ReCal: Reward Calibration for RL-based LLM Routing

Anonymous ACL submission

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 signals exhibit significant variability across instances, where some instances produce higher or more variable rewards, introducing optimization bias that favors trivial samples over informative ones. 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 variability through variance-aware reweighting and per-dataset normalization. Experiments on seven datasets demonstrate that ReCal consistently improves routing performance, and training stability over baselines[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; Chen and Heinzelman, 2005), achieving promising performance gains. Despite recent progress, existing routing methods often struggle to learn effective policies under queries with varying difficulty and routing uncertainty. In practice, learning signals can exhibit uneven statistical properties across instances and datasets, leading to inconsistent policy updates during training and biased optimization toward easier decision patterns. Consequently, routing policies may fail to learn reliable behaviors for challenging 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, which often correspond to different routing decisions and model-selection strategies into a single score. As shown in Figure 1(a), similar reward improvements may arise from entirely different routing behaviors, 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 produce highly dispersed rewards due to large disagreement among candidate models, while others yield concentrated rewards with limited routing ambiguity. Different datasets further induce substantially different reward scales and variances. Such heterogeneity can bias optimization toward easier routing cases (Schulman et al., 2017; Yu et al., 2025), while

[1]Code: https://anonymous.4open.science/r/ReCal

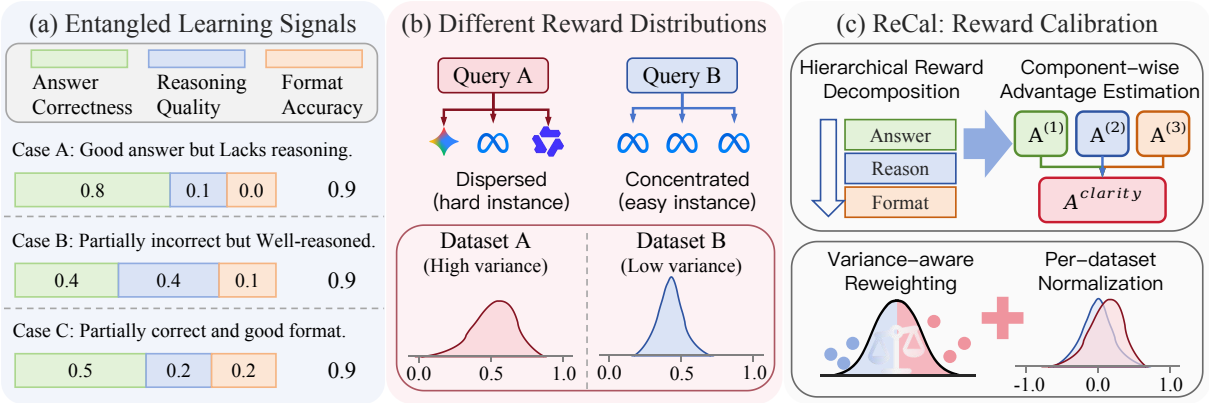

Figure 1: (a) Scalarized rewards entangle multiple objectives, leading to ambiguous credit assignment. (b) Reward distributions vary widely across routing instances and datasets, causing optimization to be dominated by high-variance or easier cases, while informative cases are under-optimized. (c) ReCal provides clear, component-wise learning signals and applies distribution-aware optimization to calibrate policy updates.

informative but underrepresented routing decisions receive insufficient learning attention. These observations raise two central challenges for RL-based LLM routing: **(1)** how to transform entangled routing rewards into clear learning signals for precise routing credit assignment; and **(2)** how to calibrate optimization signals across heterogeneous routing instances and datasets, thereby assigning appropriate emphasis to uncertain and informative routing decisions during optimization.

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, which preserves objective-specific relative improvements during optimization. The resulting component-wise advantages are then combined through weighted aggregation to reflect the relative importance of different objectives while reducing interference among conflicting optimization signals. Building on the calibrated rewards, we further improve signal 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 disagreement among candidate responses. At the dataset level, we apply per-dataset normalization to prevent datasets with disproportionately large reward scales or variances from dominating policy updates. Together, these mechanisms transform routing optimization from directly fitting raw rewards into a calibrated learning process that explicitly separates objective-level supervision from distribution-level optimization variability. This allows the routing policy to focus on informative routing decisions while maintaining stable learning across heterogeneous tasks.

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 under heterogeneous tasks.
- We introduce a hierarchical and disentangled reward mechanism that performs component-wise advantage estimation before aggregation, enabling clearer credit assignment and reducing interference among competing objectives.
- We propose a distribution-aware optimization strategy that reduces optimization bias across heterogeneous data through variance-aware reweighting and per-dataset normalization.
- Extensive experiments on multiple datasets indicate that our method consistently improves routing performance and training stability.

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 optimize routing policies directly with reward signals, enabling adaptive trade-offs between performance and efficiency (Zhang et al., 2025; Chen and Heinzelman, 2005). 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

Amount of RL-based work studies how to construct informative reward design, including reward shaping (Ng et al., 1999), multi-objective RL (Hayes et al., 2022), and advantage-based optimization (Schulman et al., 2017). These approaches aim to improve how different aspects of feedback are represented and utilized during policy updates. However, these approaches mainly focus on improving the semantic quality of reward signals, rather than how heterogeneous reward distributions influence optimization behavior across instances and datasets. 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 and 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 heterogeneous tasks jointly affect both what information learning signals contain and how strongly they influence policy updates.

3 Preliminaries

3.1 RL-based LLM Routing

Given a query x, LLM routing selects an action a from a candidate set $\mathcal{A} = \{a_1, \ldots, a_K\}$ using a routing policy $\pi_\theta(a|x)$ parameterized by θ, where K denotes the number of routing candidates. After executing action $a \sim \pi_\theta(\cdot|x)$, the selected model generates a response y, which is evaluated by a reward function $r(x, a, y)$.

In practical routing systems, rewards are usually composed of multiple objectives, such as answer correctness, reasoning quality, formatting behavior, or routing efficiency. Existing methods typically aggregate them into a scalar reward:

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

where M denotes the number of reward components, $r^{(m)}$ is the reward of the m-th objective, and λ_m is its corresponding weight.

Given a training batch $\mathcal{D} = \{(x_i, a_i, y_i, r_i)\}_{i=1}^{N}$ with batch size N, RL-based routing optimizes the policy by maximizing the expected reward:

$$\max_{\theta} \ \mathbb{E}_{a \sim \pi_\theta(\cdot|x)}[r(x, a, y)]. \tag{2}$$

3.2 Policy Optimization

Most existing routing methods adopt Proximal Policy Optimization (PPO)-style (Schulman et al., 2017) optimization. Let $\pi_{\theta_{\text{old}}}$ denote the old routing policy. PPO optimizes the clipped objective $\mathcal{L}_{\text{PPO}}$:

$$\mathbb{E}\left[\min\left(\rho_t A_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon) A_t\right)\right], \tag{3}$$

where $\rho_t = \frac{\pi_\theta(a_t|x_t)}{\pi_{\theta_{\text{old}}}(a_t|x_t)}$ is the policy ratio for action a_t under query x_t, ϵ is the clipping threshold, and A_t denotes the estimated advantage, which is typically obtained directly from the scalarized reward, i.e., $A_t = \text{Adv}(r_t)$.

Recent RL frameworks further adopt Group Relative Policy Optimization (GRPO)-style optimization. For each query x_i, the policy samples a response group $\mathcal{Y}_i = \{y_{i,1}, \ldots, y_{i,G}\}$ containing G responses, where each response receives a scalar reward $r_{i,j}$. GRPO computes group-relative advantages by normalizing rewards within each group:

$$A_{i,j} = \frac{r_{i,j} - \mu_i}{\sigma_i + \varepsilon}, \tag{4}$$

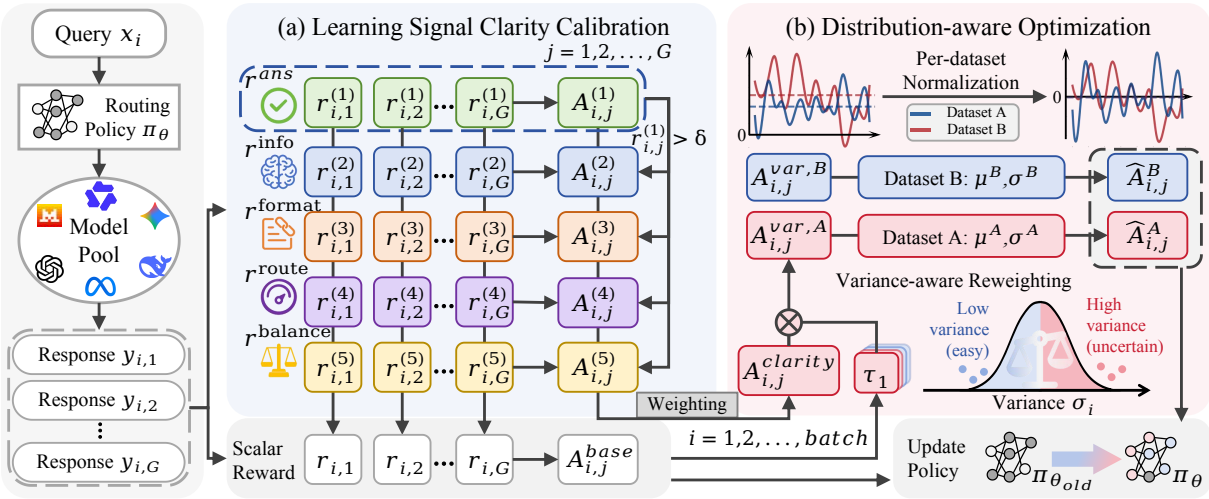

Figure 2: Overview of ReCal. Given a query, the routing policy samples multiple routing actions and obtains a response group. ReCal first decomposes routing rewards into multiple hierarchical objectives and performs component-wise advantage estimation before weighted aggregation. It then calibrates optimization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

where $\sigma_i = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j}-\mu_i)^2}$ and $\mu_i = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}$ denote the standard deviation and group mean, respectively, and ε is a small constant for numerical stability.

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 bias under heterogeneous reward distributions through variance-aware reweighting and per-dataset normalization. The resulting calibrated advantages are finally integrated into a GRPO-style policy optimization objective.

4.1 Learning Signal Clarity Calibration

Rather than estimating advantages directly from aggregated scalarized rewards, ReCal separates reward components before advantage estimation, allowing relative improvements under different objectives to be modeled independently.

4.1.1 Hierarchical Reward Decomposition

For each response $y_{i,j}$ in response group $\mathcal{Y}_i$, ReCal defines a set of reward components: $\mathbf{r}_{i,j} = \{r^{\text{ans}}_{i,j}, r^{\text{info}}_{i,j}, r^{\text{format}}_{i,j}, r^{\text{route}}_{i,j}, r^{\text{balance}}_{i,j}\}$. r^{ans} is computed as the F1 score between the generated answer and the ground-truth. r^{info} measures the quality of returned called-model information by computing the F1 score with the ground-truth. r^{format} is a binary reward indicating whether the response follows the required output format. r^{route} reflects routing efficiency based on the number of external model calls during reasoning. r^{balance} encourages balanced router utilization by rewarding models with lower historical routing frequency. To stabilize long-term routing behavior, the routing frequency statistics are accumulated with exponential moving average (EMA) decay. Let $c_t^{(k)}$ denote the accumulated routing count of model a_k at training step t, and $\hat{c}_t^{(k)}$ denotes the routing count within the current step. The EMA-updated routing statistics are computed as:

$$c_t^{(k)} = \alpha c_{t-1}^{(k)} + \hat{c}_t^{(k)}, \tag{5}$$

where $\alpha \in [0,1)$ is the EMA decay coefficient. 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. Auxiliary rewards are activated only when the response satisfies a correctness threshold:

$$r_{i,j}^{(m)} = \mathbb{I}(r_{i,j}^{\text{ans}} > \delta) \cdot \tilde{r}_{i,j}^{(m)}, \tag{6}$$

where $\tilde{r}_{i,j}^{(m)}$ denotes the raw reward, δ is the activation threshold, and $\mathbb{I}(\cdot)$ is the indicator function.

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.

For the m-th reward objective, the component-wise advantage is computed as:

$$A_{i,j}^{(m)} = \frac{r_{i,j}^{(m)} - \mu_i^{(m)}}{\sigma_i^{(m)} + \varepsilon}, \tag{7}$$

where $\sigma_i^{(m)} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j}^{(m)} - \mu_i^{(m)})^2}$ and $\mu_i^{(m)} = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}^{(m)}$ denote the standard deviation and mean of the m-th reward component within response group $\mathcal{Y}_i$.

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

$$A_{i,j}^{\text{clarity}} = \sum_{m=1}^{M} \lambda_m A_{i,j}^{(m)}, \tag{8}$$

where λ_m denotes the weight of the m-th objective.

Compared with scalarized optimization, this formulation preserves objective-specific relative improvements before aggregation. As a result, the routing policy receives clearer, more discriminative supervision, enabling it to understand which samples are more conducive to policy improvement.

4.2 Distribution-aware Optimization

Although disentangled advantages improve learning signal clarity, optimization can still become biased under heterogeneous reward distributions. ReCal therefore further calibrates optimization variability across both instances and datasets.

4.2.1 Variance-aware Reweighting

For each response group $\mathcal{Y}_i$, with the reward mean $\mu_i^{\text{group}} = \frac{1}{G}\sum_{j=1}^{G} r_{i,j}$, we measures routing uncertainty using the reward variance within the group:

$$\sigma_i^{\text{group}} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_{i,j} - \mu_i^{\text{group}})^2}. \tag{9}$$

Intuitively, larger intra-group variance indicates stronger disagreement among sampled routing decisions, suggesting that the corresponding query is more uncertain and potentially more informative. ReCal therefore rescales optimization strength according to the relative variance level:

$$\tau_i = \text{clip}\left(\frac{\sigma_i^{\text{group}}}{\bar{\sigma}}, \tau_{\min}, \tau_{\max}\right), \tag{10}$$

where $\bar{\sigma}$ denotes the average group variance within the batch, and $\tau_{\min}$ and $\tau_{\max}$ are clipping thresholds. Then the variance-calibrated advantage is computed as:

$$A_{i,j}^{\text{var}} = \tau_i A_{i,j}^{\text{clarity}}. \tag{11}$$

This mechanism allocates larger optimization weights to uncertain routing cases.

4.2.2 Per-dataset Normalization

Different datasets may exhibit substantially different reward magnitudes and variances, causing certain datasets to dominate optimization. To improve optimization comparability across heterogeneous data sources, ReCal normalizes advantages independently within each dataset.

Let $\mathcal{D}^{(d)}$ denote the set of samples from dataset d. We compute dataset-wise statistics:

$$\mu^{(d)} = \frac{1}{|\mathcal{D}^{(d)}|}\sum_{(i,j)\in\mathcal{D}^{(d)}} A_{i,j}^{\text{var}}, \tag{12}$$

$$\sigma^{(d)} = \sqrt{\frac{1}{|\mathcal{D}^{(d)}|}\sum_{(i,j)\in\mathcal{D}^{(d)}} (A_{i,j}^{\text{var}} - \mu^{(d)})^2}. \tag{13}$$

The final calibrated advantage is:

$$\hat{A}_{i,j} = \frac{A_{i,j}^{\text{var}} - \mu^{(d)}}{\sigma^{(d)} + \epsilon}. \tag{14}$$

This normalization aligns optimization scales across datasets and reduces optimization bias caused by heterogeneous reward distributions.

4.3 Overall Training Objective

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

$$\mathbb{E}\left[\min\left(\rho_{i,j}\hat{A}_{i,j}, \text{clip}(\rho_{i,j}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,j}\right)\right], \tag{15}$$

where $\rho_{i,j} = \frac{\pi_\theta(a_{i,j}|x_i)}{\pi_{\theta_{\text{old}}}(a_{i,j}|x_i)}$ denotes the policy ratio between the updated policy and the previous policy.

ReCal explicitly calibrates both learning signal clarity and optimization weighting before policy updates, resulting in more stable and informative policy learning under heterogeneous scenarios.

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).

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**	0.471

Table 1: Experimental results on seven QA datasets w.r.t. Exact Match. **Bold** indicates the best score in each column for each method, and underline indicates the secondary.† indicates in-domain evaluation; all others are out-of-domain.

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 (Qwen 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 and up to 150 training steps. 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 OpenRouter [4]: Qwen2.5-7B-Instruct (Qwen 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), Mixtral8x22B-Instruct (Jiang et al., 2024), and Gemma-2-27B-Instruct (Team et al., 2024).

5.2 Main Results

As shown in Table 1, our proposed ReCal achieves the best average performance among all compared methods on seven QA benchmarks, demonstrating the effectiveness of learning signal calibration for RL-based LLM routing. Compared with non-routing methods, RL-based routing substantially improves performance across all QA tasks, validating the importance of adaptive model selec-

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

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

[4] https://openrouter.ai/

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**

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.

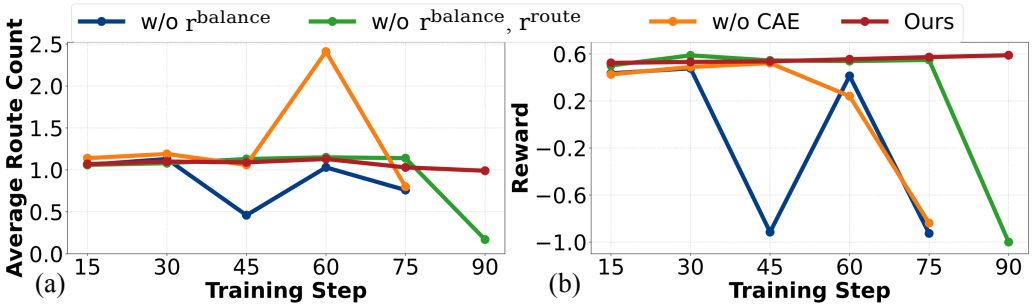

Figure 3: As the training steps increase, (a) and (b) display the average route count and reward for Ours, and for versions that remove $r^{balance}$, $r^{balance}$&r^{route}, and CAE, respectively.

tion. Router-R1 achieves strong gains over heuristic and discriminative routers, showing the advantage of end-to-end policy optimization. However, our method consistently outperforms Router-R1 on most datasets, indicating that improving the clarity and comparability of learning signals is critical for effective routing optimization.

Ours achieves the best overall average EM score of 0.481, surpassing Router-R1 by 6.7% relatively (0.481 vs. 0.414). The improvements are particularly significant 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 calibrating learning signals is especially beneficial for routing policy with high uncertainty.

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 transfer to unseen out-of-domain datasets, indicating that the proposed calibration mechanisms improve the generalization ability of routing policies rather than overfitting to specific training distributions.

We further observe that heuristic and discriminative routing methods exhibit relatively unstable performance across datasets. Prompt LLM and RouterDC perform competitively on certain datasets, such as TriviaQA or Bamboogle, but their overall performance remains substantially lower than RL-based methods. This suggests that heuristics or discriminative routing strategies struggle to adapt to heterogeneous requirements across tasks.

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)** consistently degrades performance across datasets, while removing both causes the largest drop. This verifies that both learning signal clarity and optimization comparability are important for RL-based routing.

Removing LSC decreases the average score to 0.418, indicating that directly optimizing scalarized rewards leads to weaker credit assignment and noisier optimization signals. Removing only Component-wise Advantage Estimation **(CAE)** still causes a noticeable drop to 0.430, demonstrating that component-wise advantage estimation itself is critical for preserving objective-specific supervision during optimization. DAO further improves performance from 0.456 to 0.481, with especially clear gains on challenging multi-hop datasets such as 2Wiki and Musique. This suggests that heterogeneous reward distributions introduce optimization bias during RL training, and distribution-aware calibration helps stabilize learning across different datasets and routing cases.

As Figure 3 shows, removing r^{balance} introduces obvious training instability and leads to a substantial decrease in reward during the optimization, indicating that the balance regularization term is cru-

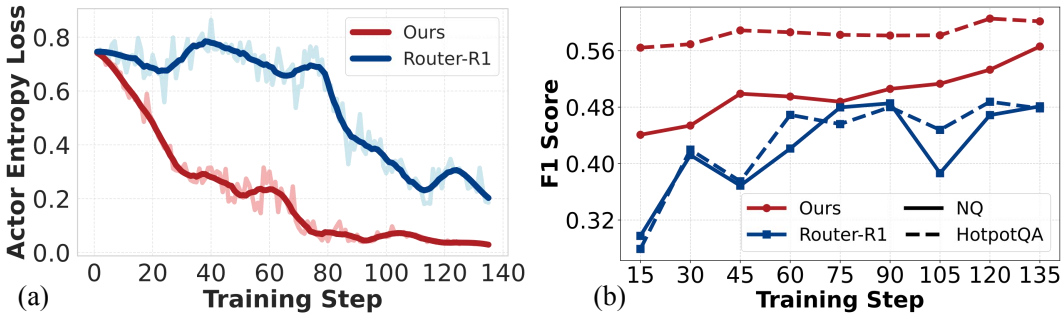

Figure 4: As the training steps increase, (a) shows the entropy loss convergence curves for Ours and Router-R1. (b) shows the F1 scores of Ours and Router-R1 on the NQ and HotpotQA test sets.

cial for preventing the routing policy from collapsing. Removing both r^{balance} and r^{route} also sharply decreases both the routing counts and the reward in the later phases of training. Removing CAE produces a different failure mode: the routing count exhibits abnormal spikes while the reward rapidly collapses later. This indicates that directly estimating advantages from scalarized rewards introduces entangled and unstable optimization signals. In contrast, the full ReCal framework maintains the smoothest routing behavior and the most stable reward improvement throughout training.

5.3.2 Stability and Convergence of Training

Figure 4(a) presents the entropy loss curves during RL training. ReCal exhibits substantially smoother and faster convergence, with entropy steadily decreasing throughout training and converging to a low-variance regime in early stages. In contrast, Router-R1 shows persistent oscillations and slower convergence, indicating unstable policy updates under scalarized reward optimization.

As shown in Figure 4(b), ReCal consistently yields more stable performance gains across both datasets, whereas Router R1 exhibited significant fluctuations and competition across the datasets. In particular, during training steps 60–75, Router-R1 improves on NQ while simultaneously degrading on HotpotQA, suggesting that the optimization process becomes biased toward dominant dataset-specific reward signals. This behavior reveals the existence of cross-dataset optimization interference under heterogeneous reward distributions. By contrast, ReCal jointly improves performance on both datasets throughout training. These results demonstrate that calibrating both reward clarity and optimization distribution improves not only routing performance but also the stability and consistency of policy optimization under heterogeneous tasks.

5.3.3 Generalization to Unseen LLMs

To evaluate the generalization ability of ReCal under expanding model pools, we introduce two additional unseen LLMs, DeepSeek-V3 (DeepSeek-AI et al., 2025) and Qwen2.5-72B-Instruct (Qwen et al., 2025), during inference only. We augment the routing prompt with textual descriptions of the newly models, allowing the routing policy to select them. As shown in Table 1, **Ours+** maintains strong performance and further improves several datasets. This result suggests that ReCal does not overfit to a fixed routing space. Instead, the learned routing policy can generalize to newly introduced models by leveraging semantic descriptions.

6 Conclusion

In this paper, we propose ReCal, a reward calibration framework for RL-based LLM routing. ReCal introduce a hierarchical reward decomposition mechanism with component-wise advantage estimation to provide clearer objective-level supervision, together with a distribution-aware optimization strategy that calibrates optimization variability across heterogeneous instances and datasets through variance-aware reweighting and per-dataset normalization. Extensive experiments demonstrate that ReCal consistently improves routing performance, training stability, and generalization over routing baselines. Further analyses show that ReCal produces more stable optimization under heterogeneous routing scenarios. Beyond the current setting, we believe ReCal provides a promising direction for future RL-based LLM routing. In future work, we plan to extend ReCal to broader routing scenarios. Another important direction is to develop more adaptive and automatic calibration strategies that can dynamically adjust reward structures and optimization variability under evolving model pools and task distributions.

Limitations

ReCal is evaluated primarily on QA-oriented routing tasks, and its effectiveness for other routing settings, such as multimodal systems remains to be explored. Furthermore, ReCal currently relies on manually designed reward components and weighting coefficients, which may require additional tuning when adapting to new tasks or model pools.

Ethical Considerations

We use only publicly available datasets that are widely adopted in the LLM community and contain no personally identifiable information (PII) in our experiments. All evaluated models are accessed through publicly available APIs or open-source checkpoints under their respective licenses.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Lei Chen and W.B. Heinzelman. 2005. Qos-aware routing based on bandwidth estimation for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):561–572.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. 2024. 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. Curran Associates Inc.

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, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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

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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv: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. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1).

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *Preprint*, arXiv:2403.12031.

Changxin Huang, Guangrun Wang, Zhibo Zhou, Ronghui Zhang, and Liang Lin. 2021. Reward-adaptive reinforcement learning: Dynamic policy gradient optimization for bipedal locomotion. *Preprint*, arXiv:2107.01908.

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, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Fengqing Jiang, Zhangchen Xu, Luyao Niu, Boxin Wang, Jinyuan Jia, Bo Li, and Radha Poovendran. 2023. Identifying and mitigating vulnerabilities in llm-integrated applications. *Preprint*, arXiv:2311.16153.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Preprint*, arXiv: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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Patrick Mannion, Sam Devlin, Karl Mason, Jim Duggan, and Enda Howley. 2017. Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing*, 263:60–73. Multiobjective Reinforcement Learning: Theory and Applications.

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

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. Routellm: Learning to route llms with preference data. *Preprint*, arXiv: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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

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. 2025. xrouter: Training cost-aware llms orchestration system via reinforcement learning. *Preprint*, arXiv:2510.08439.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

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

Dimitrios Sikeridis, Dennis Ramdass, and Pranay Pareek. 2025. Pickllm: Context-aware rl-assisted large language model routing. In *International Workshop on AI for Transportation*, pages 227–239. Springer.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artificial Intelligence*, 299:103535.

Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 814–823, Lille, France. PMLR.

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, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. 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. Curran Associates Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Rui Yu, Shenghua Wan, Yucen Wang, Chen-Xiao Gao, Le Gan, Zongzhang Zhang, and De-Chuan Zhan. 2025. Reward models in deep reinforcement learning: a survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25.

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

A 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>

For the question ***"Who is responsible for introducing the principle of uniformitarianism?"***, the correct answer is ***['James Hutton' 'geologist***

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 3: Experimental results on seven QA datasets w.r.t. F1 score. **Bold** indicates the best score in each column for each method, and underline indicates the secondary.† indicates in-domain evaluation; all others are out-of-domain.

James Hutton']. Above are the responses generated by Ours (ReCal) and Router-R1. Our approach ensures the correctness of the answer by invoking different models, whereas Router-R1 easily provides incorrect answers in complex contexts.

<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>

B More Experimental Results

We present extensive experimental results on seven QA datasets with respect to F1-Score in Table 3. ReCal consistently outperforms all baselines across both general and multi-hop QA tasks, achieving the highest average F1-scores. It maintains high accuracy even after introducing unseen models, and achieves state-of-the-art performance.

C Hyperparameter Settings

Hyperparameter	Value
Learning Rate (Actor)	1e-6
Total Batch Size	64
Max Training Steps	150
Max Routing Steps	3
Response group size G	4
GPU Utilization Ratio	0.6
Rollout Sampling Temperature	1.0
EMA decay α	0.9
$\tau_{\min}, \tau_{\max}$	0.8, 1.25

Table 4: Hyperparameter Settings

D Notation

Notation Table	
(A) Routing and Policy Optimization	
x	Input query.
$\mathcal{A} = \{a_1, \ldots, a_K\}$	Candidate routing action set, where K is the number of candidates.
a	Routing action sampled from the routing policy.
$\pi_\theta(a\|x)$	Routing policy parameterized by θ.
θ	Parameters of the routing policy.
y	Response generated after selecting routing action a.
$\mathcal{D} = \{(x_i, a_i, y_i, r_i)\}_{i=1}^N$	Training batch with batch size N.
$\rho_{i,j} = \frac{\pi_\theta(a_{i,j}\|x_i)}{\pi_{\theta_{\text{old}}}(a_{i,j}\|x_i)}$	PPO/GRPO probability ratio between updated and old policies.
ϵ	Clipping threshold in PPO/GRPO optimization.
ε	A small constant for numerical stability.
$\mathcal{L}_{\text{ReCal}}$	Final GRPO-style training objective of ReCal.
(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.
$r(x, a, y)$	Scalar reward evaluating the routing decision and generated response.
$r^{(m)}$	Reward component of the m-th objective.
M	Number of reward components.
λ_m	Weight assigned to the m-th reward component.
$\mathbf{r}_{i,j}$	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-efficiency reward related to external model invocation frequency.
$r_{i,j}^{\text{balance}}$	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_t	Standard advantage estimate used in PPO-style optimization.
$A_{i,j}^{(m)}$	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.
$\mu_i^{(m)}$	Mean reward of the m-th component within response group $\mathcal{Y}_i$.
$\sigma_i^{(m)}$	Standard deviation of the m-th reward component within response group $\mathcal{Y}_i$.
μ_i	Mean scalar reward within response group $\mathcal{Y}_i$.
σ_i^{group}	Intra-group reward standard deviation used to measure routing uncertainty.
$\bar{\sigma}$	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 samples belonging to dataset d.
$\mu^{(d)}$	Mean advantage value within dataset d.
$\sigma^{(d)}$	Standard deviation of advantages within dataset d.
d	Dataset index.

Table 5: Summary of notations used in ReCal.