahman Maarouf, Dominik Bär, Dominique Geissler, Jonas Schweisthal,  
530 Nicolas Pröllochs, Claire E Robertson, Steve Rathje, Jochen Hartmann, Saif M Mohammad, et al.  
531 Using natural language processing to analyse text data in behavioural science. *Nature Reviews*  
532 *Psychology*, 4(2):96–111, 2025.

533

534 Simona Frenda, Gavin Abercrombie, Valerio Basile, Alessandro Pedrani, Raffaella Panizzon,  
535 Alessandra Teresa Cignarella, Cristina Marco, and Davide Bernardi. Perspectivist approaches  
536 to natural language processing: a survey. *Language Resources and Evaluation*, 59(2):1719–1746,  
537 2025.

538

539 Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al.  
540 Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*,  
541 36:5539–5568, 2023.

542

543 Bishwamittra Ghosh, Sarah Hasan, Naheed Anjum Arifat, and Arijit Khan. Logical consistency of  
544 large language models in fact-checking. *arXiv preprint arXiv:2412.16100*, 2024.

540 Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao  
 541 Bi, Yu Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming—  
 542 the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

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

546 Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. Llm multi-agent sys-  
 547 tems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

548 Mohanakrishnan Hariharan. Reinforcement learning: Advanced techniques for llm behavior op-  
 549 timization. *ESP International Journal of Advancements in Computational Technology (ESP-  
 550 IJACT)*, 2(2):84–101, 2025.

551 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,  
 552 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. *URL*  
 553 <https://arxiv.org/abs/2103.03874>, 2, 2024.

554 Jingcheng Hu, Yinmin Zhang, Qi Han, Dixin Jiang, Xiangyu Zhang, and Heung-Yeung Shum.  
 555 Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base  
 556 model. *arXiv preprint arXiv:2503.24290*, 2025.

557 Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and  
 558 Robert McHardy. Challenges and applications of large language models. *arXiv preprint  
 559 arXiv:2307.10169*, 2023.

560 Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank  
 561 Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, et al. Chatgpt for  
 562 good? on opportunities and challenges of large language models for education. *Learning and  
 563 individual differences*, 103:102274, 2023.

564 Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: state  
 565 of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744,  
 566 2023.

567 Anton Korikov, Pan Du, Scott Sanner, and Navid Rekabsaz. Batched self-consistency improves llm  
 568 relevance assessment and ranking. *arXiv preprint arXiv:2505.12570*, 2025.

569 Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal,  
 570 Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan.  
 571 Llm post-training: A deep dive into reasoning large language models. *arXiv preprint  
 572 arXiv:2502.21321*, 2025.

573 Hanyu Lai, Xiao Liu, Junjie Gao, Jiale Cheng, Zehan Qi, Yifan Xu, Shuntian Yao, Dan Zhang,  
 574 Jinhua Du, Zhenyu Hou, et al. A survey of post-training scaling in large language models. In *Pro-  
 575 ceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume  
 576 1: Long Papers)*, pp. 2771–2791, 2025.

577 Alireza Rashidi Laleh and Majid Nili Ahmadabadi. A survey on enhancing reinforcement learning in  
 578 complex environments: Insights from human and llm feedback. *arXiv preprint arXiv:2411.13410*,  
 579 2024.

580 Xiang Li, Jiayi Xin, Qi Long, and Weijie J Su. Evaluating the unseen capabilities: How many  
 581 theorems do llms know? *arXiv preprint arXiv:2506.02058*, 2025.

582 Xingtai Lv, Yuxin Zuo, Youbang Sun, Hongyi Liu, Yuntian Wei, Zhekai Chen, Lixuan He, Xuekai  
 583 Zhu, Kaiyan Zhang, Bingning Wang, et al. Towards a unified view of large language model  
 584 post-training. *arXiv preprint arXiv:2509.04419*, 2025.

585 Xuezhe Ma, Xiaomeng Yang, Wenhan Xiong, Beidi Chen, Lili Yu, Hao Zhang, Jonathan May, Luke  
 586 Zettlemoyer, Omer Levy, and Chunting Zhou. Megalodon: Efficient llm pretraining and inference  
 587 with unlimited context length. *Advances in Neural Information Processing Systems*, 37:71831–  
 588 71854, 2024.

594 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri  
 595 Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement  
 596 with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.  
 597

598 Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Am-  
 599 atriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*,  
 600 2024.

601 Youssef Mroueh, Nicolas Dupuis, Brian Belgodere, Apoorva Nitsure, Mattia Rigotti, Kristjan Gree-  
 602 newald, Jiri Navratil, Jerret Ross, and Jesus Rios. Revisiting group relative policy optimization:  
 603 Insights into on-policy and off-policy training. *arXiv preprint arXiv:2505.22257*, 2025.

604 Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman,  
 605 Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language  
 606 models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

607

608 Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and  
 609 Stefan Zohren. A survey of large language models for financial applications: Progress, prospects  
 610 and challenges. *arXiv preprint arXiv:2406.11903*, 2024.

611 Augustus Odena, Dieterich Lawson, and Christopher Olah. Changing model behavior at test-time  
 612 using reinforcement learning. *arXiv preprint arXiv:1702.07780*, 2017.

613

614 David Osowiechi, Gustavo A Vargas Hakim, Mehrdad Noori, Milad Cheraghalkhani, Ali Bahri,  
 615 Moslem Yazdanpanah, Ismail Ben Ayed, and Christian Desrosiers. Nc-ttt: A noise contrastive  
 616 approach for test-time training. In *Proceedings of the IEEE/CVF Conference on Computer Vision  
 617 and Pattern Recognition*, pp. 6078–6086, 2024.

618 Young-Jin Park, Kristjan Greenewald, Kaveh Alim, Hao Wang, and Navid Azizan. Know what you  
 619 don't know: Uncertainty calibration of process reward models. *arXiv preprint arXiv:2506.09338*,  
 620 2025.

621

622 Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander  
 623 Panchenko, and Mikhail Salnikov. How much knowledge can you pack into a lora adapter without  
 624 harming llm? *arXiv preprint arXiv:2502.14502*, 2025.

625 Jianing Qi, Hao Tang, and Zhigang Zhu. Verifierq: Enhancing llm test time compute with q-learning-  
 626 based verifiers. *arXiv preprint arXiv:2410.08048*, 2024.

627

628 Yuxiao Qu, Matthew YR Yang, Amirth Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan  
 629 Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-  
 630 tuning. *arXiv preprint arXiv:2503.07572*, 2025.

631 Qwen, , An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan  
 632 Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang,  
 633 Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin  
 634 Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li,  
 635 Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,  
 636 Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv  
 637 preprint arXiv:2412.15115*, 2025.

638 Soham Sane. Hybrid group relative policy optimization: A multi-sample approach to enhancing  
 639 policy optimization. *arXiv preprint arXiv:2502.01652*, 2025.

640

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

643

644 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,  
 645 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathemati-  
 646 cal reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

647 Charlie Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. Offline rl for natural language  
 648 generation with implicit language q learning. In *ICLR*, 2023.

648 Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Ke-  
 649 icher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated  
 650 confidence expression of large language models. *arXiv preprint arXiv:2503.02623*, 2025.

651

652 Hao Sun and Mihaela van der Schaar. Inverse reinforcement learning meets large language model  
 653 post-training: Basics, advances, and opportunities. *arXiv preprint arXiv:2507.13158*, 2025.

654

655 Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time train-  
 656 ing with self-supervision for generalization under distribution shifts. In *International conference*  
 657 *on machine learning*, pp. 9229–9248. PMLR, 2020.

658

659 Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang  
 660 Wang, Raluca Popa, and Ion Stoica. Judgebench: A benchmark for evaluating LLM-based judges.  
 661 In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=G0dkxFayVq>.

662

663 Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez,  
 664 Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*,  
 29(8):1930–1940, 2023.

665

666 Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang,  
 667 Jiangyue Yan, Yao Su, et al. A survey on post-training of large language models. *arXiv e-prints*,  
 668 pp. arXiv–2503, 2025.

669

670 Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin,  
 671 Jinhu Fu, Yibo Yan, Hanjun Luo, et al. A comprehensive survey in llm (-agent) full stack safety:  
 672 Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

673

674 Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu,  
 675 Guoyin Wang, and Eduard Hovy. Reinforcement learning enhanced llms: A survey. *arXiv preprint*  
 676 *arXiv:2412.10400*, 2024.

677

678 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdh-  
 679 ery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models.  
 680 *arXiv preprint arXiv:2203.11171*, 2022.

681

682 Yuchen Yan, Jin Jiang, Yang Liu, Yixin Cao, Xin Xu, Mengdi Zhang, Xunliang Cai, and Jian Shao.  
 683  $\mathcal{S}^3$ cmath: Spontaneous step-level self-correction makes large language models better mathemat-  
 684 ical reasoners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp.  
 685 25588–25596, 2025.

686

687 An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jian-  
 688 hong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical  
 689 expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

690

691 Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian  
 692 Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system  
 693 at scale. *arXiv preprint arXiv:2503.14476*, 2025.

694

695 Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning: The  
 696 effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*, 2024.

697

698 Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou,  
 699 Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning  
 700 for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a.

701

702 Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng  
 703 Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group  
 704 relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025b.

705

706 Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan  
 707 Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language  
 708 models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025c.

702 Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan  
703 Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint*  
704 *arXiv:2505.23884*, 2025d.

705

706 Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang,  
707 Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Trans-*  
708 *actions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

709 Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Ku-  
710 likov. The majority is not always right: RL training for solution aggregation. *arXiv preprint*  
711 *arXiv:2509.06870*, 2025.

712

713 Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao  
714 Xiong, Jessica Fan, Yurong Mou, et al. Rmb: Comprehensively benchmarking reward models in  
715 llm alignment. *arXiv preprint arXiv:2410.09893*, 2024.

716 Zhi Zhou, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Xiaoxing Ma, and Yu-Feng Li. Bridging  
717 internal probability and self-consistency for effective and efficient llm reasoning. *arXiv preprint*  
718 *arXiv:2502.00511*, 2025.

719 Elaine Zosa, Ville Komulainen, and Sampo Pyysalo. Got compute, but no data: Lessons from  
720 post-training a finnish llm. *arXiv preprint arXiv:2503.09407*, 2025.

721

722 Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen  
723 Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint*  
724 *arXiv:2504.16084*, 2025.

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 A DETAILS OF JUDGE MODEL  
757758  
759 A.1 TRAINING DATA CONSTRUCTION  
760

761 The training data for our judge model was constructed using the training split of the MATH dataset as  
762 the source of labeled data. Each sample in this dataset contains a mathematical problem (question)  
763 and its corresponding golden answer. For each question, we employed our backbone model to  
764 generate four diverse candidate responses. To encourage diversity, we used nucleus sampling with a  
765 temperature of 0.6 and a top-p value of 0.9. Each of the generated responses was then automatically  
766 evaluated for correctness by comparing its final answer to the golden answer from the MATH dataset.  
767 A binary score was assigned: a score of 1 indicates a correct response, while a score of 0 indicates  
768 an incorrect one. This procedure resulted in a dataset comprising  $\sim 26,000$  training samples and  
769  $\sim 3,000$  test samples. Each sample is a triplet containing the original question, a model-generated  
770 response, and its assigned binary correctness score.  
771

772 A.2 MODEL ARCHITECTURE  
773

774 We fine-tune our backbone model it-  
775 self to act as the judge model for  
776 the subsequent test-time RL, pre-  
777 dicting a scalar score for a given  
778 question-response pair. In this pa-  
779 per, we consistently use the judge  
780 model built upon Qwen2.5-Math-  
781 1.5B to reduce computational over-  
782 head. Specifically, we initialized  
783 the model from the pretrained back-  
784 bone model. We then introduced  
785 a new special token, `<| judge|>`,  
786 into the tokenizer’s vocabulary. This  
787 token is appended to the input se-  
788 quence to signal the model to per-  
789 form the evaluation task. To pro-  
790 duce a fixed-size representation of the  
791 entire input sequence, we employed  
792 an attention pooling strategy, which  
793 uses a learnable query vector to com-  
794 pute an attention-weighted average of  
795 the transformer’s final hidden states.  
796 This allows the model to dynamically  
797 determine which tokens are most rel-  
798 evant for the final score prediction.  
799 Finally, the pooled hidden represen-  
800 tation is fed into a regression head to  
801 predict the final score. This head is a  
802 two-layer MLP with a SiLU activation function. The hidden dimension of the MLP is set to twice  
803 the model’s hidden size, and a dropout rate of 0.05 is applied for regularization.  
804

805 A.3 TRAINING HYPERPARAMETERS  
806

807 The model was trained using the hyperparameters specified in Table 6. We use AdamW for opti-  
808 mization with an initial learning rate of  $2 \times 10^{-5}$  that followed a cosine-annealing decay schedule  
809 with a warm-up ratio of 0.03 and a weight decay coefficient of 0.01. The model was trained for 3  
epochs with a per-device mini-batch of 4. Binary cross-entropy (BCE) loss was used as the training  
objective.

Table 6: Hyperparameters for Judge Model Training.

Parameter Category	Value
<b>Model Configuration</b>	
Base Model	Qwen2.5-Math-1.5B
Pooling Strategy	Attention Pooling
Regression Head	2-layer MLP
Head Hidden Multiplier	2
Head Dropout	0.05
<b>Data Configuration</b>	
Training Samples	$\sim 26,000$
Evaluation Samples	$\sim 3,000$
Max Sequence Length	4096
<b>Optimizer &amp; Scheduler</b>	
Optimizer	AdamW
Learning Rate	$2 \times 10^{-5}$
LR Scheduler	Cosine Annealing
Warmup Ratio	0.03
Weight Decay	0.01
<b>Training Strategy</b>	
Training Epochs	3
Per-Device Train Batch Size	4
Per-Device Eval Batch Size	8
Gradient Accumulation Steps	1
Loss Function	BCE Loss

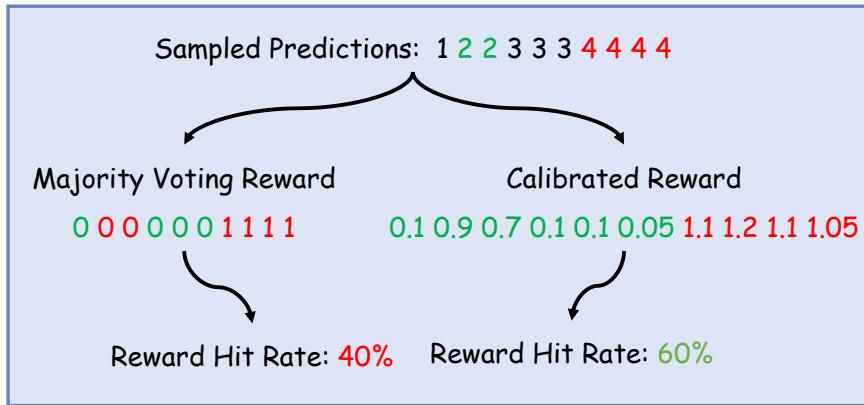


Figure 3: A simple case study of reward calibration.

## B A SIMPLE CASE STUDY

Figure 3 provides a simple case study illustrating the critical limitation of voting-based reward mechanisms and the subsequent benefit of our DREAM’s decoupled calibration. In this scenario, the policy model’s sampled predictions are systematically biased, with the incorrect answer ’4’ appearing most frequently. A conventional majority voting approach, as employed in TTRL, would consequently assign a positive reward exclusively to the erroneous majority answer while penalizing all other candidates, including the correct answer ’2’. This creates a perverse incentive structure, initiating a vicious cycle of error accumulation where the policy is guided to reinforce its own most frequent mistake. The resulting reward signal is highly misleading, as evidenced by a low “Reward Hit Rate” of 40%, which provides a poor and often detrimental gradient for policy optimization.

In contrast, DREAM effectively mitigates this failure mode by decoupling reward estimation from the policy model’s consensus. By incorporating an independent reliability score from the judge model, the final calibrated reward corrects for the bias of majority voting. As shown, the correct but infrequent answers ’2’ is able to receive significant reward scores (0.9 and 0.7) despite their zero voting-based reward. This crucial step recovers the learning signal for the correct reasoning path, which would have otherwise been lost. Consequently, the calibrated reward structure is far more aligned with the ground truth, increasing the effective hit rate to 60% and guiding the reinforcement learning process toward a more accurate policy, rather than merely a more confident one.

Furthermore, the strength of this decoupled approach is not over-rely on a perfectly omniscient judge model. Even a moderately effective judge, one that is not flawless but is better at evaluation than the policy model is at generation, provides substantial benefits. The key advantage lies in introducing a source of evaluation whose errors are likely uncorrelated with the policy model’s systemic biases. For instance, even if a moderately accurate judge assigned a slightly lower score to the correct answer ’2’ (e.g., 0.6, even 0.4) and a small non-zero score to an incorrect answer ’3’ (e.g., 0.2 or 0.1), the final calibrated reward for ’2’ would still be substantially higher than that for other incorrect, non-majority answers like ’1’ and ’3’. The judge’s primary role is thus to break the “tyranny of the majority” and create a more nuanced relative reward landscape. This allows group-wise optimization methods like GRPO to better distinguish between varying degrees of correctness among the candidates, preventing the policy from collapsing into a confident but fundamentally flawed state.

## C THE USE OF LARGE LANGUAGE MODELS

In this manuscript, we use LLMs to help polish writing at the sentence level and check grammar.