RECAL: REWARD CALIBRATION FOR RL-BASED LLM ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) routing has emerged as an effective paradigm for leveraging the complementary strengths of multiple LLMs through dynamic model and reasoning-strategy selection. Recent reinforcement learning (RL)-based routing methods further improve routing quality by optimizing routing policies from interaction feedback. However, they still struggle to provide informative and comparable learning signals under heterogeneous tasks with varying difficulty. In practice, multiple objectives (e.g., correctness, format behavior) are aggregated into a single scalar reward, leading to ambiguous credit assignment and conflicting optimization signals. Moreover, reward distributions vary substantially across instances and datasets, making their optimization contributions difficult to compare and potentially leading to imbalanced policy updates. To address these issues, we propose **ReCal**, a **Re**ward **Cal**ibration framework for RL-based LLM routing. We first introduce a hierarchical reward decomposition mechanism with component-wise advantage estimation. We further propose a distribution-aware optimization strategy that calibrates optimization contributions through variance-aware reweighting and per-dataset normalization. Experiments on seven QA benchmarks show that ReCal consistently improves routing performance and achieves more stable training dynamics compared with baselines. Additional experiments on mathematical reasoning further show that this calibration procedure can be applied beyond QA through task-specific reward instantiation.[1]

1 INTRODUCTION

LLM routing has emerged as an effective paradigm for leveraging the complementary strengths of multiple LLMs by dynamically selecting appropriate models or reasoning strategy for each query. Among existing approaches, a line of recent work formulates routing as a sequential decision-making problem and applies RL to optimize routing policies (Sikeridis et al., 2025; Zhang et al., 2025; Qian et al., 2025), achieving promising performance gains. Despite recent progress, existing routing methods often struggle to learn effective policies under queries with varying difficulty and routing ambiguity. In practice, learning signals can exhibit uneven statistical properties across instances and datasets, leading to heterogeneous optimization contributions during training. Consequently, routing policies may receive poorly calibrated supervision across different routing cases.

We argue that a key bottleneck of RL-based LLM routing lies in the learning signals used for routing policy optimization. Unlike standard generation tasks, routing policies must learn query-dependent model-selection behaviors under heterogeneous routing difficulty. However, there are two key technical issues with learning signals in current routing frameworks. First, routing policies are typically optimized using scalarized rewards that aggregate multiple objectives, such as answer correctness and formatting behavior, into a single score, even though different objectives may favor different routing decisions and model-selection strategies. As shown in Figure 1(a), the same scalarized reward may arise from different combinations of objective-level outcomes, making it difficult for the policy to identify which routing decision performs better. Consequently, scalarized rewards obscure routing credit assignment and weaken policy learning (Mannion et al., 2017; Silver et al., 2021; Huang et al., 2021). Second, routing rewards exhibit highly heterogeneous statistical properties across instances and datasets. As shown in Figure 1(b), some routing cases exhibit highly dispersed

[1] Implementation is available: `https://anonymous.4open.science/r/ReCal`

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.

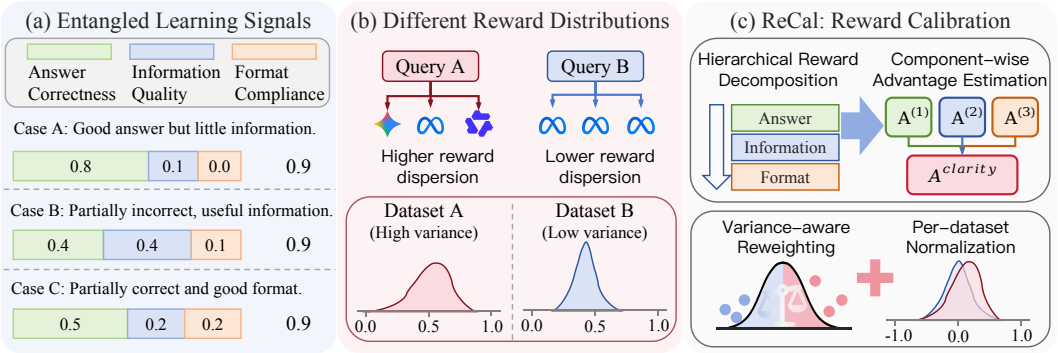

Figure 1: (a) Scalarized rewards entangle multiple objectives, leading to ambiguous credit assignment. (b) Reward distributions vary widely across routing instances and datasets, resulting in heterogeneous optimization contributions and potentially imbalanced policy updates. (c) ReCal provides clear, component-wise learning signals and applies distribution-aware optimization to calibrate optimization contributions across heterogeneous routing instances and datasets.

rewards within the sampled response group, while others yield more concentrated reward distributions. Different datasets further induce substantially different reward scales and variances. Such heterogeneity makes optimization contributions difficult to compare across instances and datasets, potentially leading to imbalanced policy updates. These observations raise two central challenges for RL-based LLM routing: **(1)** how to clarify learning signals for precise credit assignment from entangled rewards; and **(2)** how to calibrate optimization contributions across heterogeneous instances and datasets for optimization comparability and training stability.

To address these challenges, we propose a **Re**ward **Cal**ibration Framework for LLM routing, abbreviated as **ReCal**. Rather than directly optimizing with raw routing rewards, ReCal treats policy learning as a two-stage calibration process: it first restructures reward signals into clearer supervision, and then adjusts the contribution of these signals to policy updates under heterogeneous data distributions. We introduce a hierarchical, disentangled reward mechanism to improve learning signal clarity **(Challenge 1)**. Instead of estimating a single advantage from aggregated scalarized rewards, we separately estimate the advantage of each reward component to preserve objective-specific relative improvements. The resulting component-wise advantages are then combined through weighted aggregation to reflect the relative importance of different objectives. Building on the calibrated rewards, we further improve optimization comparability **(Challenge 2)** under heterogeneous reward distributions through a distribution-aware optimization strategy. At the instance level, we introduce variance-aware reweighting to dynamically calibrate update strength according to the reward dispersion within each sampled response group. At the dataset level, we apply per-dataset normalization to improve the comparability of optimization contributions across datasets with different reward scales and variances. This improves the comparability of learning signals across heterogeneous routing cases and leads to more stable training dynamics.

Our contributions are summarized as follows:

- We propose a learning-signal calibration perspective for RL-based LLM routing, which improves policy learning by jointly modeling learning signal clarity and optimization comparability across heterogeneous routing instances and datasets.
- We introduce a hierarchical and disentangled reward mechanism that performs component-wise advantage estimation before aggregation, enabling clearer credit assignment while preserving objective-specific relative improvements during policy optimization.
- We propose a distribution-aware optimization strategy that calibrates optimization contributions across heterogeneous routing instances and datasets through variance-aware reweighting and per-dataset normalization before policy optimization.
- Extensive experiments on seven QA benchmarks show consistent improvements in routing performance and more stable training, while additional evaluations on mathematical reasoning show that the same calibration procedure can be applied beyond QA routing tasks through task-specific reward instantiation.

2 Related Work

2.1 LLM Routing and RL-based Optimization

Existing LLM routing methods can be broadly categorized into three groups. Early approaches rely on heuristic or rule-based strategies (Hu et al., 2024), which use predefined criteria such as confidence or cost to select models. Supervised learning-based routers (Ong et al., 2025; Feng et al., 2024; Chen et al., 2024) further improve flexibility by training classifiers or scoring models to predict the best routing decisions. More recently, RL-based approaches such as PickLLM (Sikeridis et al., 2025), Router-R1 (Zhang et al., 2025), and xRouter (Qian et al., 2025) optimize routing policies from reward feedback, enabling adaptive trade-offs between answer quality and model usage. These methods typically optimize a scalar reward that combines multiple objectives, such as answer quality and format accuracy. While effective, these methods primarily focus on improving routing decisions themselves, with limited attention to how learning signal design affects policy optimization.

2.2 Learning Signal Design and Optimization

A substantial body of RL research studies how learning signals should be constructed and utilized during policy optimization, including reward shaping (Ng et al., 1999), multi-objective RL (Hayes et al., 2022), and advantage-based optimization (Schulman et al., 2017). These approaches address complementary aspects of learning-signal design, such as structuring feedback across objectives and transforming rewards into effective optimization signals. However, they do not explicitly consider how heterogeneous reward distributions across routing instances and datasets interact with multi-component reward structures during policy optimization. Complementary to reward structuring, another line of work studies how heterogeneous data distributions affect optimization dynamics, focusing on techniques such as reward scaling and normalization in policy optimization (Ouyang et al., 2022; Rafailov et al., 2024), sample reweighting and off-policy correction (Swaminathan & Joachims, 2015), and curriculum or uncertainty-aware training (Bengio et al., 2009; Kendall et al., 2018). These methods aim to stabilize training and improve data efficiency by adjusting the scale or importance of learning signals. However, they primarily study either the semantic structure of learning signals or their optimization behavior under heterogeneous data, but rarely consider the interaction between the two. This separation becomes limiting in LLM routing, where task heterogeneity shapes both what information learning signals contain and their influence on policy updates.

3 Preliminaries

3.1 RL-based LLM Routing

Given a query x, the routing policy interacts with the candidate models over up to T routing steps. At each step, it generates a routing action and receives the selected model's response as context for subsequent generation or the final answer. We denote the router-generated actions in a rollout by a. The rollout is evaluated by a reward function $r(x, a, y)$, where y is the final response. These interactions define the router's learning signal. In practical routing systems, rewards are usually composed of multiple objectives, such as answer correctness, information quality, formatting behavior, or the number of external-model calls. Existing methods typically aggregate them into a scalar reward:

$$r = \sum_{m=1}^{M} \lambda_m r^{(m)}, \tag{1}$$

where M is the number of components, $r^{(m)}$ and λ_m are objective m's reward and its weight.

3.2 Policy Optimization

We optimize routing policies with a GRPO-style objective. For each query x_i, the policy samples G rollouts, each containing a complete routing sequence $a_{i,j}$ and a final response $y_{i,j}$. We denote the resulting response group by $\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,G}\}$, with scalar reward $r_{i,j}$ for each rollout. The standard PPO clipping objective and scalarized-reward group-relative advantage are provided in Appendix A.1. For the scalarized baseline, we denote the corresponding group-relative advantage by $A^{\text{base}}_{i,j}$ while ReCal's calibrated advantages and final objective are defined in Sections 4.

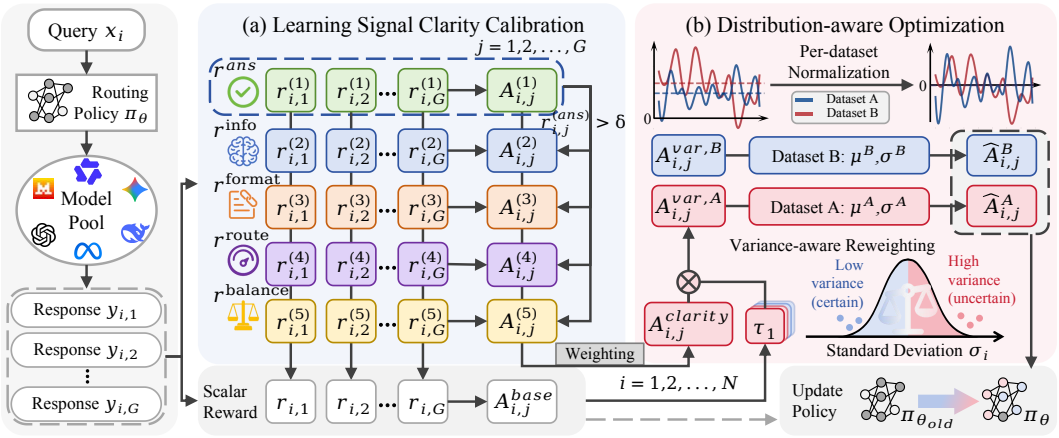

Figure 2: Overview of ReCal. Given a query, the routing policy samples multiple routing actions and obtains a response group. ReCal first decomposes routing rewards into multiple hierarchical objectives and performs component-wise advantage estimation before weighted aggregation. It then calibrates optimization contributions through variance-aware reweighting and per-dataset normalization under heterogeneous reward distributions before the final policy update.

4 The Proposed Framework: ReCal

Figure 2 illustrates the overall framework of ReCal. Starting from a response group sampled by the routing policy, ReCal performs a two-stage learning-signal calibration process before policy optimization. First, ReCal improves learning-signal clarity through hierarchical reward decomposition and component-wise advantage estimation. It then calibrates optimization contributions across heterogeneous routing instances and datasets through variance-aware reweighting and per-dataset normalization, before using the resulting advantages in a GRPO-style objective.

4.1 Learning Signal Clarity Calibration

Rather than estimating advantages directly from aggregated scalarized rewards, ReCal separates reward components before advantage estimation, preserving objective-specific relative improvements.

4.1.1 Hierarchical Reward Decomposition

For each j-th rollout of query x_i, whose final response $y_{i,j}$ belongs to the response group $\mathcal{Y}_i$, ReCal computes a set of reward components: $\mathbf{r}_{i,j} = \{r_{i,j}^{\text{ans}}, r_{i,j}^{\text{info}}, r_{i,j}^{\text{format}}, r_{i,j}^{\text{route}}, r_{i,j}^{\text{balance}}\}$. r^{ans} is computed from the final answer by comparing it with the ground truth. r^{info} measures the quality of information returned by external models during the rollout, using F1 against the ground truth. r^{format} is a binary reward indicating whether the final response follows the required output format. r^{route} is derived from the number of external model calls in the routing sequence and encourages the router to perform multi-round routing. r^{balance} encourages balanced router utilization based on historical routing counts. To stabilize routing behavior, we maintain exponentially decayed routing counts for each candidate model. Let $\hat{c}_s^{(k)}$ and $c_s^{(k)}$ denote the routing count and accumulated routing count of model a_k at training step s, respectively. The routing statistics are updated as:

$$c_s^{(k)} = \alpha c_{s-1}^{(k)} + \hat{c}_s^{(k)}, \tag{2}$$

where $\alpha \in [0, 1)$ controls the decay of historical routing counts. r^{balance} is then inversely correlated with the normalized routing proportion of the selected model.

These objectives are organized hierarchically according to their dependency on answer correctness. Let $\mathcal{M}_{\text{aux}} = \{\text{info}, \text{format}, \text{route}, \text{balance}\}$ denote the auxiliary reward components, which are activated only when the response satisfies a correctness threshold:

$$r_{i,j}^{(m)} = \mathbb{I}(r_{i,j}^{\text{ans}} > \delta) \cdot \tilde{r}_{i,j}^{(m)}, \qquad m \in \mathcal{M}_{\text{aux}}, \tag{3}$$

where $\tilde{r}_{i,j}^{(m)}$ denotes the raw reward, δ is the activation threshold, and $\mathbb{I}(\cdot)$ is the indicator function. The answer reward is not subject to this gating rule. The concrete reward components and correctness-gated activation rule above are instantiated for QA routing. The subsequent calibration operations, including component-wise advantage estimation, variance-aware reweighting, and per-dataset normalization, do not depend on these particular reward semantics. For other tasks, available feedback can define the reward components and gate while retaining the same calibration procedure.

4.1.2 Component-wise Advantage Estimation

Instead of estimating advantages after scalar reward aggregation, ReCal computes group-relative advantages independently for each reward component before scalar reward aggregation:

$$\mu_i^{(m)} = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}^{(m)}, \quad \sigma_i^{(m)} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j}^{(m)} - \mu_i^{(m)})^2}, \quad A_{i,j}^{(m)} = \frac{r_{i,j}^{(m)} - \mu_i^{(m)}}{\sigma_i^{(m)} + \varepsilon}, \tag{4}$$

where $\mu_i^{(m)}$ and $\sigma_i^{(m)}$ are the mean and standard deviation of component m within group $\mathcal{Y}_i$.

The final clarity-calibrated advantage is then obtained through weighted aggregation:

$$A_{i,j}^{\text{clarity}} = \sum_{m=1}^{M} \lambda_m A_{i,j}^{(m)}. \tag{5}$$

Here, λ_m weights the standardized component-wise advantage $A_{i,j}^{(m)}$ for objective m. Compared with scalarized optimization, this formulation preserves objective-specific relative improvements before aggregation, providing clearer objective-level credit signals for policy optimization.

4.2 Distribution-aware Optimization

Although disentangled advantages improve learning-signal clarity, heterogeneous reward distributions can still yield optimization contributions that are difficult to compare across instances and datasets. ReCal therefore calibrates these contributions at both the instance and dataset levels.

4.2.1 Variance-aware Reweighting

For dispersion estimation, we use the scalar reward $r_{i,j} = \sum_{m=1}^{M} \lambda_m r_{i,j}^{(m)}$. This scalarization serves only as a group-level dispersion signal within each sampled response group, while component-wise advantages are computed independently before aggregation. For each response group $\mathcal{Y}_i$, with the reward mean $\mu_i = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}$, we use the intra-group reward standard deviation as an observable proxy for disagreement among sampled routing decisions:

$$\sigma_i = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j} - \mu_i)^2}. \tag{6}$$

Intuitively, a larger intra-group reward standard deviation indicates greater reward dispersion among sampled routing decisions, which provides an observable basis for instance-level reweighting. ReCal therefore rescales optimization strength and computes the variance-calibrated advantage as:

$$\tau_i = \text{clip}\left(\frac{\sigma_i}{\bar{\sigma} + \varepsilon}, \tau_{\min}, \tau_{\max}\right), \qquad A_{i,j}^{\text{var}} = \tau_i A_{i,j}^{\text{clarity}}, \tag{7}$$

where N is the number of query groups in the current training batch, $\bar{\sigma} = \frac{1}{N}\sum_{i=1}^{N} \sigma_i$ denotes the average intra-group reward standard deviation within that batch, and $\tau_{\min}$ and $\tau_{\max}$ are clipping thresholds. This mechanism assigns larger weights to routing cases with greater reward dispersion within the response groups, while the clipping bounds limit the magnitude of the reweighting.

4.2.2 PER-DATASET NORMALIZATION

Different datasets may exhibit substantially different reward magnitudes and variances, resulting in heterogeneous advantage scales and making their optimization contributions difficult to compare directly. Therefore, ReCal normalizes advantages independently within each dataset. For each dataset d represented in the current batch, let $\mathcal{D}^{(d)}$ denote its rollout-response indices (i, j). The dataset-wise statistics are computed separately using only the samples within the batch:

$$\mu^{(d)} = \frac{1}{|\mathcal{D}^{(d)}|} \sum_{(i,j)\in\mathcal{D}^{(d)}} A^{\text{var}}_{i,j}, \quad \sigma^{(d)} = \sqrt{\frac{1}{|\mathcal{D}^{(d)}|} \sum_{(i,j)\in\mathcal{D}^{(d)}} (A^{\text{var}}_{i,j} - \mu^{(d)})^2}, \quad \hat{A}_{i,j} = \frac{A^{\text{var}}_{i,j} - \mu^{(d)}}{\sigma^{(d)} + \varepsilon}. \tag{8}$$

This normalization aligns advantage scales across datasets within the current training batch, improving the comparability of learning signals under heterogeneous reward distributions.

4.3 OVERALL TRAINING OBJECTIVE

$\hat{A}_{i,j}$ is integrated into a GRPO-style clipped policy optimization objective:

$$\mathcal{L}_{\text{ReCal}}(\theta) = \mathbb{E}\left[\min\left(\rho_{i,j}\hat{A}_{i,j}, \text{clip}(\rho_{i,j}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,j}\right)\right], \quad \rho_{i,j} = \frac{\pi_\theta(a_{i,j}|x_i)}{\pi_{\theta_{\text{old}}}(a_{i,j}|x_i)}, \tag{9}$$

where $a_{i,j}$ denotes the complete routing sequence of the j-th rollout for query x_i. The sequence-level ratio $\rho_{i,j}$ is a shorthand for the token-level probability ratios evaluated over the valid router tokens in $a_{i,j}$. In implementation, the clipped GRPO loss is computed only on these valid router tokens; tokens from responses returned by external models are used as context but do not contribute to the objective. Overall, ReCal provides a unified objective for heterogeneous routing instances and datasets by calibrating learning-signal clarity and optimization contributions before policy updates.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets We evaluate ReCal on seven question-answering (QA) datasets: (1) General QA: Natural Question (NQ) (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023); (2) Multi-Hop QA: HotpotQA (HpQA) (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), Musique (Trivedi et al., 2022), and Bamboogle (Bamb) (Press et al., 2023).

Baselines We compare ReCal with three categories of methods: **(1) No routing**: Direct Inference (Vanilla), Chain-of-Thought (CoT) (Wei et al., 2022), supervised fine-tuning (SFT), Retrieval-Augmented Generation (RAG) based on the external knowledge Wikipedia-18 (Karpukhin et al., 2020) and the retriever E5 (Wang et al., 2024), and Search-R1 (Jin et al., 2025); **(2) Heuristic & Discriminative routing**: Largest LLM, Prompt LLM, Prompt LLM* (Prompt LLM + task decomposition), KNN Router(Hu et al., 2024), KNN Router* (KNN Router + task decomposition), MLP Router(Hu et al., 2024), RouteLLM(Ong et al., 2025), RouterDC(Chen et al., 2024), and GraphRouter(Feng et al., 2024); **(3) RL-based routing**: Router-R1(Zhang et al., 2025).

Implementation Details We used Qwen2.5-3B-Instruct (Yang et al., 2025) as the base model and conducted RL training using GRPO within the VeRL [2] framework on NVIDIA A100 GPUs, with a batch size of 64. Similar to Router-R1, we combined 7K samples each from the NQ and HotpotQA datasets to create a training set. After training, we evaluated performance on the two in-domain datasets and the other five out-of-domain datasets by sampling 500 test instances from each of them (excluding Bamboogle, which contained a total of 125 test instances). All baseline models were trained (where applicable) and evaluated on consistent datasets and settings. We selected a set of six models of varying sizes and architectures via the NVIDIA NIM API [3] and

[2] https://github.com/verl-project/verl

[3] https://build.nvidia.com/

Table 1: EM results on seven QA datasets. **Bold** denotes the best and underline denotes the secondary; † denotes in-domain evaluation, and others are out-of-domain. **Ours+** uses two additional LLMs introduced only at inference time.

Methods	General QA			Multi-Hop QA				Avg.
	NQ†	TriviaQA	PopQA	HpQA†	2Wiki	Musique	Bamb	
No Routing								
Vanilla	0.092	0.260	0.122	0.140	0.266	0.026	0.040	0.135
CoT	0.126	0.358	0.160	0.168	0.208	0.046	0.224	0.184
SFT	0.212	0.400	0.160	0.198	0.256	0.052	0.112	0.199
RAG	0.298	0.540	0.366	0.216	0.146	0.078	0.224	0.267
Search-R1	0.328	0.510	0.324	0.236	0.278	0.090	0.272	0.291
Heuristic & Discriminative Routing								
Largest LLM	0.296	0.578	0.354	0.278	0.274	0.104	0.480	0.338
Prompt LLM	0.300	0.580	0.340	0.268	0.262	0.108	0.448	0.329
Prompt LLM*	0.258	0.500	0.256	0.206	0.248	0.078	0.472	0.288
KNN Router	0.262	0.528	0.222	0.224	0.196	0.066	0.360	0.265
KNN Router*	0.236	0.478	0.232	0.154	0.234	0.072	0.384	0.256
MLP Router	0.252	0.460	0.222	0.198	0.210	0.072	0.360	0.253
RouteLLM	0.230	0.516	0.192	0.216	0.206	0.058	0.312	0.247
RouterDC	0.278	0.592	0.282	0.244	0.218	0.080	0.504	0.314
GraphRouter	0.276	0.586	0.280	0.234	0.180	0.076	0.448	0.297
RL-based Routing								
Router-R1	0.334	0.668	0.400	0.412	0.476	0.186	0.424	0.414
Ours	**0.404**	**0.712**	0.466	0.442	**0.560**	**0.212**	0.568	**0.481**
Ours+	0.390	0.698	**0.466**	**0.448**	0.554	0.206	**0.584**	0.478

OpenRouter [4]: Qwen2.5-7B-Instruct (Yang et al., 2025), LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), LLaMA-3.1-70B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), Mixtral-8x22B-Instruct (Jiang et al., 2024), and Gemma-2-27B-Instruct (Team et al., 2024). ReCal and Router-R1 use the same training data, base routing model, candidate model pool, and batch size. We train both methods until convergence. ReCal converges within 150 training steps, whereas Router-R1 requires approximately 210 steps because of its slower convergence.

5.2 Main Results

As shown in Table 1, ReCal achieves the highest average performance among all compared methods on seven QA benchmarks under a common evaluation protocol. Compared with no-routing baselines, Router-R1 and ReCal achieve higher performance across all seven benchmarks, indicating the empirical benefit of adaptive routing under the evaluated setting. Router-R1 also obtains a higher average EM than the heuristic and discriminative routing baselines, while ReCal further improves over Router-R1 on all seven benchmarks. Together, these results support the benefit of calibrating learning-signal clarity and optimization comparability in RL-based routing.

Ours achieves the best overall average EM score of 0.481, outperforming Router-R1 by 6.7 percentage points (0.481 vs. 0.414). The improvements are particularly pronounced on more challenging multi-hop QA benchmarks. For example, ReCal improves performance from 0.476 to 0.560 on 2Wiki and from 0.424 to 0.568 on Bamboogle. These results suggest that learning-signal calibration is particularly beneficial on challenging multi-hop routing tasks.

On in-domain datasets, ReCal also demonstrates strong improvements. Compared with Router-R1, our method improves EM from 0.334 to 0.404 on NQ and from 0.412 to 0.442 on HotpotQA. More importantly, the gains consistently extend to unseen out-of-domain datasets, suggesting that the learned routing policy generalizes beyond its training distributions to these unseen datasets.

We further observe that heuristic and discriminative routing methods exhibit substantial performance variation across datasets. Prompt LLM and RouterDC perform competitively on certain benchmarks, such as TriviaQA or Bamboogle, but their overall average performance remains lower than that of RL-based routing methods. This suggests that fixed heuristic or discriminative routing strategies may be less consistent across heterogeneous task requirements.

[4] https://openrouter.ai/

Table 2: Results of Ablation. LSC means ***Learning Signal Clarity*** Calibration, CAE means ***Component-wise Advantage Estimation***, DAO means ***Distribution-aware Optimization***. † indicates in-domain evaluation. **Bold** denotes the best and underline denotes the secondary.

Methods	NQ†	TriviaQA	PopQA	HpQA†	2Wiki	Musique	Bamb	Avg.
w/o both	0.324	0.618	0.332	0.354	0.334	0.132	0.456	0.364
w/o LSC	0.346	0.666	0.390	0.378	0.464	0.156	0.528	0.418
w/o CAE	0.330	0.662	0.420	0.408	0.496	0.160	0.536	0.430
w/o DAO	0.366	0.690	0.438	0.436	0.502	0.188	0.568	0.456
Ours	**0.404**	**0.712**	**0.466**	**0.442**	**0.560**	**0.212**	**0.568**	**0.481**

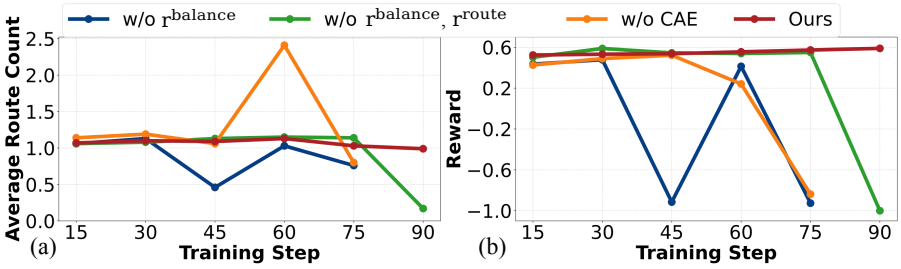

Figure 3: As training progresses, for full ReCal and versions that remove $r^{balance}$, $r^{balance}$&r^{route}, and CAE, (a) and (b) display the average route count and reward, respectively.

5.3 In-depth Analysis

5.3.1 Ablation Study

As shown in Table 2, removing either Learning Signal Clarity Calibration **(LSC)** or Distribution-aware Optimization **(DAO)** generally degrades performance across datasets, while removing both results in the largest drop. These results provide empirical evidence that both learning-signal clarity and optimization comparability contribute to effective RL-based routing. Using the same five reward components, the scalarized control obtains an average EM of 0.415 versus 0.481 for ReCal, while finer-grained ablations of the calibration operations are reported in Appendix D.

Removing LSC decreases the average score to 0.418, supporting the benefit of preserving objective-specific learning signals instead of directly optimizing scalarized rewards in the evaluated QA setting. Removing only Component-wise Advantage Estimation **(CAE)** reduces the average score to 0.430, indicating that component-wise relative advantage estimation provides an additional benefit within the learning-signal calibration mechanism. Removing DAO decreases the average score from 0.481 to 0.456, with particularly clear drops on challenging multi-hop datasets such as 2Wiki and Musique. This result indicates that calibrating optimization contributions across heterogeneous routing cases and datasets provides an empirical benefit across most evaluated datasets. As Figure 3 shows, removing r^{balance} results in less stable training behavior and a substantial reward decrease in the later stage of optimization, accompanied by signs of routing collapse. Removing both r^{balance} and r^{route} similarly leads to sharp decreases in routing counts and reward during later training. Removing CAE exhibits a different failure pattern: the routing count shows pronounced spikes while the reward subsequently drops sharply. This behavior is consistent with less stable training dynamics when advantages are estimated from scalarized rewards. In contrast, the full ReCal framework exhibits smoother routing behavior and more stable reward improvement throughout training.

5.3.2 Stability and Convergence of Training

Figure 4(a) presents the actor entropy loss during RL training. ReCal shows a substantially faster decrease in entropy and reaches a relatively stable low-entropy regime earlier in training, whereas Router-R1 exhibits larger oscillations and converges more slowly. These observations indicate that ReCal yields more stable training dynamics under the evaluated setting.

As shown in Figure 4(b), ReCal also exhibits more consistent performance improvements across NQ and HotpotQA during training, while Router-R1 shows larger fluctuations across the two datasets. In

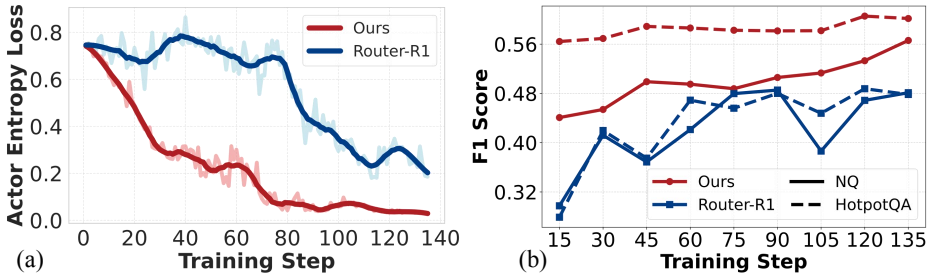

Figure 4: As training progresses, (a) shows the actor entropy loss of ReCal and Router-R1, and (b) reports their F1 trajectories on fixed validation subsets of NQ and HotpotQA.

particular, during training steps 60–75, Router-R1 improves on NQ while simultaneously degrading on HotpotQA. This behavior provides empirical evidence consistent with cross-dataset optimization interference under heterogeneous reward distributions. By contrast, ReCal improves both datasets more consistently over training, suggesting that learning-signal calibration helps maintain more balanced training dynamics across heterogeneous tasks.

5.3.3 Adaptation to an Expanded Model Pool

To evaluate whether ReCal can accommodate changes in the candidate model pool, we introduce two additional LLMs, DeepSeek-V3 (DeepSeek-AI et al., 2025) and Qwen2.5-72B-Instruct (Yang et al., 2025), only at inference time. These models are not included in the routing pool during training. We augment the routing prompt with textual descriptions of the newly introduced models, allowing the routing policy to select them. As shown in Table 1, **Ours+** maintains competitive overall performance and improves several datasets after expanding the candidate model pool. These results indicate that the learned routing policy can accommodate newly introduced models and make use of their textual descriptions without retraining the router.

5.3.4 Application Beyond QA

We further evaluate whether the same calibration procedure applies beyond QA on mathematical reasoning tasks. When trained on DAPO-Math-17K (Yu et al., 2025), ReCal achieves accuracies of 0.425 and 0.620 on AMC23 (math-ai, 2023) and MATH500 (Lightman et al., 2023), compared with 0.350 and 0.466 for Router-R1, respectively. Importantly, the calibration procedure remains unchanged, while only the task-specific reward instantiation is adapted to mathematical reasoning. Detailed experimental settings and reward definitions are provided in Appendix E.

6 Conclusion

In this paper, we propose ReCal, a reward calibration framework for RL-based LLM routing. ReCal combines hierarchical reward decomposition and component-wise advantage estimation to clarify objective-level supervision, together with distribution-aware optimization that calibrates optimization contributions across routing instances and datasets through variance-aware reweighting and per-dataset normalization. Together, these components provide a unified calibration procedure for learning from heterogeneous routing instances and datasets during policy optimization. Experiments on seven QA benchmarks show that ReCal consistently improves routing performance and achieves more stable training dynamics compared with the evaluated RL-based routing baseline. Additional evaluations further indicate that the learned routing policy generalizes to unseen QA datasets and accommodates an expanded model pool, while the same calibration procedure can be applied to mathematical reasoning through task-specific reward instantiation. The current evaluation uses a single base routing model and one primary RL-based routing baseline under a fixed candidate model pool, with QA as the main evaluation setting. The mathematical reasoning experiment adapts the task-specific reward instantiation, and the API-based evaluation may also be affected by provider-side model updates. These factors delimit the scope of the present findings and motivate future evaluation beyond text-based routing, including broader routing settings and more adaptive reward specification and calibration strategies for larger model pools.

AI USE STATEMENT

Generative AI tools were used to adapt the manuscript to the ICLR 2027 template and format references. They were not used to generate experimental data or results in this formatting pass. The authors reviewed all AI-assisted work and take responsibility for the final content of this paper, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We use only publicly available datasets that are widely adopted in the LLM community. All evaluated models are accessed through publicly available APIs or open-source checkpoints under their respective licenses.

REPRODUCIBILITY STATEMENT

The method and optimization objective are described in Sections 4–4.3. Appendix sections provide additional experimental results, hyperparameters, and implementation details. An anonymized implementation is provided through the URL given in the abstract for reproducibility.

REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. Routerdc: query-based router by dual contrastive learning for assembling large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li,

Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *The Thirteenth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2410.03834.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily

Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), April 2022. ISSN 1387-2532. doi: 10.1007/s10458-022-09552-y. URL https://doi.org/10.1007/s10458-022-09552-y.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system, 2024. URL https://arxiv.org/abs/2403.12031.

Changxin Huang, Guangrun Wang, Zhibo Zhou, Ronghui Zhang, and Liang Lin. Reward-adaptive reinforcement learning: Dynamic policy gradient optimization for bipedal locomotion, 2021. URL https://arxiv.org/abs/2107.01908.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. URL https://arxiv.org/abs/1705.07115.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Patrick Mannion, Sam Devlin, Karl Mason, Jim Duggan, and Enda Howley. Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing*, 263:60–73, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.05.090. URL https://www.sciencedirect.com/science/article/pii/S0925231217311037. Multi-objective Reinforcement Learning: Theory and Applications.

math-ai. AMC23. Hugging Face dataset, 2023. URL https://huggingface.co/datasets/math-ai/amc23.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pp. 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Cheng Qian, Zuxin Liu, Shirley Kokane, Akshara Prabhakar, Jielin Qiu, Haolin Chen, Zhiwei Liu, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. xrouter: Training cost-aware llms orchestration system via reinforcement learning, 2025. URL https://arxiv.org/abs/2510.08439.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

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

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

Dimitrios Sikeridis, Dennis Ramdass, and Pranay Pareek. Pickllm: Context-aware rl-assisted large language model routing. In *International Workshop on AI for Transportation*, pp. 227–239. Springer, 2025. URL https://arxiv.org/abs/2412.12170.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103535. URL https://www.sciencedirect.com/science/article/pii/S0004370221000862.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 814–823, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/swaminathan15.html.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur,

Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL https://arxiv.org/abs/2212.03533.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

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

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *arXiv preprint arXiv:2506.09033*, 2025.

A Additional Method and Implementation Details

A.1 PPO and GRPO Background

RL-based routing policies can be optimized using Proximal Policy Optimization (PPO)-style (Schulman et al., 2017) objectives. Let $\pi_{\theta_{\text{old}}}$ denote the old routing policy. PPO optimizes the clipped objective:

$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}\left[\min\left(\rho_t A_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t\right)\right], \quad \rho_t = \frac{\pi_\theta(a_t|x_t)}{\pi_{\theta_{\text{old}}}(a_t|x_t)}, \tag{10}$$

where ρ_t is the token- or action-level policy ratio for position t, ϵ is the clipping threshold, and A_t denotes a generic advantage estimate. For the scalarized-reward GRPO baseline considered below, this generic advantage is instantiated as the rollout-level quantity $A_{i,j}^{\text{base}}$.

Recent RL frameworks further adopt Group Relative Policy Optimization (GRPO)-style optimization (Shao et al., 2024). For each query x_i, the policy samples G rollouts, where the j-th rollout consists of a complete routing sequence $a_{i,j}$ and its final response $y_{i,j}$. The resulting response group is $\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,G}\}$, and each response receives a scalar reward $r_{i,j}$. GRPO computes the scalarized-reward group-relative baseline advantage:

$$\mu_i = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}, \quad \sigma_i = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j}-\mu_i)^2}, \quad A_{i,j}^{\text{base}} = \frac{r_{i,j}-\mu_i}{\sigma_i + \varepsilon}, \tag{11}$$

where μ_i and σ_i denote the mean and standard deviation of the scalarized rewards within response group $\mathcal{Y}_i$, respectively, and ε is a small constant for numerical stability.

A.2 Computational Cost

ReCal adds no model forward passes or API calls. Component-wise advantage estimation, variance-aware reweighting, and per-dataset normalization operate only on scalar reward/advantage statistics and normalization values already available in each training batch. The balance reward maintains only decayed routing-count statistics. Relative to response generation, this additional computation is negligible and limited to batch-level scalar operations.

B CASE STUDY

Response of ReCal

<think> To answer the question ""Who is responsible for introducing the principle of uniformitarianism?"", I need to perform a search for the historical context and the originator of this geological principle. I will seek information from a specialized LLM to accurately identify the person who introduced this principle. </think>
<search> LLaMA-3.1-70B-Instruct: Who is responsible for introducing the principle of uniformitarianism? </search>
<information> The principle of uniformitarianism is attributed to James Hutton (1726-1797), a Scottish geologist and physician. However, it was Charles Lyell (1797-1875), an English geologist, who popularized and further developed this concept in his influential book ""Principles of Geology"" (1830-1833). Lyell's work built upon Hutton's ideas and presented a comprehensive framework for understanding the Earth's geological history through the principle of uniformitarianism. In essence, while Hutton introduced the concept, Lyell is often credited with fully articulating and promoting the principle of uniformitarianism, making it a cornerstone of modern geology. </information>
<think> After consulting the information provided, I now have a clearer understanding of the role of James Hutton and Charles Lyell in the development of the principle of uniformitarianism. Hutton is credited with the initial introduction of the concept, but Lyell is recognized for popularizing and fully establishing this principle in geological studies. </think>
<search> Mixtral-8x22B-Instruct: Who is responsible for introducing the principle of uniformitarianism? </search>
<information> The principle of uniformitarianism was introduced by the Scottish geologist James Hutton. He proposed this principle in the late 18th century, suggesting that the geological processes that shaped Earth in the past are the same slow-acting processes that we observe today. This principle contrasted with catastrophism, which held that Earth's features were formed by sudden, violent events. Hutton's uniformitarianism, also known as the theory of gradualism, had a significant impact on the development of modern geology. </information>
<answer> James Hutton </answer>

Response of Router-R1

<think> The principle of uniformitarianism is a theory of the earth's past and present processes, so historically it was more commonly developed through empirical observations and integrative analysis rather than solely credited to a single person. Multiple researchers often make significant contributions as they gather, analyze, or theorize about the same evidence, sometimes reinforcing and building on each other's work. </think>
<search> LLaMA-3.1-70B-Instruct: Who introduced the principle of uniformitarianism? When answering, consider historical evidence and recognition among scientists for the principle's development. </search>
<information> The principle of uniformitarianism was introduced by James Hutton (1726-1797) and further developed by Charles Lyell (1797-1875). However, it was Lyell who popularized and extensively applied the concept in his influential book ""Principles of Geology"" (1830-1833). Lyell's work built upon Hutton's earlier ideas, and his book became a seminal text in the field of geology. Lyell's uniformitarianism was widely accepted by scientists, including Charles Darwin, who was heavily influenced by Lyell's work during his voyage on the Beagle. Historical evidence and recognition among scientists confirm that Lyell is often credited with the development and popularization of uniformitarianism, while Hutton is acknowledged as a precursor to Lyell's work. </information>
<answer> Charles Lyell </answer>

For the question ***"Who is responsible for introducing the principle of uniformitarianism?"***, the correct answer is ***James Hutton***. In this example, ReCal obtains the correct answer after invoking multiple models, whereas Router-R1 returns an incorrect answer.

Table 3: F1 results on seven QA datasets. **Bold**/underline denote the best/second-best results; † denotes in-domain evaluation, and others are out-of-domain. **Ours+** uses two additional LLMs introduced only at inference time.

Methods	General QA			Multi-Hop QA				Avg.
	NQ†	TriviaQA	PopQA	HpQA†	2Wiki	Musique	Bamb	
No Routing								
Vanilla	0.162	0.341	0.154	0.215	0.304	0.081	0.112	0.196
CoT	0.218	0.431	0.185	0.260	0.251	0.106	0.332	0.255
SFT	0.289	0.460	0.207	0.281	0.291	0.121	0.173	0.260
RAG	0.414	0.622	0.452	0.307	0.187	0.134	0.303	0.346
Search-R1	0.407	0.575	0.383	0.328	0.317	0.145	0.387	0.363
Heuristic & Discriminative Routing								
Largest LLM	0.431	0.695	0.423	0.416	0.397	0.199	0.608	0.453
Prompt LLM	0.437	0.694	0.409	0.407	0.393	0.205	0.579	0.446
Prompt LLM*	0.373	0.600	0.315	0.313	0.355	0.165	0.580	0.386
KNN Router	0.388	0.627	0.281	0.341	0.289	0.141	0.496	0.366
KNN Router*	0.360	0.572	0.272	0.248	0.299	0.138	0.493	0.340
MLP Router	0.368	0.557	0.272	0.295	0.277	0.142	0.460	0.339
RouteLLM	0.362	0.635	0.251	0.335	0.284	0.135	0.423	0.346
RouterDC	0.410	0.694	0.328	0.381	0.293	0.165	0.623	0.413
GraphRouter	0.412	0.690	0.324	0.366	0.258	0.154	0.546	0.393
RL-based Routing								
Router-R1	0.449	0.745	0.458	0.494	0.529	0.263	0.596	0.504
Ours	**0.508**	**0.784**	**0.505**	0.529	0.613	**0.316**	0.690	**0.564**
Ours+	0.499	0.776	0.502	**0.534**	**0.613**	0.308	**0.710**	0.563

Table 4: Fine-grained ablation results. All settings use the same reward components.

Setting	Avg. EM
Router-R1-style scalarized advantage	0.415
w/o variance-aware reweighting	0.458
w/o per-dataset normalization	0.463
ReCal	**0.481**

C More Experimental Results

We present additional F1 results on seven QA datasets in Table 3. ReCal consistently outperforms the compared baselines across both general and multi-hop QA tasks, achieving the highest average F1 score. It also maintains competitive performance after introducing unseen models.

D Fine-grained Ablation

LSC comprises both hierarchical reward decomposition and component-wise advantage estimation, whereas DAO comprises variance-aware reweighting and per-dataset normalization. Accordingly, "w/o LSC" retains DAO but removes both hierarchical reward design and CAE, while "w/o both" removes both LSC and DAO. The same-reward scalarized control starts from the "w/o both" setting, replaces its reward with the same five components used by ReCal, and directly scalarizes these components before advantage estimation, without correctness gating or DAO.

To separate the effect of reward components from the way they are used, we construct a control that uses the same five reward components as ReCal but aggregates them before advantage estimation. As shown in Table 4, this control obtains an average EM of 0.415, compared with 0.481 for ReCal. Removing only variance-aware reweighting or per-dataset normalization reduces the average EM to 0.458 and 0.463, respectively. These results show that the improvement cannot be explained by the additional reward components alone, and that both operations in distribution-aware optimization contribute to the final performance in the evaluated fine-grained ablation setting.

E Mathematical Reasoning: Experimental Details

We train ReCal on DAPO-Math-17K and evaluate it on AMC23 and MATH500. The answer reward r^{ans} is binary and indicates whether the final mathematical answer is correct. For responses satisfying the correctness gate, the information reward r^{info} evaluates the mathematical information in the `<information>` field, including signals such as formulas, numerical values, and operators. The component-wise advantage estimation, variance-aware reweighting, and per-dataset normalization remain unchanged; only the task-specific reward instantiation is adapted to mathematical reasoning. The mathematical reasoning experiments use the same six-model candidate pool as the main QA experiments.

F Hyperparameter Settings and Sensitivity

Table 5: Hyperparameter Settings

Hyperparameter	**Value**
Learning Rate (Actor)	1e-6
Total Batch Size	64
ReCal maximum training steps	150
Max Routing Steps	3
Response group size G	4
GPU Utilization Ratio	0.6
Rollout Sampling Temperature	1.0
Correctness threshold δ	0.3
Routing reward weight λ_{route}	0.2
Routing-count decay α	0.9
$\tau_{\min}, \tau_{\max}$	0.8, 1.25

Correctness threshold. The correctness threshold $\delta = 0.3$ is selected according to the answer-correctness distribution observed on the training data.

Sensitivity to variance-aware clipping bounds. We further examine the sensitivity of ReCal to the clipping bounds used in variance-aware reweighting. Compared with the default setting of $(\tau_{\min}, \tau_{\max}) = (0.8, 1.25)$, using a wider clipping range of $(0.5, 2.0)$ produces no substantial performance difference. This suggests that ReCal is not overly sensitive to the exact choice of clipping bounds within the tested range under the evaluated QA-routing setting.

Sensitivity to the routing reward weight. We further examine the sensitivity to the weight of the routing reward r^{route}. The default weight is set to 0.2. Increasing it to 0.5 causes the number of external model calls to continuously increase during training and eventually leads to optimization collapse. This observation suggests that overly emphasizing multi-round routing can destabilize the learned routing behavior, while a moderate weight provides a more reliable optimization signal under the evaluated training configuration.

G Notation

Table 6: Summary of notations used in ReCal.

Notation Table	
(A) Routing and Policy Optimization	
x	Input query.
a	Complete routing sequence of a rollout.
$\pi_\theta(a\|x), \theta$	Routing policy for query x and its parameters, respectively.
i, j, m, k, d	Indices for query, rollout within a response group, reward component, candidate model, and dataset, respectively.
N, T	Number of query groups in the current batch and maximum routing steps per rollout, respectively.
t, s	Action/token position index in generic PPO notation and training-step index for routing counts, respectively.
y	Final response generated after executing routing sequence a.
A_t, ρ_t	Generic PPO advantage estimate and current-to-old policy ratio at position t, respectively.
$\rho_{i,j} = \frac{\pi_\theta(a_{i,j}\|x_i)}{\pi_{\theta_{\text{old}}}(a_{i,j}\|x_i)}$	Rollout-level shorthand for token-level current-to-old policy probability ratios over valid router tokens in $a_{i,j}$; external-model response tokens are excluded by the training loss mask.
ϵ, ε	PPO/GRPO clipping threshold and a small numerical-stability constant, respectively.
$\mathcal{L}_{\text{PPO}}, \mathcal{L}_{\text{ReCal}}$	Generic clipped PPO objective and ReCal's final GRPO-style training objective, respectively.
$c_s^{(k)}, \hat{c}_s^{(k)}$	Accumulated routing count and current-step routing count of candidate model a_k at training step s, respectively.
α	Decay coefficient for historical routing counts.
(B) Response Group and Reward Modeling	
$\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,G}\}$	A group of G responses sampled for query x_i.
G, M	Number of responses per group and number of reward components, respectively.
$r(x, a, y)$	Scalar reward evaluating the routing decision and generated response.
$r^{(m)}$	Reward component of the m-th objective.
$r_{i,j}$	Scalarized reward for rollout (i, j), used for group-level statistics.
$\mathcal{M}_{\text{aux}}$	Set of auxiliary reward components activated only when $r_{i,j}^{\text{ans}} > \delta$.
λ_m	Weight for objective m, used both to form the scalar reward and to aggregate standardized component-wise advantages.
$\mathbf{r}_{i,j}$	Reward vector of response $y_{i,j}$.
$r_{i,j}^{\text{ans}}$	Answer correctness reward based on F1 between prediction and ground-truth.
$r_{i,j}^{\text{info}}$	Information-quality reward computed from the F1 between generated supporting information and ground-truth.
$r_{i,j}^{\text{format}}$	Formatting reward indicating whether the response satisfies the format.
$r_{i,j}^{\text{route}}$	Routing reward derived from external-model invocation count.
$r_{i,j}^{\text{balance}}$	Routing-balance reward encouraging balanced model utilization.
$\tilde{r}_{i,j}^{(m)}$	Raw auxiliary reward before hierarchical gating.
δ	Correctness threshold for activating auxiliary rewards.
$\mathbb{I}(\cdot)$	Indicator function.
(C) Advantage Estimation and Calibration	
$A_{i,j}^{\text{base}}$	Scalarized-reward group-relative baseline advantage.
$A_{i,j}^{(m)}$	Component-wise advantage for the m-th reward objective.
$A_{i,j}^{\text{clarity}}$	Aggregated advantage after component-wise advantage estimation.
$A_{i,j}^{\text{var}}$	Variance-aware reweighted advantage.
$\hat{A}_{i,j}$	Final calibrated advantage after dataset-level normalization.
$\mu_i^{(m)}, \sigma_i^{(m)}$	Mean and standard deviation of reward component m within response group $\mathcal{Y}_i$, respectively.
μ_i, σ_i	Mean and standard deviation of scalar rewards within response group $\mathcal{Y}_i$; σ_i is also a proxy for routing disagreement.
$\bar{\sigma}$	Average intra-group standard deviation within the current batch.
τ_i	Variance-aware scaling factor for query x_i.
$\tau_{\min}, \tau_{\max}$	Lower and upper clipping bounds for τ_i.
(D) Dataset-level Normalization	
$\mathcal{D}^{(d)}$	Set of rollout-response indices (i, j) belonging to dataset d in the current batch.
$\mu^{(d)}, \sigma^{(d)}$	Mean and standard deviation of advantages within dataset d, respectively.