RANGE-GRPO: POLICY OPTIMIZATION VIA PAIRWISE RELATIONS AMONG REWARD INTERVALS

Anonymous authors
Paper under double-blind review

ABSTRACT

As the use of large language models (LLMs) expands, post-training has become increasingly important for adapting them to downstream tasks. However, obtaining reliable supervision remains costly, especially in domains without reference answers or executable verifiers. LLM-as-a-Judge provides scalable pseudo-rewards for unlabeled responses, but a single point score does not explicitly represent reward uncertainty. This motivates representing pseudo-rewards as conformally calibrated reward ranges. We propose **Range-GRPO**, a semi-supervised post-training framework that combines limited labeled data with unlabeled prompts. In Group Relative Policy Optimization (GRPO), learning signals depend on relative reward comparisons within each rollout group. The proposed objective compares reward ranges pairwise rather than reducing them to point rewards, allowing interval uncertainty to affect both the magnitude and direction of these signals. Our theoretical analysis characterizes this distinction and shows that the proposed objective recovers the Dr.GRPO advantage when all reward ranges collapse to points. Empirically, Range-GRPO achieves the highest in-distribution and out-of-distribution average performance among the evaluated semi-supervised methods while using a smaller total response budget than the compared baselines.

1 INTRODUCTION

Adapting large language models (LLMs) to new tasks and domains requires reliable supervision, yet obtaining it at scale remains difficult. Existing benchmarks provide limited coverage of task distributions (Kiela et al., 2021) and may suffer from data leakage (Roberts et al., 2024; White et al., 2025). These limitations motivate semi-supervised post-training methods that use limited labeled data to learn from large collections of unlabeled prompts. LLM-as-a-Judge (LaaJ) offers a scalable way to assign pseudo rewards to unlabeled responses (Lee et al., 2024; Bai et al., 2022). However, a judge may assign high scores to flawed reasoning and low scores to valid but atypical responses. It may also react to surface cues unrelated to correctness (Ye et al., 2025; Zheng et al., 2023; Wu & Aji, 2025), but a single scalar score does not express the uncertainty of that evaluation. Conformal calibration offers a promising approach to representing this uncertainty through reward intervals (Sheng et al., 2025). In this setting, it measures the discrepancies between predicted and reference rewards on a small labeled calibration set and uses the empirical error distribution to construct intervals for new responses. Under exchangeability, these intervals cover the reference reward with marginal coverage of at least $1-\alpha$ (Angelopoulos & Bates, 2023). This guarantee provides statistical support for the calibrated intervals, but does not specify how to construct effective policy learning signals from them.

We therefore seek an objective that uses calibrated intervals directly to determine relative credit among responses. Some existing approaches predict task outputs as intervals and derive rewards by comparing these intervals with known reference values (Lin et al., 2026; Liu et al., 2026; Wu et al., 2026). These comparisons cannot yield rewards for unlabeled responses without the corresponding reference information. More closely related, Uncertainty-Aware Reward Modeling (UARM) follows a reweighting approach for Group Relative Policy Optimization (GRPO) (Shao et al., 2024), using the widths of calibrated intervals to scale advantages computed from point rewards (Pan et al., 2026). However, this rescaling can adjust advantage magnitudes but cannot change the direction determined by the point rewards, even when those rewards provide a misleading learning signal.

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.

We address this gap with **Range-GRPO**, a semi-supervised post-training framework that learns from unlabeled prompts using conformally calibrated reward intervals. The proposed objective aggregates pairwise midpoint differences into group-relative advantages, using the widths of both reward intervals to weight the contribution of each comparison. By changing the balance between positive and negative pairwise contributions, this weighting can alter both advantage magnitudes and signs even when the interval midpoints remain fixed. When all reward intervals collapse to points, this objective recovers the credit assignment of Dr.GRPO as a special case (Liu et al., 2025).

As the policy learns, its response distribution can drift from the initial labeled calibration distribution, motivating weighted conformal inference to account for this shift. To construct reward intervals under this drift, we adapt Robust Weighted Conformal Inference (RoWCI), which performs shift-aware LaaJ calibration in a low-cardinality judgment state space (Noh et al., 2026).

Our main contributions are as follows:

- We propose **Range-GRPO**, the first semi-supervised GRPO framework to construct on-policy learning signals through pairwise comparisons of conformal reward intervals.
- We theoretically show that pairwise interval comparisons exploit range information beyond scalar reduction to determine both the magnitude and direction of learning signals, while retaining Dr.GRPO as the point-reward special case.
- Empirically, Range-GRPO outperforms the evaluated range-based methods on the synthetic task. On real reasoning benchmarks, it achieves the highest ID and OOD average accuracies among the evaluated semi-supervised methods, with relative improvements of **5.2%** and **7.0%** over the strongest baseline, respectively.

2 Related work

2.1 Reward Supervision

Reinforcement learning with verifiable rewards (RLVR) typically derives rewards from ground truth answers (Guo et al., 2025; Ma et al., 2025) or executable verifiers. Prior work also combines outcome rewards with signals derived from internal model states (Sun et al., 2026). Methods based on majority voting use agreement among unlabeled rollouts to construct pseudo labels (Wei et al., 2025; Zuo et al., 2025; Yang et al., 2026). Another line of work uses LaaJ to assign rewards according to explicit rubrics (Zhou et al., 2026; Gunjal et al., 2026). TraPO, a semi-supervised RLVR framework, uses a small labeled set to select unlabeled samples whose learning trajectories align with those of labeled samples (Yang et al., 2026). Together, these approaches broaden the scope of available reward supervision beyond direct outcome verification. However, consensus rewards may reinforce incorrect answers, while point rewards from LLM judges may reflect biases unrelated to response quality. Range-GRPO uses judge estimates with limited labeled calibration data to construct conformal reward intervals that represent reward uncertainty on unlabeled rollouts.

2.2 Conformal Calibration

Conformal calibration is a promising approach to quantifying uncertainty in LLM judge evaluations (Sheng et al., 2025). It constructs prediction intervals from labeled calibration data and provides marginal coverage guarantees under exchangeability (Angelopoulos & Bates, 2023). However, policy updates can shift the response distribution away from the calibration distribution, potentially violating the exchangeability assumption underlying these guarantees. Weighted Conformal Inference (WCI) addresses covariate shift by reweighting calibration samples using density ratios (Tibshirani et al., 2019). RoWCI, which we adopt, mitigates weight instability due to nuisance variation in LLM judge inputs (Noh et al., 2026). Recent studies have integrated conformal prediction into LLM training. CONST selects prompts for annotation based on conformal prediction set size (Xiao et al., 2026). CFA uses conformal prediction sets to estimate answer reliability and reweight preference pairs while retaining the given preference labels (Chen et al., 2026). UARM uses calibrated reward interval widths to rescale GRPO advantages derived from point rewards. However, its interval calibration procedure does not explicitly correct for shifts between the calibration and response distributions (Pan et al., 2026). In contrast, Range-GRPO accounts for these shifts when constructing conformal reward intervals for unlabeled rollouts.

2.3 RANGE-BASED LEARNING SIGNALS

Several studies derive scalar rewards by comparing predicted ranges with task targets. BAGEN rewards narrow ranges that contain the realized cost of the remaining trajectory (Lin et al., 2026). REC-RL rewards predicted count ranges that contain the target, assigning larger rewards as both bounds approach it (Liu et al., 2026). TempR1 uses the overlap between predicted and ground-truth temporal intervals as a reward signal (Wu et al., 2026). Other studies use ranges to account for uncertainty in training signals. In its interval setting, Variance-Aware PPO samples scalar rewards uniformly from a predicted reward range and uses the corresponding variance to regularize policy updates (Banerjee et al., 2025). IRPM-Interval constructs Student's t confidence intervals over chosen and rejected score groups to define binary rollout rewards (Song et al., 2026). UARM uses the median quantile as a point reward and the width of a conformally calibrated reward range to reweight response advantages without changing their signs (Pan et al., 2026). Range-GRPO instead assigns relative credit within each rollout group through pairwise comparisons of conformal reward intervals, allowing reward uncertainty to influence both advantage magnitudes and signs.

3 PRELIMINARIES

3.1 GROUP RELATIVE POLICY OPTIMIZATION

Group Relative Policy Optimization (GRPO) is a critic-free variant of Proximal Policy Optimization (PPO) that estimates advantages by comparing the rewards of multiple responses to the same prompt (Shao et al., 2024). For a prompt $q \sim P_Q$, the old policy samples G responses $o_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$. Under outcome supervision, GRPO computes $A_i = (r_i - \bar{r})/\sigma_r$ from the response rewards $r_1, \ldots, r_G$, where $\bar{r} = G^{-1}\sum_j r_j$ and $\sigma_r = \text{std}(r_1, \ldots, r_G)$. For token k, let $\rho_{i,k} = \pi_\theta(o_{i,k} \mid q, o_{i,<k})/\pi_{\theta_{\text{old}}}(o_{i,k} \mid q, o_{i,<k})$. GRPO applies the same response-level advantage A_i to all tokens in o_i and maximizes

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{k=1}^{|o_i|}\left\{\min\big(\rho_{i,k}A_i,\ \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon)A_i\big) - \beta D_{i,k}^{\text{KL}}\right\}\right]. \quad (1)$$

In Eq. (1), $D_{i,k}^{\text{KL}}$ penalizes deviation from a fixed reference policy π_{ref}, while ϵ and β control the clipping range and KL regularization strength, respectively. Since A_i determines the direction and magnitude of the learning signal for o_i, the quality of advantage estimates directly affects policy updates. We build on Dr.GRPO, a variant of GRPO that computes A_i without normalization by σ_r and removes response length normalization from the objective (Liu et al., 2025). For convenience, Appendix A summarizes the notation used throughout the paper.

3.2 REWARD RANGE CONSTRUCTION

We adapt Robust Weighted Conformal Inference (ROWCI) to construct reward ranges for responses generated by the current policy when reference rewards are unavailable (Noh et al., 2026). At a fixed optimization step t, let $X = (q, o)$, where $o \sim \pi_{\theta_t}(\cdot \mid q)$, and let $Y \in [0, 1]$ denote the corresponding reference reward. Let P_{src} and P_t denote the source and target distributions over (X, Y), respectively. We observe both X and Y in the source data but only X in rollouts from π_{θ_t}. Following RoWCI, we represent X using a low-cardinality judgment state $Z = \phi(X)$ to reduce sensitivity to surface cues in LaaJ evaluations. When some judgment-state components depend on relationships with other responses in the same rollout group, the state extractor additionally conditions on the corresponding group. We suppress this group-context argument in the notation $\phi(X)$. We perform distribution-shift reweighting in the judgment-state space and assume $P_{\text{src}}(Y \mid Z) \approx P_t(Y \mid Z)$ and $P_t^Z \ll P_{\text{src}}^Z$, where P_{src}^Z and P_t^Z denote the source and target distributions of Z, respectively.

Given a fixed state scorer $s_Z : \mathcal{Z} \to [0, 1]$, we compute the nonconformity score $E_j^Z = |s_Z(Z_j) - Y_j|$ for each source calibration pair (X_j, Y_j), where $Z_j = \phi(X_j)$. A binary domain classifier outputs $\widehat{p}_t^{\text{tgt}}(z) \approx \Pr(C = 1 \mid Z = z)$, where $C = 0$ and $C = 1$ denote the source and target domains, respectively. We compute the calibration weights as $\widehat{w}_t(z) = \frac{m_{\text{src}}}{m_t}\frac{\widehat{p}_t^{\text{tgt}}(z)}{1-\widehat{p}_t^{\text{tgt}}(z)}$, where m_{src} and m_t are

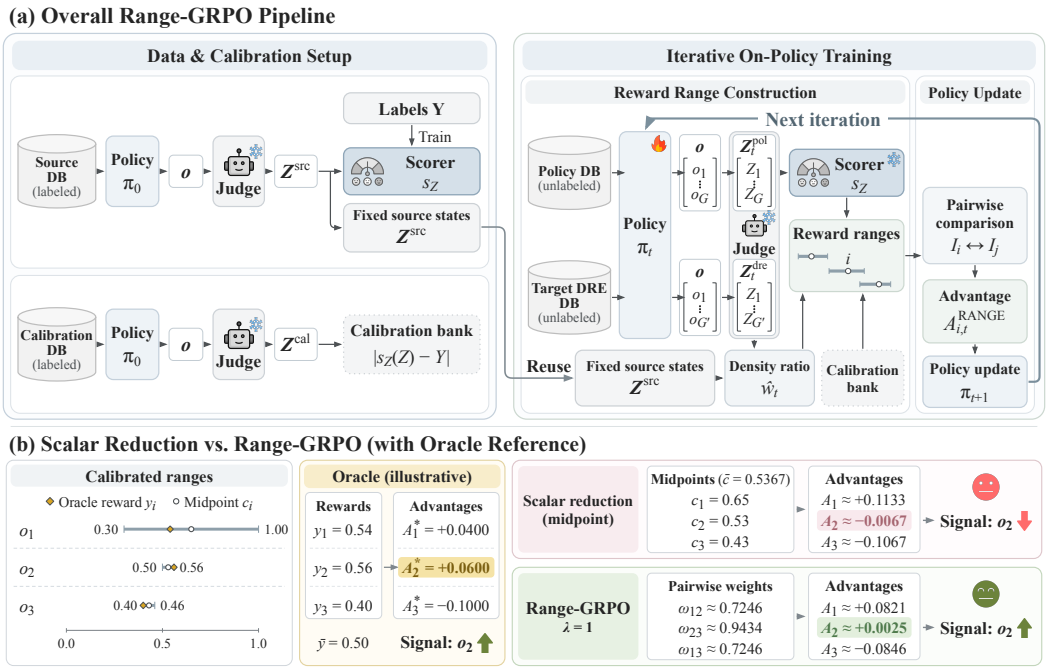


Figure 1: **Overview of Range-GRPO and its learning signals. (a)** At initialization, Range-GRPO fits a state scorer on labeled source rollouts and computes nonconformity scores on calibration rollouts. Each iteration generates policy and target DRE rollouts with G and G' responses, respectively, and updates density ratios using the target states and reused source states. The density ratios, fixed scorer, and calibration scores are used to construct reward ranges, from which pairwise interval comparisons construct the advantages for policy updates. **(b)** An illustrative comparison uses identical reward intervals for both methods. Although the interval for o_2 lies above that for o_3 with the same width, midpoint reduction assigns it a negative advantage under Dr.GRPO. Range-GRPO instead downweights pairwise contributions involving wider, more uncertain intervals, turning this advantage positive and aligning the direction of the learning signal with the oracle preference.

the numbers of source and target samples used to train the domain classifier. The reward range is constructed as $\widehat{\mathcal{I}}_{\alpha,t}(X) = \big[s_Z(\phi(X)) - \widehat{\tau}_{\alpha,t}(\phi(X)),\ s_Z(\phi(X)) + \widehat{\tau}_{\alpha,t}(\phi(X))\big] \cap [0,1]$, where $\widehat{\tau}_{\alpha,t}(z)$ is the weighted conformal cutoff for nominal marginal coverage $1-\alpha$. For the weighted cutoff, we include a point mass at $+\infty$ with weight $\widehat{w}_t(z)$ for the new response. In the main experiments, we instead normalize the nonconformity scores to account for variation in reward uncertainty across scorer outputs, as described in Section C.3. Range-GRPO constructs a reward interval for each of the G responses to a prompt.

4 Method

We formulate the Range-GRPO objective and characterize its pairwise credit assignment relative to reducing each reward range to a point reward at a fixed relative position. Range-GRPO recovers Dr.GRPO when all reward ranges collapse to points. Finally, we describe the on-policy training procedure that integrates this objective with shift-aware conformal reward calibration, as illustrated in Figure 1.

4.1 Range-GRPO Objective

At iteration t, the current policy $\pi_t := \pi_{\theta_t}$ generates G responses $o_i \sim \pi_t(\cdot \mid q)$ for each prompt q. Each response is associated with a calibrated reward interval $\mathcal{I}_i := \widehat{\mathcal{I}}_{\alpha,t}(X_i) = [L_i, U_i]$, whose center and half-width are $c_i = (L_i + U_i)/2$ and $h_i = (U_i - L_i)/2$, respectively. Below, we suppress the dependence on q, t, and α, with i indexing the response of interest and j a comparison response.

Definition (Range-GRPO objective)**.** For each pair of responses $i \neq j$, define the comparison weight $\omega_{ij} = \frac{\lambda}{h_i+h_j+\lambda}$, where $\lambda > 0$. The corresponding pairwise signal is $g_{ij} = \omega_{ij}(c_i - c_j)$. Range-GRPO aggregates these signals into the response-level advantage

$$A_i^{\mathrm{RANGE}} = \frac{1}{G}\sum_{j\neq i} g_{ij} = \frac{1}{G}\sum_{j\neq i} \omega_{ij}(c_i - c_j). \tag{2}$$

We define the Range-GRPO objective by replacing A_i with A_i^{RANGE} in Eq. (1) and removing the response length normalization factor $1/|o_i|$.

The center difference $c_i - c_j$ determines the direction of the pairwise signal, while ω_{ij} scales its magnitude. We use reward interval width as a proxy for uncertainty in the reward estimate. Importantly, the reliability of a pairwise comparison depends on the uncertainty of both responses. A wide interval for either response reduces confidence in the difference between their centers, even when the other interval is narrow. We therefore define ω_{ij} using the combined half-width $h_i + h_j$, so that uncertainty in either response attenuates the corresponding pairwise contribution. The reference width λ controls the sensitivity of this attenuation to interval width.

Proposition 1 (Range-GRPO under Zero Interval Widths)**.** *Suppose $\mathcal{I}_i = [r_i, r_i]$ for all $i = 1, \ldots, G$. Then $\omega_{ij} = 1$ for every pair, and*

$$A_i^{\mathrm{RANGE}} = \frac{1}{G}\sum_{j\neq i}(r_i - r_j) = r_i - \bar{r} = A_i^{\mathrm{Dr.GRPO}}. \tag{3}$$

Thus, Dr.GRPO is a special case of Range-GRPO when all reward intervals collapse to points.

4.2 Preserving Range Information through Pairwise Credit Assignment

A point reduction maps each reward interval $\mathcal{I}_i = [L_i, U_i]$ to a single scalar reward. For a fixed relative position $\xi \in [0, 1]$, define $r_i^{\mathrm{pt}} = \xi L_i + (1-\xi)U_i = c_i + (1-2\xi)h_i$. The lower bound (LB), midpoint, and upper bound (UB) correspond to $\xi = 1$, $\xi = 1/2$, and $\xi = 0$, respectively. Since different combinations of c_i and h_i can yield the same r_i^{pt}, the scalar reward alone does not uniquely determine the underlying interval. This limitation becomes explicit under midpoint reduction, where $r_i^{\mathrm{pt}} = c_i$ is independent of h_i.

We examine how this information loss affects group-relative credit assignment. For a rollout group $\mathcal{G}$, let $A_i^{\mathrm{pt}}(\mathcal{G}) = r_i^{\mathrm{pt}} - \bar{r}^{\mathrm{pt}}(\mathcal{G})$, where $\bar{r}^{\mathrm{pt}}(\mathcal{G})$ denotes the mean point reward in $\mathcal{G}$. In the following, we isolate the effect of interval width by holding midpoint rewards fixed.

Proposition 2 (Effect of Pairwise Range Information on Credit Assignment)**.** *Let the point rewards be interval midpoints ($\xi = 1/2$) and consider two rollout groups with $G = 3$, $\mathcal{G}_j = \{i, \ell, j\}$ and $\mathcal{G}_k = \{i, \ell, k\}$, where $r_j^{\mathrm{pt}} = r_k^{\mathrm{pt}} > r_i^{\mathrm{pt}} > r_\ell^{\mathrm{pt}}$ and $h_j < h_k$. Suppose $|g_{ik}| < g_{i\ell} < |g_{ij}|$. Then*

$$\begin{aligned}
&(1) \quad A_i^{\mathrm{RANGE}}(\mathcal{G}_j) < 0 < A_i^{\mathrm{RANGE}}(\mathcal{G}_k),\\
&(2) \quad A_i^{\mathrm{pt}}(\mathcal{G}_j) = A_i^{\mathrm{pt}}(\mathcal{G}_k),\\
&(3) \quad \operatorname{sign}\big(A_i^{\mathrm{wt}}(\mathcal{G}_j)\big) = \operatorname{sign}\big(A_i^{\mathrm{wt}}(\mathcal{G}_k)\big),
\end{aligned}$$

where $A_i^{\mathrm{wt}}(\mathcal{G}) = \eta_i(\mathcal{G}) A_i^{\mathrm{pt}}(\mathcal{G})$ with $\eta_i(\mathcal{G}) > 0$.

Proposition 2 shows that the advantage sign for response i under Range-GRPO can reverse as interval widths change, even when the midpoint rewards remain fixed. For response i, $c_j = c_k > c_i > c_\ell$ implies $g_{i\ell} > 0$ and $g_{ij}, g_{ik} < 0$. Since $h_j < h_k$, the wider interval of k yields $\omega_{ik} < \omega_{ij}$, thereby reducing the magnitude of its negative pairwise contribution.

The resulting advantages of response i are $A_i^{\mathrm{RANGE}}(\mathcal{G}_j) = \frac{1}{G}(g_{i\ell} + g_{ij})$ and $A_i^{\mathrm{RANGE}}(\mathcal{G}_k) = \frac{1}{G}(g_{i\ell} + g_{ik})$. The assumed inequality gives $g_{i\ell} + g_{ij} < 0 < g_{i\ell} + g_{ik}$, establishing result (1). Under midpoint reduction, j and k retain the same point reward despite having different interval widths, yielding the same point-reduction advantage in result (2). Finally, $\eta_i(\mathcal{G}_j)$ and $\eta_i(\mathcal{G}_k)$ are both positive. Response-level reweighting therefore preserves the sign of the common point-reduction advantage, so result (3) follows.

Algorithm 1 The Range-GRPO Algorithm

Input: Prompt sets $\mathcal{D}_U^{\mathrm{pol}}, \mathcal{D}_L^{\mathrm{src}}, \mathcal{D}_L^{\mathrm{cal}}, \mathcal{D}_U^{\mathrm{dre}}$; initial policy π_0; frozen judge J
Hyperparameters: miscoverage rate α; reference width λ; group size G; number of training iterations T
Stage 1: Data and Calibration Setup
Generate source rollouts from $\mathcal{D}_L^{\mathrm{src}}$ and calibration rollouts from $\mathcal{D}_L^{\mathrm{cal}}$ using π_0
Assign binary rewards $\mathcal{Y}^{\mathrm{src}}$ and $\mathcal{Y}^{\mathrm{cal}}$ based on answer correctness
Extract source states $\mathcal{Z}^{\mathrm{src}}$ and calibration states $\mathcal{Z}^{\mathrm{cal}} = \{Z_j^{\mathrm{cal}}\}_{j=1}^{n_{\mathrm{cal}}}$ with J
Fit the state scorer $s_Z : \mathcal{Z} \to [0,1]$ on $(\mathcal{Z}^{\mathrm{src}}, \mathcal{Y}^{\mathrm{src}})$
Compute calibration nonconformity scores $\{E_j^Z\}_{j=1}^{n_{\mathrm{cal}}}$ as defined in Section 3.2
Fix s_Z and retain $\mathcal{Z}^{\mathrm{src}}$ for DRE and $\{(Z_j^{\mathrm{cal}}, E_j^Z)\}_{j=1}^{n_{\mathrm{cal}}}$ for calibration throughout training
for $t = 0, \ldots, T-1$ **do**
Stage 2: On-Policy Reward Range Construction
    Sample a policy prompt batch $\mathcal{Q}_t$ from $\mathcal{D}_U^{\mathrm{pol}}$
    Generate G rollouts per prompt in $\mathcal{Q}_t$ using π_t
    Sample fresh target DRE rollouts from π_t on $\mathcal{D}_U^{\mathrm{dre}}$
    Extract $\mathcal{Z}_t^{\mathrm{pol}}$ and $\mathcal{Z}_t^{\mathrm{dre}}$ with J
    Fit a binary domain classifier on $\mathcal{Z}^{\mathrm{src}}$ and $\mathcal{Z}_t^{\mathrm{dre}}$ to estimate the target-to-source density ratio $\widehat{w}_t(\cdot)$
    for each policy rollout X_i with state $Z_{i,t}^{\mathrm{pol}}$ **do**
        Compute the weighted conformal cutoff $\widehat{\tau}_{\alpha,t}(Z_{i,t}^{\mathrm{pol}})$ as defined in Section 3.2
        $\widehat{\mathcal{I}}_{\alpha,t}(X_i) \leftarrow \left[s_Z(Z_{i,t}^{\mathrm{pol}}) - \widehat{\tau}_{\alpha,t}(Z_{i,t}^{\mathrm{pol}}), s_Z(Z_{i,t}^{\mathrm{pol}}) + \widehat{\tau}_{\alpha,t}(Z_{i,t}^{\mathrm{pol}})\right] \cap [0,1]$
    end for
Stage 3: Range-GRPO Policy Update
    Compute the advantage $A_{i,t}^{\mathrm{RANGE}}$ for each rollout within each prompt group using Eq. (2)
    Update π_t by optimizing the Range-GRPO objective on the policy rollouts to obtain π_{t+1}
end for
Output: final policy π_T

UARM (Pan et al., 2026) provides a concrete example of the response-level weighting in statement (3). For a matched comparison with Range-GRPO, we use the same reward intervals and set $r_i = c_i$ for every response. The UARM reweighting rule then gives

$$A_i^{\mathrm{UARM}}(\mathcal{G}) = \eta_i^{\mathrm{UARM}} A_i^{\mathrm{pt}}(\mathcal{G}), \qquad \eta_i^{\mathrm{UARM}} = \frac{\sigma_{\mathrm{signal}}}{\sigma_{\mathrm{signal}}^2 + \sigma_{\mathrm{noise},i}^2} > 0. \tag{4}$$

The scaling factor can be written as $\frac{\sigma_{\mathrm{signal}}^2}{\sigma_{\mathrm{signal}}^2+\sigma_{\mathrm{noise},i}^2}\frac{1}{\sigma_{\mathrm{signal}}} = \eta_i^{\mathrm{UARM}} > 0$, since $\sigma_{\mathrm{signal}} > 0$. The positive scaling factor can change the advantage magnitude, but the centered midpoint reward still determines its sign. Under this matched comparison, UARM satisfies statement (3). In contrast, Range-GRPO applies interval-dependent weights to the individual pairwise signals before aggregation, which allows the advantage sign to change as shown in Proposition 2.

4.3 On-Policy Training Procedure

Range-GRPO combines the proposed objective with the reward range construction in Section 3.2 for on-policy training. At initialization, it fits the state scorer and computes nonconformity scores using rollouts from the initial policy. Each iteration then generates fresh policy and target DRE rollouts, updates the density ratio weights, and constructs reward ranges for the policy update.

Calibration Setup. We use four disjoint sets of prompts: an unlabeled policy set $\mathcal{D}_U^{\mathrm{pol}}$, a labeled source set $\mathcal{D}_L^{\mathrm{src}}$, a labeled calibration set $\mathcal{D}_L^{\mathrm{cal}}$, and an unlabeled target set $\mathcal{D}_U^{\mathrm{dre}}$ for density ratio estimation (DRE). The source and calibration sets come from the same labeled prompt distribution, whereas the policy and target DRE sets come from the same unlabeled prompt distribution.

The initial policy π_0 generates rollouts from $\mathcal{D}_L^{\mathrm{src}}$ and $\mathcal{D}_L^{\mathrm{cal}}$. Let ϕ denote the fixed state-extraction procedure implemented by the frozen judge J, so that each rollout X_i is mapped to a judgment state $Z_i = \phi(X_i)$. The same state-extraction procedure is applied to all four sets throughout training. Each source and calibration rollout receives a binary reward $Y_i \in \{0,1\}$ based on answer correctness. We fit the state scorer s_Z using the source states and rewards and compute calibration nonconformity scores as described in Section 3.2. The state scorer, source and calibration states, and calibration nonconformity scores remain fixed throughout training.

Table 1: **Range-GRPO achieves the highest mean test accuracy among the compared range-based methods.** We select the checkpoint with the highest validation accuracy over ten training epochs and report its test accuracy. All values are means ± sample standard deviations across three seeds. Cov. and Width summarize empirical coverage and mean interval width on unlabeled responses over the full training phase. We set the miscoverage rate to $\alpha = 0.1$. Bold and underlined values indicate the best and second best results, respectively, for accuracy and interval width.

Advantage	Method	Acc. (%) ↑	Cov. (%)	Width ↓
		Scalar reductions		
GRPO	LB	27.63 ± 2.84	93.69 ± 0.65	0.3942 ± 0.0098
	Midpoint	23.97 ± 7.61	93.95 ± 1.47	0.4070 ± 0.0120
	UB	24.60 ± 2.71	92.64 ± 0.89	0.3930 ± 0.0122
Dr.GRPO	LB	30.33 ± 7.68	93.29 ± 0.27	0.3753 ± 0.0126
	Midpoint	$\underline{31.77 \pm 14.11}$	93.02 ± 0.11	$\underline{0.3699 \pm 0.0094}$
	UB	29.83 ± 4.84	93.35 ± 0.75	0.3785 ± 0.0136
		Main comparison		
UARM		20.43 ± 3.61	94.41 ± 0.23	0.4311 ± 0.0090
Range-GRPO (Ours)		$\mathbf{39.40 \pm 7.70}$	93.05 ± 0.70	$\mathbf{0.3595 \pm 0.0206}$

On-Policy Training. At iteration t, the current policy π_t and the unlabeled prompt distribution induce the target distribution P_t. We sample a batch of prompts from $\mathcal{D}_U^{\text{pol}}$ and generate G rollouts per prompt. Using the same policy and sampling settings, we also generate fresh rollouts from the fixed set $\mathcal{D}_U^{\text{dre}}$. The judge J extracts states from both sets of rollouts, but we exclude the target DRE rollouts from policy updates.

We fit a binary domain classifier on the fixed source states and the current target DRE states, without using correctness rewards, to update the estimated state density ratios $\widehat{w}_t$. Using these weights with the fixed calibration states and nonconformity scores, we construct a reward range $\widehat{\mathcal{I}}_{\alpha,t}(X_i)$ for each policy rollout X_i. For each prompt, we compute advantages by comparing its G reward ranges pairwise using Eq. (2). We then optimize the Range-GRPO objective on the policy rollouts using these advantages to obtain π_{t+1}. Algorithm 1 summarizes the procedure.

5 Experiments

5.1 Toy Experiment

Setup. We adapt a synthetic task from prior work (Lee et al., 2026) into a candidate selection task. GPT-2-small (Radford et al., 2019) serves as the policy, and the prompt does not include the ranking rule. The rank of the selected candidate determines the true reward. Accuracy measures how often the policy selects the candidate with rank 1, which receives the highest reward. The labeled training pool serves only to fit the state scorer and calibrate reward intervals.

We compare Range-GRPO with LB, Midpoint, UB, and UARM. LB, Midpoint, and UB have separate variants using the GRPO (Shao et al., 2024) and Dr.GRPO (Liu et al., 2025) advantage formulas. All methods share the data allocation and use the same on-policy reward range construction described in Section 4.3. This controlled comparison focuses on how the methods derive learning signals from reward ranges. We use the true rewards of unlabeled responses only to compute empirical coverage.

All methods draw from the same labeled and unlabeled pools and have equal budgets for total rollout generation, including auxiliary generation. We train all methods for ten epochs with $G = 8$.

Results. Table 1 shows that Range-GRPO achieves the highest mean test accuracy (39.40%) among the compared methods, exceeding the best scalar reduction result by 7.63 percentage points. All methods achieve mean empirical coverage above the nominal 90% level on unlabeled responses over the full training phase. Range-GRPO also yields the smallest mean interval width (0.3595). Appendix B.2 provides additional training details and validation curves.

Table 2: **Range-GRPO achieves the highest ID and OOD averages among semi-supervised methods.** We report published baseline results from TraPO (Yang et al., 2026), while methods marked with * are evaluated under our experimental setting. For the reproduced TraPO configurations, values in parentheses indicate the entropy regularization coefficients. All training methods use 1,024 (1K) ID and 1,024 (1K) OOD prompts. Semi-supervised methods use labeled ID prompts and unlabeled OOD prompts, whereas the fully supervised model uses labels for all 2K prompts. GPQA† denotes GPQA-diamond. Bold and underlined scores indicate the best and second-best results among semi-supervised methods, respectively, with ties receiving the same formatting. The fully supervised result is provided for reference only and excluded from this ranking.

	In-Distribution						Out-of-Distribution			
Model / Method	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	**Avg.**	ARC-c	GPQA†	MMLU-Pro	**Avg.**
Backbone										
Qwen-Base	11.5/4.9	31.3	43.6	7.4	15.6	19.0	18.2	11.1	16.9	15.4
Qwen-Instruct	12.5/10.2	48.5	80.4	32.7	41.0	37.6	70.3	24.7	34.1	43.0
Fully supervised										
Supervised (2K)	17.3/12.4	56.8	81.4	38.6	44.8	41.9	82.0	38.9	52.4	57.8
Semi-supervised										
TTRL	16.4/13.6	49.9	66.9	26.5	37.8	35.2	62.0	31.8	43.5	45.8
Self-certainty	16.0/10.9	53.0	78.4	34.2	39.0	38.6	77.1	32.8	45.7	51.9
Token-level Entropy	17.7/11.0	51.7	77.0	33.1	41.0	38.6	76.5	30.8	44.7	50.7
Sentence-level Entropy	15.7/10.0	51.4	77.4	34.9	37.5	37.8	75.1	31.3	44.3	50.2
TraPO (entropy=0.001)*	17.7/12.2	55.9	79.2	**37.5**	41.3	40.6	80.4	34.8	43.5	52.9
UARM*	17.1/9.5	53.2	80.4	33.5	40.9	39.1	78.9	36.9	34.2	50.0
Range-GRPO (ours), $\lambda = 1$	**20.6/14.4**	**57.0**	**83.0**	33.8	**47.3**	**42.7**	**80.9**	**40.4**	**48.4**	**56.6**

5.2 Main Experiments

We evaluate Range-GRPO on established reasoning benchmarks to complement the synthetic toy experiment. These experiments examine whether Range-GRPO can effectively learn from unlabeled prompts with limited labeled supervision and generalize beyond the labeled source domain. Appendix C provides implementation details.

Baselines. We adapt the training and evaluation setting used for Table 2 in Yang et al. (2026), without increasing the data or policy rollout budgets. The comparison includes the semi-supervised results reported in that table for TTRL (Zuo et al., 2025), Self-certainty (Zhao et al., 2026), and the Token-level Entropy and Sentence-level Entropy baselines (Agarwal et al., 2025). We additionally reproduce TraPO under our experimental setting and implement UARM (Pan et al., 2026) with a quantile reward model as a direct baseline.

Training Setup. For policy training, we use Qwen2.5-Math-7B (Yang et al., 2024) and generate $G = 8$ responses per prompt. The training data comprise 1,024 labeled mathematics prompts from OpenR1-Math-220k (Hugging Face, 2025) and 1,024 unlabeled MMLU-Pro prompts (Wang et al., 2024). For Range-GRPO and UARM, we reserve part of the training data for their auxiliary components, including reward estimation, calibration, and density ratio estimation (DRE) where applicable. Training follows the TraPO schedule of ten labeled warmup epochs followed by five epochs using both labeled and unlabeled prompts. Labeled updates use binary correctness rewards based on reference answers, while each method defines its own learning signals for unlabeled updates.

Reward Range Construction. After labeled warmup and before unlabeled policy updates begin, we use the resulting policy to generate the source and calibration responses for reward range construction. A frozen Qwen3-4B-Instruct (Yang et al., 2025) judge extracts judgment states for these responses. We then fit a lightweight XGBoost scorer on the source states and compute calibration nonconformity scores. The scorer and calibration scores remain fixed during subsequent training, while density ratios are updated at each iteration using fresh target responses from the current policy.

For each unlabeled response, we combine its scorer prediction with the weighted calibration scores to construct a calibrated reward range with $\alpha = 0.1$.

Evaluation. We evaluate performance on both in-distribution (ID) mathematical benchmarks and out-of-distribution (OOD) benchmarks outside the labeled mathematics domain. The ID benchmarks include AIME 2024, AIME 2025, AMC, MATH-500 (Lightman et al., 2024), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), whereas the OOD benchmarks include ARC-c (Clark et al., 2018), GPQA-diamond (Rein et al., 2024), and MMLU-Pro. We report avg@32 on AIME 2024, AIME 2025, and AMC, and pass@1 on the remaining benchmarks.

Quantitative Analysis. Table 2 shows that Range-GRPO achieves the highest ID and OOD averages among semi-supervised methods. It also achieves the best accuracy on eight of the nine individual benchmarks. Both averages exceed those of our TraPO reproduction with an entropy coefficient of 0.001. Notably, Range-GRPO attains this performance while reserving part of the same training prompt pool for scorer fitting, calibration, and density ratio estimation rather than policy optimization. Despite this auxiliary allocation, its total response budget, including both policy and judge outputs, is approximately 18.2% smaller than that of TraPO.

Detailed prompt allocation and response accounting are provided in Section C. Range-GRPO also achieves higher ID and OOD averages than UARM, which applies uncertainty weights at the response level. This result is consistent with the benefit of preserving interval-dependent pairwise structure when constructing learning signals. Range-GRPO also remains competitive with the fully supervised 2K setting, exceeding its ID average while remaining close on OOD. Together, these results support the effectiveness of Range-GRPO for semi-supervised post-training across the evaluated benchmarks.

Qualitative Analysis. To examine the practical benefit of preserving pairwise range information as analyzed in Section 4.2, we compare its credit assignment with scalar reduction in the Range-GRPO training logs. Table 3 illustrates how pairwise weighting based on interval widths can reverse the advantage sign relative to the scalar-reduction baseline on an unlabeled OOD training instance. We compare Range-GRPO with midpoint scalarization under Dr.GRPO, which achieves the second-highest mean test accuracy in Table 1. Both methods use the same eight responses and calibrated intervals. The midpoint baseline assigns R7 a negative advantage because its midpoint lies below the group mean. The six negative comparisons involve wider counterpart intervals than the positive comparison with R1. Range-GRPO therefore assigns smaller weights to those six comparisons, yielding a positive advantage for R7. This example suggests that interval-aware pairwise weighting can produce a learning signal better aligned with response correctness.

Table 3: **Uncertainty-aware pairwise weighting can reverse the learning signal for a correct response.** On this unlabeled OOD training instance, midpoint scalarization under Dr.GRPO assigns the correct response R7 a negative advantage because its midpoint is 0.7384, below the group mean of 0.7402. Range-GRPO (λ=1) instead yields a positive advantage by weighting pairwise comparisons according to interval uncertainty. Correctness labels are shown only for post-hoc interpretation.

Response	Correctness	Calibrated range
R1	Incorrect	$[0.0000, 0.1173]$
R2–R6, R8	Correct	$[0.7082, 1.0000]$
R7	Correct	$[0.4767, 1.0000]$

6 Conclusion

We present Range-GRPO, a semi-supervised post-training framework that learns from unlabeled prompts with limited labeled supervision. The proposed objective assigns relative credit to each response through pairwise comparisons of reward intervals. Our theoretical analysis shows that Range-GRPO includes Dr.GRPO as a special case under zero-width intervals, while its pairwise credit assignment allows range information to affect both the direction and magnitude of the learning signal. Empirically, Range-GRPO outperforms the other semi-supervised methods in both ID and OOD average performance and achieves the best result on eight of the nine individual benchmarks. Together, these findings highlight the value of using interval relations to determine not only the strength of a learning signal but also its direction. Beyond the evaluated tasks, future work could extend this framework to domains where reward supervision is uncertain and expert annotations are limited.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing to improve the clarity and readability of the manuscript. The authors developed the core ideas and designed the methodology. Generative AI tools were also used for code organization and cleanup. We reviewed and revised all AI-assisted text to ensure that it accurately reflects the intended technical content and claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=UfFTBEsLgI.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Debangshu Banerjee, Kintan Saha, and Aditya Gopalan. Towards reliable, uncertainty-aware alignment, 2025. URL https://arxiv.org/abs/2507.15906.

Tiejin Chen, Xiaoou Liu, Vishnu Nandam, Kuan-Ru Liou, and Hua Wei. Conformal feedback alignment: Quantifying answer-level reliability for robust llm alignment. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3561–3572, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Hugging Face. Open R1: A fully open reproduction of DeepSeek-R1, January 2025. URL https://github.com/huggingface/open-r1.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, pp. 4110–4124, 2021.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of*

the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 26874–26901. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24t.html.

Seonggyun Lee, Sungjun Lim, Seojin Park, Soeun Cheon, and Kyungwoo Song. Semi-supervised preference optimization with limited feedback. In *International Conference on Learning Representations*, volume 2026, pp. 51284–51313, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

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 *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Yuxiang Lin, Zihan Wang, Mengyang Liu, Yuxuan Shan, Longju Bai, Junyao Zhang, Xing Jin, Boshan Chen, Jinyan Su, Xingyao Wang, et al. Bagen: Are llm agents budget-aware? *arXiv preprint arXiv:2606.00198*, 2026.

Hui Liu, Yunlai Teng, Kunlong Bai, Pengfei Qi, Haotian Yan, Liang Li, and Junlan Feng. Rec-rl: Referring expression counting via gaussian and range-based reward optimization. *arXiv preprint arXiv:2605.16460*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025.

Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pp. 63–70, 2002.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun MA, and Wenhu Chen. General-reasoner: Advancing LLM reasoning across all domains. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pBFVoll8Xa.

Kangjun Noh, Ryunyi LEE, Heejin Jung, Young Jin Kim, and Kyungwoo Song. Reliable LLM-as-a-judge evaluation via transferable calibration. In *ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems*, 2026. URL https://openreview.net/forum?id=AIBoVLkc31.

Licheng Pan, Haocheng Yang, Haoxuan Li, Yichen Sun, Yunsheng Lu, Shijian Wang, Lei Shen, Yuan Lu, Zhixuan Chu, and Hao Wang. Uncertainty-aware reward modeling for stable rlhf. *arXiv preprint arXiv:2606.19818*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Manley Roberts, Himanshu Thakur, Christine Herlihy, Colin White, and Samuel Dooley. To the cutoff... and beyond? a longitudinal perspective on LLM data contamination. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=m2NVG4Htxs.

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

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. Analyzing uncertainty of LLM-as-a-judge: Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11286–11328, 2025.

Haonan Song, Qingchen Xie, Huan Zhu, Feng Xiao, Luxi Xing, Liu Kang, Fuzhen Li, Zhiyong Zheng, Feng Jiang, Ziheng Li, Kun Yan, Qingyi Si, Yanghua Xiao, Hongcheng Guo, and Fan Yang. IRPM: Intergroup relative preference modeling for pointwise generative reward models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=JuiHYauZNk.

Zhongxiang Sun, Qipeng Wang, Haoyu Wang, Xiao Zhang, and Jun Xu. Mechanistic detection and mitigation of hallucination in large reasoning models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0271e4dc4e81459121a07b9634d5e79d-Abstract-Conference.html.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. First SFT, second RL, third UPT: Continual improving multi-modal LLM reasoning via unsupervised post-training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HL1j92hb6z.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, et al. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 91595–91631, 2025.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 297–312, 2025.

Tao Wu, Li Yang, Gen Zhan, Yabin Zhang, Yiting Liao, Junlin Li, Deliang Fu, Li Zhang, and Limin Wang. TempR1: Improving temporal understanding of MLLMs via temporal-aware multi-task reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2756–2767, 2026.

Zhiping Xiao, Yusheng Zhao, Qixin Zhang, Jiaye Xie, Wanjia Zhao, Weizhi Zhang, Xiao Luo, Philip S. Yu, and Ming Zhang. Sample lottery: Unsupervised discovery of critical instances for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=76OZBE4Rb6.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shenzhi Yang, Guangcheng Zhu, Haobo Wang, Xing Zheng, Yingfan Ma, Zhongqi Chen, Bowen Song, Weiqiang Wang, Junbo Zhao, and Gang Chen. TraPO: A semi-supervised reinforcement learning framework for boosting LLM reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 115067–115102, 2026.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. In *International Conference on Learning Representations*, volume 2025, pp. 102351–102390, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, volume 2026, pp. 2548–2581, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for open-ended LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IM8ZHjhD0v.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=VuVhgEiu20.

A NOTATION

Table 4: **Notation used throughout the paper.** Symbols are grouped by their roles and ordered approximately by their first appearance in the paper.

Notation	Meaning
GRPO and policy optimization	
$q,\ P_Q$	Prompt and prompt distribution.
$G,\ o_i,\ o_{i,k}$	Rollout-group size, response i, and token k of response i.
$\pi_\theta,\ \pi_{\theta_{\text{old}}},\ \pi_{\text{ref}},\ \pi_t$	Parameterized policy, old rollout policy, fixed reference policy, and current policy $\pi_t := \pi_{\theta_t}$.
$r_i,\ \bar{r},\ \sigma_r$	Response reward, group-mean reward, and group reward standard deviation.
$A_i,\ A_i^{\text{Dr.GRPO}}$	GRPO and Dr.GRPO response-level advantages.
$\rho_{i,k},\ D_{i,k}^{\text{KL}}$	PPO likelihood ratio and token-level KL penalty.
$\epsilon,\ \beta$	PPO clipping parameter and KL regularization coefficient.
Reward-range construction	
$t,\ X=(q,o),\ Y$	Optimization iteration, prompt-response pair, and reference reward.
$P_{\text{src}},\ P_t$	Source and iteration-t target distributions over (X,Y).
$Z=\phi(X),\ \mathcal{Z}$	Low-cardinality judgment state, state-extraction mapping, and state space.
$P_{\text{src}}^Z,\ P_t^Z$	Source and target marginal distributions of the judgment state.
$s_Z,\ E_j^Z$	Fixed state scorer and absolute calibration nonconformity score.
$C,\ \widehat{p}_t^{\text{tgt}}(z)$	Domain indicator (0: source; 1: target) and estimated posterior $\Pr(C=1 \mid Z=z)$.
$\widehat{w}_t(z),\ m_{\text{src}},\ m_t$	Target-to-source density-ratio weight and source/target sample counts used for its estimation.
$\alpha,\ \widehat{\tau}_{\alpha,t}(z)$	Miscoverage level and calibrated reward-space radius.
$\widehat{\mathcal{I}}_{\alpha,t}(X)$	Calibrated reward interval for prompt-response pair X.
Range-GRPO credit assignment	
$\mathcal{I}_i=[L_i,U_i]$	Reward interval for response i, with lower and upper endpoints L_i and U_i.
$c_i,\ h_i$	Interval center $c_i=(L_i+U_i)/2$ and half-width $h_i=(U_i-L_i)/2$.
$\lambda,\ \omega_{ij}$	Reference-width parameter and pairwise reliability weight $\omega_{ij}=\lambda/(h_i+h_j+\lambda)$.
g_{ij}	Pairwise credit signal $g_{ij}=\omega_{ij}(c_i-c_j)$.
A_i^{RANGE}	Range-GRPO response-level advantage obtained by aggregating pairwise signals.
$\xi,\ r_i^{\text{pt}}$	Relative position for scalar interval reduction and the resulting point reward.
$\mathcal{G},\ \bar{r}^{\text{pt}}(\mathcal{G}),\ A_i^{\text{pt}}(\mathcal{G})$	Rollout group, its mean point reward, and point-reduction advantage.
$A_i^{\text{wt}}(\mathcal{G}),\ \eta_i(\mathcal{G})$	Generic positively reweighted point advantage and its scaling factor.
$\eta_i^{\text{UARM}},\ \sigma_{\text{signal}},\ \sigma_{\text{noise},i}$	UARM scaling factor and its signal/noise quantities.
Training procedure	
$\mathcal{D}_U^{\text{pol}},\ \mathcal{D}_L^{\text{src}},\ \mathcal{D}_L^{\text{cal}},\ \mathcal{D}_U^{\text{dre}}$	Unlabeled policy, labeled source, labeled calibration, and unlabeled target-DRE prompt sets.
$J,\ \mathcal{Y}^{\text{src}},\ \mathcal{Y}^{\text{cal}}$	Frozen judge and source/calibration reward collections.
$\mathcal{Z}^{\text{src}},\ \mathcal{Z}^{\text{cal}},\ n_{\text{cal}}$	Source and calibration state collections and number of calibration rollouts.
$\mathcal{Q}_t,\ \mathcal{Z}_t^{\text{pol}},\ \mathcal{Z}_t^{\text{dre}},\ T$	Policy-prompt batch, policy/DRE state collections at iteration t, and total number of training iterations.
Appendix-specific notation	
$Z_i=(z_{i1},\dots,z_{i5})$	Five-dimensional judgment state used in the main experiments.
$R_j,\ \psi(v),\ \widehat{q}_{\alpha,t}(Z)$	Normalized nonconformity score, residual-scale function, and weighted cutoff for normalized conformal calibration.
ζ	Signal-variance stabilizer used in the UARM baseline.

B TOY EXPERIMENT

B.1 DATA AND TRAINING

Task. Following SSPO (Lee et al., 2026), we construct a synthetic dataset using words from the Natural Language Toolkit (NLTK) word list (Loper & Bird, 2002). We define a task-specific ranking rule for the candidate words. Each prompt contains ten candidate words labeled A–J, and the policy selects one letter. We first sort the words by increasing length, preserving the input order for words of equal length. We then reorder only the three shortest words alphabetically, treating uppercase and lowercase letters alike. The remaining seven words retain their order. A selection is correct if $\mathrm{rank}(o) = 1$, while the reward is determined by $r(o) = (11 - \mathrm{rank}(o))/10$. In the example shown in Table 5, the final order begins with *nominee*, *savoy*, and *wanton*. Training, validation, and test sets have disjoint prompt compositions.

Data allocation. For policy training and auxiliary range construction, we use 1,200 synthetic prompts: 200 labeled and 1,000 unlabeled. We allocate these prompts to scorer fitting, calibration, density ratio estimation (DRE), and policy training. Table 6 summarizes this allocation together with the separate validation and test sets. All methods use the same prompt partitions.

Table 5: Example candidates with their ranks and hidden rewards.

Choice	Word	Length	Rank	Reward
A	intoxicating	12	8	0.3
B	transcendental	14	10	0.1
C	tenacious	9	5	0.6
D	nocturne	8	4	0.7
E	knickerbocker	13	9	0.2
F	savoy	5	2	0.9
G	hypodermic	10	6	0.5
H	blossomtime	11	7	0.4
I	nominee	7	1	1.0
J	wanton	6	3	0.8

Table 6: Prompt allocation, separating policy learning from range construction.

Split	Role	Prompts
Labeled	Scorer / source DRE	100
	Calibration	100
Unlabeled	Policy	900
	Target DRE	100
Validation	Model selection	400
Test	Evaluation	1,000
Total		**2,600**

Training and model selection. All methods train for ten epochs without warmup or labeled policy updates. Table 7 summarizes the training settings. We select learning rates from $\{1, 2, 4, 5\} \times 10^{-5}$ and KL coefficients from $\{0, 0.01, 0.1\}$. For Range-GRPO, we select $\lambda = 0.2$ from the grid $\{0.2, 0.4, 0.6, 0.8, 1\}$. We select the best checkpoint in each run based on validation accuracy, then choose hyperparameters using the mean of these best validation accuracies across three seeds.

Table 7: **Policy training configuration and selected hyperparameters.** (a) summarizes the shared training configuration for all methods. (b) lists the selected learning rate and KL coefficient for each method. We select the hyperparameters by mean best validation accuracy across three seeds. LR denotes the learning rate.

(a) Training configuration

Setting	Value
Model	GPT-2-small
Batch size	16
Responses per prompt	8
Epochs	10
Warmup epochs	0
Optimizer	AdamW
Clipping	0.2
Entropy coefficient	0.01
Checkpoint	Best validation accuracy
Seeds	6200, 6201, 6202

(b) Selected hyperparameters

Method	LR	KL
LB (GRPO)	4×10^{-5}	0.1
Midpoint (GRPO)	5×10^{-5}	0
UB (GRPO)	5×10^{-5}	0.1
LB (Dr.GRPO)	5×10^{-5}	0
Midpoint (Dr.GRPO)	5×10^{-5}	0
UB (Dr.GRPO)	5×10^{-5}	0
UARM	5×10^{-5}	0.01
Range-GRPO ($\lambda = 0.2$)	5×10^{-5}	0

B.2 Learning Signals

Advantage construction. All methods in this toy comparison use the interval construction procedure in Section B.3 but apply different rules to compute advantages from the resulting intervals. The scalar baselines use the lower bound (LB), upper bound (UB), or midpoint of each interval as the reward and compute either GRPO or Dr.GRPO advantages. Range-GRPO uses the objective defined in Section 4.1 with $\lambda = 0.2$ and no additional advantage normalization.

We adapt UARM (Pan et al., 2026) by using the midpoint of each interval as the reward estimate for the corresponding response. The shared interval construction procedure replaces the original quantile reward model so that all advantage rules use the same source of uncertainty. We retain the published advantage formula and use a signal variance stabilizer of 10^{-6}.

Training dynamics. Figure 2 shows the validation curves for each method at the selected hyperparameter settings. Range-GRPO shows a stronger upward trend in validation accuracy, particularly relative to the scalar baselines using GRPO advantages.

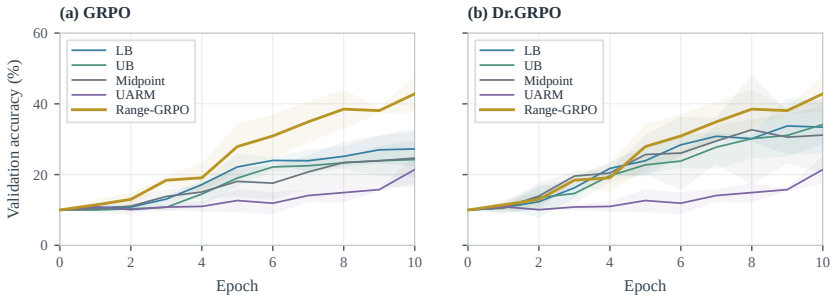


Figure 2: **Validation accuracy at the selected hyperparameter settings.** Curves show the mean $\pm$ sample standard deviation across three seeds. We show scalar baselines with GRPO and Dr.GRPO advantages separately and include the same UARM and Range-GRPO curves in both comparisons.

B.3 Range Construction

Response features. To simplify state construction in the toy task, we compute Z directly from word features rather than using an LLM judge. For each selected word, Z contains five fixed features: Brown corpus frequency, Brown genre count, WordNet synset count, WordNet noun status, and surprisal of the first token under GPT-2 after a fixed end-of-sequence context. We apply $\log(1 + x)$ to the first three features and use fixed standardization for all five.

Scorer and calibration scores. We adapt the RoWCI scorer (Noh et al., 2026) to the toy task using an XGBoost regressor that predicts rewards from Z. Using responses from the initial pretrained policy, we fit the scorer and compute calibration nonconformity scores on their respective prompt partitions. The fitted scorer and calibration scores remain fixed for the rest of training. All methods share the fitted scorer, calibration scores, and DRE source data.

Updates during training. At each update using unlabeled responses, a random forest estimates density ratios from a fixed source feature set and features extracted from new target responses sampled by each method's current policy on the shared target prompt pool. These ratios reweight the fixed calibration scores to account for changes in the response distribution without new target reward labels. For numerical stability, we cap the estimated density-ratio weights at their 95th percentile. The weighted cutoff also accounts for the weight of the query response. We clip intervals to $[0, 1]$ and use this full range when the cutoff is infinite. We exclude unlabeled rewards from policy optimization and use them for coverage diagnostics at a nominal level of 90%.

Response budget. Table 8(b) details the response budget for policy training, scorer fitting, calibration, and target DRE, excluding validation and test evaluations. Source DRE reuses the scorer responses and therefore requires no additional generations. Because we compute Z directly from word features, the toy experiment does not require an LLM judge.

Table 8: **Range construction settings and response budget.** (a) summarizes the range construction settings. (b) reports the response budget for policy training and range construction.

(a) Range construction settings

Setting	Value
Scorer	XGBoost regressor
Trees / depth	200 / 4
DRE	Random forest
Trees / depth	200 / 8
DRE target per update	255 responses
Refresh	Each unlabeled update
Nominal coverage	90%
Weight cap	95th percentile

(b) Response budget

Category	Responses
Unlabeled policy	72,000
Scorer	5,600
Calibration	5,600
Target DRE	145,350
Total	**228,550**

C Experimental Details

We adapt the experimental setting of Table 2 in TraPO (Yang et al., 2026) for both Range-GRPO and UARM (Pan et al., 2026). For all methods, we initialize the policy from Qwen2.5-Math-7B (Yang et al., 2024). Following TraPO, we select the checkpoint with the highest mean validation accuracy across AIME, AMC, MATH, Minerva, and OlympiadBench.

Our code is available at `https://anonymous.4open.science/r/RANGE-GRPO-BAE4`.

C.1 TraPO

We reproduce TraPO using its official implementation. The entropy coefficient differs between the paper and the official implementation: the paper uses 0.01, whereas the released configuration uses 0.001. We therefore evaluate both entropy coefficients while keeping all other settings fixed. As shown in Table 9, an entropy coefficient of 0.001 outperforms 0.01 on all benchmarks except GPQA, improving the average ID and OOD performance by 2.6 and 1.2 percentage points, respectively. We therefore report the results with an entropy coefficient of 0.001 in the main table.

For budget accounting, we follow the TraPO training procedure, which uses a warmup of 10 epochs followed by 5 epochs that combine labeled and unlabeled signals. TraPO generates rollouts for both labeled and unlabeled prompts in every training epoch. During warmup, only labeled rollouts contribute to policy updates, while unlabeled rollouts accumulate pass rate trajectories. In the subsequent stage, TraPO uses these trajectories to select reliable unlabeled samples for policy updates. We therefore count both labeled and unlabeled rollouts across all 15 training epochs. Table 10 summarizes the resulting response budget.

C.2 UARM

Reward Model Training. We use a frozen Qwen2.5-Math-7B checkpoint (Yang et al., 2024) as the RM backbone and train only a quantile head. The quantile head jointly predicts multiple reward quantiles for each prompt-response pair. For each prompt reserved for RM training, validation, or calibration, the initial policy generates 140 responses. Correctness checks against the ground truth provide binary reward targets: one for a correct answer and zero otherwise. Within each configuration, we retain the RM checkpoint with the lowest validation loss. Among configurations, we select the one with the highest mean validation AUC of the predicted median rewards. We use a separate labeled subset for calibration. Both the RM and calibration parameters remain fixed during policy training.

Policy Training. The experiment uses 2,048 prompts, comprising 1,024 labeled and 1,024 unlabeled prompts. Table 12 shows the allocation of labeled prompts to disjoint subsets. Following the warmup setting for other baselines in Table 2, training begins with ten epochs of labeled updates. The next five epochs combine labeled and unlabeled updates, giving 15 epochs in total. For labeled updates, we center the binary correctness rewards by their group mean, as in Dr.GRPO (Liu et al., 2025). Unlabeled updates use UARM advantages within the GRPO objective. We set the signal

Table 9: **Performance of reproduced TraPO configurations with different entropy coefficients.** All training methods use 1,024 (1K) ID and 1,024 (1K) OOD prompts. GPQA† denotes GPQA-diamond. Bold and underlined scores indicate the best and second-best results, respectively.

	In-Distribution						Out-of-Distribution			
Model / Method	AIME 24/25	AMC	MATH-500	Minerva	Olympiad	**Avg.**	ARC-c	GPQA†	MMLU-Pro	**Avg.**
TraPO (entropy=0.01)	<u>17.3</u>/<u>10.3</u>	<u>52.6</u>	<u>77.6</u>	<u>33.8</u>	<u>36.3</u>	<u>38.0</u>	<u>78.8</u>	**36.9**	<u>39.5</u>	<u>51.7</u>
TraPO (entropy=0.001)	**17.7/12.2**	**55.9**	**79.2**	**37.5**	**41.3**	**40.6**	**80.4**	<u>34.8</u>	**43.5**	**52.9**

Table 10: **Response budget for the main experiments.** The table compares the response budget for TraPO, UARM, and Range-GRPO. For TraPO, unlabeled rollouts include those generated during warmup. The UARM budget includes RM processing, whereas the Range-GRPO budget includes judge generations.

Category	TraPO	UARM	Range-GRPO
Labeled policy	122,880	92,160	92,160
Unlabeled rollouts	122,880	40,960	37,120
RM training	0	28,560	0
RM calibration	0	7,280	0
RM processing	0	76,800	0
Scorer fitting	0	0	1,280
Calibration	0	0	768
Target DRE	0	0	57,600
Judge generations	0	0	12,096
Total allocation	**245,760**	**245,760**	**201,024**

variance stabilizer to $\zeta = 10^{-6}$. For policy checkpoint selection, we choose the checkpoint with the highest mean accuracy across the five validation benchmarks. Table 11 lists the main training settings.

Table 11: **UARM training configuration for the cross-domain setting.** The table reports the policy training settings in (a) and the reward model training and calibration settings in (b).

(a) Policy training

Setting	Value
Model	Qwen2.5-Math-7B
Learning rate	10^{-6}
Batch size	64 prompts
Responses per prompt	8
Epochs	15
Warmup epochs	10

(b) RM training and calibration

Setting	Value
Learning rate	10^{-4}
Batch size	1,024 responses
Quantile outputs	11
Hidden dimensions	256, 64
Maximum epochs	600
Patience	30
Miscoverage α	0.1

Response Budget. For budget accounting, we count each generated response and each response processed by the RM as one budget unit. RM processing includes offline feature extraction and unlabeled response scoring. This accounting convention is used only for budget allocation. It does not assume that response generation and RM processing require the same computational cost.

C.3 RANGE-GRPO

Reward range construction. We use Qwen3-4B-Instruct-2507 [1] as a frozen judge. The judge extracts a low-cardinality judgment state Z using general criteria that are not tied to a specific task

[1] https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507

Table 12: **Data allocation for UARM and Range-GRPO in the main experiments.** (a) summarizes the UARM allocation across policy learning and RM training, validation, and calibration. (b) details the Range-GRPO allocation across policy learning, scorer fitting, calibration, source DRE, and target DRE.

(a) UARM

Role	Prompts
Labeled policy	768
RM training	184
RM validation	20
Calibration	52
Unlabeled policy	1,024
Total	**2,048**

(b) Range-GRPO

Role	Prompts
Labeled policy	768
Scorer / source DRE	160
Calibration	96
Unlabeled policy	928
Target DRE	96
Total	**2,048**

domain. Each judge input contains one problem and a group of eight responses. For each response, we provide its reasoning, submitted answer, and agreement with other responses. Reference answers and correctness labels are not provided. For response i, the judge extracts $Z_i = (z_{i1}, \ldots, z_{i5}) \in \{0, 0.5, 1\}^5$ according to Table 13. The judge produces a single deterministic structured output for each group of eight responses, with a maximum length of 768 tokens.

Table 13: **Five low-cardinality judgment states used to construct Z.** The first three states assess the response itself, while the last two capture agreement with other responses in the same rollout group. Each state takes a score of 0, 0.5, or 1.

State	Criterion	Score 0	Score 0.5	Score 1
Response-level assessment				
z_1	Answer verification	The judge identifies a contradiction when independently checking the submitted answer against the problem.	The judge cannot verify or refute the submitted answer with sufficient confidence.	The judge independently verifies the submitted answer against the problem.
z_2	Reasoning consistency	The visible reasoning contains a decisive error or conflicts with the submitted answer.	The visible reasoning is incomplete or insufficient to establish the submitted answer.	The visible reasoning consistently supports the submitted answer.
z_3	Requirement compliance	The response omits or violates an explicit task requirement.	It remains unclear whether all explicit requirements are satisfied.	The response satisfies all explicit task requirements.
Group-level agreement				
z_4	Absolute agreement	No other response gives the same answer.	One to three other responses give the same answer.	Four to seven other responses give the same answer.
z_5	Relative agreement	Fewer responses support this answer than the most supported alternative.	This answer and the most supported alternative receive equal support.	More responses support this answer than any alternative.

For the main experiments, we follow the range construction procedure in Algorithm 1 with two practical implementation choices. First, to align with the TraPO warmup schedule, we treat the policy immediately after labeled warmup as π_0. At this stage, π_0 generates the source and calibration responses for scorer fitting and calibration. We keep the fitted scorer and calibration nonconformity scores fixed throughout subsequent training.

Second, we normalize the calibration nonconformity scores so that differences in scorer predictions are reflected more clearly during range construction. For a scorer prediction $s_Z(Z) \in [0,1]$, we define

$$R_j = \frac{|Y_j - s_Z(Z_j)|}{\psi(s_Z(Z_j))}, \qquad \psi(v) = \max\{2\min(v, 1-v), 0.05\}. \tag{5}$$

For the same prediction error, the normalized nonconformity score increases as the scorer prediction approaches 0 or 1 because $\psi(v)$ becomes smaller. This makes the nonconformity score sensitive to the scorer prediction in addition to the absolute prediction error.

We compute the weighted cutoff $\widehat{q}_{\alpha,t}(Z)$ from the normalized nonconformity scores and rescale it for each query response as $\widehat{\tau}_{\alpha,t}(Z) = \psi(s_Z(Z))\,\widehat{q}_{\alpha,t}(Z)$. Here, $\psi(s_Z(Z))$ sets the query-specific error scale, while $\widehat{q}_{\alpha,t}(Z)$ provides the corresponding normalized conformal cutoff. We then use $\widehat{\tau}_{\alpha,t}(Z)$ in the range construction of Algorithm 1. Together, normalization and query-specific rescaling allow the constructed ranges to reflect differences in scorer confidence more directly. At each unlabeled update, we refit the DRE classifier using the fixed source states and fresh target states generated by the current policy. Table 14 summarizes the main range construction settings.

Table 14: **Range construction settings for the main experiments.** The scorer and calibration nonconformity scores remain fixed after warmup, whereas DRE is refreshed at each unlabeled update.

Setting	Value
Scorer	XGBoost regressor
Trees / depth	200 / 2
DRE	Random forest
Trees / depth	200 / 4
DRE target per update	768 responses
Refresh	Each unlabeled update
Nominal coverage	90%
Weight cap	95th percentile

Policy Training. Table 12(b) summarizes the prompt allocation used for Range-GRPO. The five subsets remain fixed and disjoint at the prompt level; scorer fitting and source DRE share the same responses, with correctness labels used only for scorer fitting.

We train for ten labeled warmup epochs followed by five active epochs using both labeled and unlabeled prompts. During active training, labeled updates use binary correctness rewards, while unlabeled updates use Range-GRPO advantages following Algorithm 1. Table 15 summarizes the main training settings.

Table 15: **Range-GRPO policy training settings for the main experiments.** (a) summarizes the policy training configuration. (b) lists the main optimization and Range-GRPO settings. LR denotes the learning rate.

(a) Policy training

Setting	Value
Batch size	64 prompts
Responses per prompt	8
Warmup epochs	10
Active epochs	5
Optimizer	AdamW
PPO clipping	0.2

(b) Main configuration

Setting	Value
LR	3×10^{-6}
Entropy coefficient	0.01
KL coefficient	0
λ	1
Maximum prompt tokens	3,072
Maximum response tokens	4,096

Response Budget. Table 10 details the response budget for policy training and reward range construction. We include both policy rollouts and judge generations when computing the Range-GRPO response budget. The resulting budget is 201,024 units, which is 18.2% lower than the 245,760 units

used for TraPO and UARM. Under this reduced budget, Range-GRPO still achieves strong overall performance, as shown in Table 2.

Wall-clock Runtime. Table 16 reports the observed wall-clock runtime for the main experiments. All runs use NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs. The GPU count refers to the allocation during active training. UARM uses four GPUs for policy training and one dedicated GPU for the reward model (RM), whereas Range-GRPO shares four GPUs between policy and judge phases.

Table 16: **Range-GRPO maintains comparable wall-clock training time despite the additional judge phase.** Each run consists of 10 warmup epochs followed by 5 active epochs. Phase durations and total training times are independently rounded to one decimal place.

Method	Active GPUs	Warmup (h)	Active (h)	Training (h)
TraPO (entropy=0.01)	4	12.8	10.6	23.4
TraPO (entropy=0.001)	4	12.6	7.0	19.7
UARM	5	6.5	9.9	16.4
Range-GRPO (ours)	4	6.3	10.0	16.3

Training time includes initialization, rollout generation, policy updates, and validation. Range-GRPO training time also accounts for judge loading and calls, scorer fitting, calibration, and DRE updates. UARM requires an additional 2.6 h for RM preparation before policy training, resulting in a total wall-clock time of 19.0 h when this preparation is included.

TraPO (entropy=0.01) has a longer active phase than TraPO (entropy=0.001). This difference is consistent with their average unlabeled response lengths of 2,244 and 553 tokens, respectively. Despite the additional judge phase, Range-GRPO shows a training wall-clock comparable to UARM while using four rather than five active GPUs. This practical efficiency is consistent with the reduced response budget reported in Table 10.