Before Thinking, Learn to Decide: Proactive Routing for Efficient Visual Reasoning

Yinan Zhou[1,2,3], Haokun Lin[2,3,4], Yichen Wu[5], Caifeng Shan[4], Zhenan Sun[6], Yuxin Chen[2], Teng Wang[2], Chen Ma[3,‡], Li Zhu[1,‡], and Ying Shan[2]

[1] Xi'an Jiaotong University
[2] ARC Lab, Tencent IEG
[3] City University of Hong Kong
[4] Institute of Automation, CAS
[5] Harvard University
[6] Nanjing University

Abstract. Large multimodal models have achieved strong reasoning on complex visual tasks, but their inference efficiency is often restricted by long chains of thought. A promising solution is to pair a small draft model with a large target model, enabling cooperative inference employing a routing signal that adaptively routes queries to either the draft or target model based on their difficulties for optimal efficiency and accuracy. Yet, the remaining bottleneck is to establish a reliable query difficulty signal under multimodal settings. Existing approaches designed for language models either rely on post-hoc token probabilities, which fall short in multimodal scenarios, or depend on supervised fine-tuning, which is a data-sensitive strategy. Both paradigms perform routing only after a complete output, and ignore whether the target model can actually solve the routed instances. To address this, we propose **PRP**, a **P**roactive **R**outing **P**aradigm that enables early decision-making by jointly evaluating the competence of both the draft and target models. Our **D**raft **R**ating **L**earning **(DRL)** equips the draft model with an internal confidence estimator, while **J**oint **R**ating **L**earning **(JRL)** predicts how well the target model can handle a given query, thereby prioritizing the allocation of samples it excels at rather than the hardest ones. These ratings enable fine-grained, instance-level **Proactive Routing** and substantially accelerate inference without compromising overall performance. Extensive experiments across multiple multimodal reasoning benchmarks validate our effectiveness and efficiency.

Keywords: Efficient MLLM Routing· Confidence Learning · GRPO

1 Introduction

Multimodal large language models (MLLMs) [1, 9, 12, 19, 23, 43–45, 49–51] have achieved remarkable performance on visual reasoning tasks. With the emergence

Work done during internship at Tencent. ‡Corresponding authors.

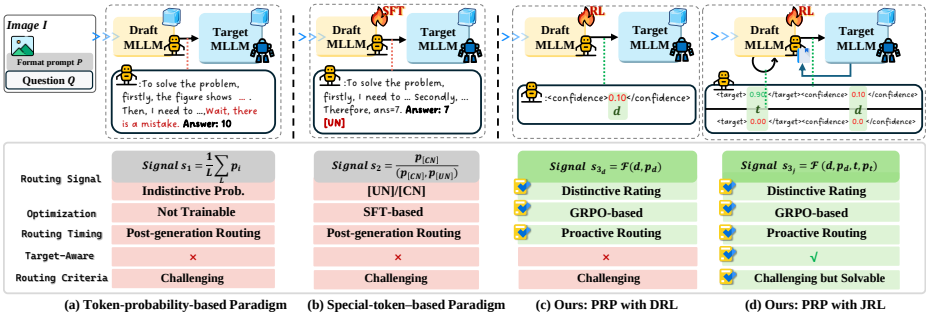

Fig. 1: Overview comparing two prior paradigms with ours. We enable proactive routing at the onset of inference. Our RL-trained rater provides finer-grained and distinctive signals, and explicitly accounts for the capability of the target model.

of reinforcement learning–based post-training methods [6,35,47], their reasoning capabilities are further strengthened, exhibiting emergent behaviors such as "aha moments" and self-reflection. However, these powerful reasoning abilities often come with lengthy and repetitive chains of thought. Such long reasoning are unnecessary for simple problems and introduce substantial inference overhead.

A natural way to reduce inference cost while preserving strong reasoning performance is to use two models of different capacities, a compact draft model that routes difficult queries to a larger target model. Several studies explore the draft-routing paradigms for LLMs, which fall into two categories. As illustrated in Fig. 1: (i) token-probability-based routing [30], which averages token probabilities as the routing signal; (ii) special-token-based routing [10] fine-tunes tokens such as `[CN]` and `[UN]` using correct/incorrect trajectories and routes queries based on their probabilities.

However, directly applying these routing methods to MLLMs remains underexplored. Our preliminary experiments reveal several issues. First, token-probability-based routing provides coarse and indistinct signals, relying on an assumed positive correlation between token probability and accuracy. Second, special-token-based routing is highly sensitive to the sampling distribution of correct and incorrect trajectories, and SFT-based methods are limited by the model's search space. Moreover, both paradigms defer routing decisions until the draft model finishes generation, preventing early invocation of the target model and adding latency. Importantly, they only assess difficulty from the draft model's perspective, ignoring whether the target model is actually suitable for the instance.

To overcome these limitations, we propose a *reinforcement learning-based* **P**roactive **R**outing **P**aradigm (**PRP**) for efficient visual reasoning. PRP initiates routing at the very beginning of generation and is aware of both the draft and target models' capabilities. We first introduce a fine-grained proactive rating training framework consisting of **D**raft **R**ating **L**earning (**DRL**) and **J**oint **R**ating **L**earning (**JRL**). DRL enables the draft model to directly assess its solvability on an input before generating any reasoning steps, making proactive routing feasible. To further evaluate the capacity of the target model, JRL ad-

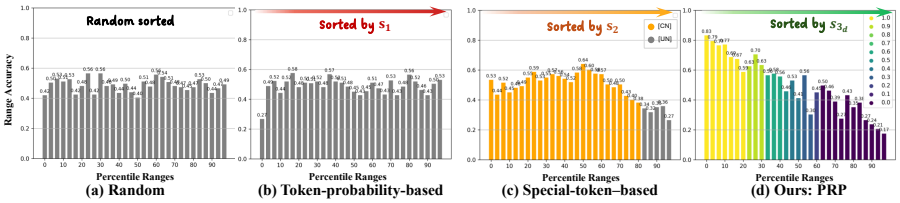

Fig. 2: Draft model's percentile-range accuracy comparison of signal ranking across 3 paradigms versus random on MathVerse. Our PRP exhibits the performance retention and distinctive rating distribution, laying a solid foundation for fine-grained routing.

ditionally predicts the target model's suitability for each instance, allowing the system to route not only difficult problems but also those that the target model handles particularly well. We then develop a three-stage score-based **Proactive Routing** scheme for efficient MLLM inference. The system first uses DRL or JRL trained models to produce draft and target ratings. These ratings are then used to rank instances using different score-based strategies. DRL model relies on fine-grained confidence distributions, while JRL model jointly evaluates the capacities of both the draft and target models to perform routing. Our contributions are summarized as follows:

- We are the first to explore proactive rating for reasoning MLLMs. We propose two GRPO-based rating learning methods that enable models to assess both their own solvability and the target model's capacity for given inputs.
- We design several score-based inference strategies for efficient visual reasoning, jointly considering the abilities of both draft and target models.
- Our methods perform quick routing in the initial stage and deliver competitive reasoning performance with query-difficulty perception, while significantly reducing computation. Notably, JRL could achieve 2.41× acceleration on MathVista while even slightly outperforming the stronger target model.

2 Analysis of Previous Routing Paradigm

Since previous routing methods are primarily designed for LLMs, we first conduct preliminary experiments by directly applying representative approaches to the multimodal reasoning LLMs. Discussed in Sec. 2.1, this leads to several notable insights, further highlighting limitations in Sec. 2.2.

2.1 Observations

Token-probability-based Routing Fails in Multimodal. Token-probability-based Routing Paradigm [30] relies on the output token-level probability distribution without training: it computes the mean probability of $s_1 = \frac{1}{L}\sum_{i=1}^{L} p_i$ over the generated sequence of length L as a routing signal. A low $s1$ is interpreted as low confidence, triggering delegation to the target model. This strategy

is coarse-grained with indistinctive signal and hinges on a presumed positive correlation between $s1$ and accuracy. However, we find that this correlation breaks down in multimodal settings with long visual reasoning: as shown in Fig. 2(a)(b), when sampling responses on MathVerse [48] and sorting them by descending $s1$, we observe no clear monotonic relationship between $s1$ and accuracy.

Special-token-based Routing Paradigm Exhibits Sensitivity to SFT Data. Special-token-based Routing [10] starts with a pre-sampling stage before training for supervised fine-tuning data preparation. During pre-sampling, correct completions are attached by a `[CN]` token and incorrect ones by a `[UN]` token, enabling the representation of the correctness of its own reasoning. During training, for the correct sequence, the entire output sequence is directly fine-tuned, including the response and the `[CN]` token. For the incorrect sequence, only the `[UN]` token is fine-tuned to prevent performance decline. At inference time, the special token probabilities at the sequence tail provide the routing signal $s_2 = \frac{p_{[CN]}}{p_{[CN]}+p_{[UN]}}$.This approach requires supervised fine-tuning on both positive and negative trajectories with special tokens, which in turn restricts the model's search space. Moreover, its performance is highly sensitive to the sampling distribution of positive versus negative examples in the dataset. As shown in Fig. 2(c), after trained on diverse math data MMK12 [32], the draft model still struggles to reliably predict `[CN]` token for the complex thinking.

2.2 Common Limitations

Beyond the observations discussed above, these case-level routing paradigms share several common shortcomings. First, both paradigms defer routing decisions until the draft model has generated the entire response. As a result, the target model cannot be invoked earlier, which restricts potential efficiency gains and introduces unnecessary latency. Second, they fail to evaluate whether the selected target model is actually suitable for the given query. For instances that exceed the target model's capability, routing them to the target model leads only to repetitive or low-value reasoning, offering limited performance improvement while constraining inference speed. These limitations motivate our proactive routing paradigm for efficient visual reasoning.

3 Training Method

3.1 Preliminary

Group Relative Policy Optimization (GRPO). As a post-training method, GRPO improves reasoning by comparing multiple completions per sample, scoring format compliance and answer accuracy, and using the reward differences to compute advantages. In the multimodal setting, the policy model π_θ and reference model $\pi_{\theta_{old}}$ are initialized with pretrained weights. Given an input $x = (P, I, Q)$, where P is a predefined formatting prompt and (I, Q) is an

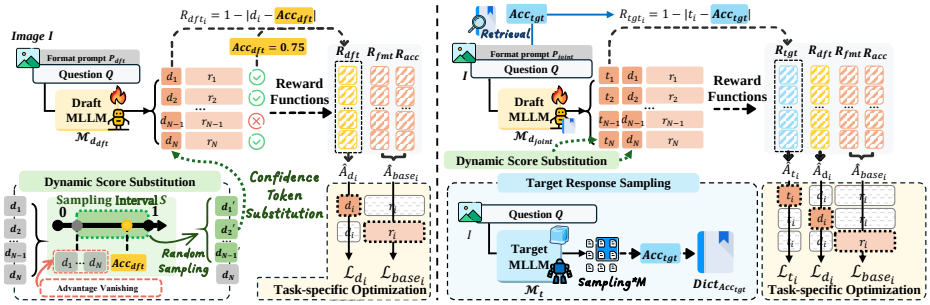

Fig. 3: Left: Illustration of Draft Rating Learning. We design reward R_{dft} and adopt Dynamic Score Substitution to avoid advantage vanishing, while Task-specific Optimization independently optimizes the R_{dft}-related tokens during training. Right: Illustration of Joint Rating Learning with the target model. We extend the Draft Rating Learning pipeline by introducing R_{tgt}, which is trained with pre-sampled Acc_{tgt} as supervision, thereby accounting for the capability of the target model.

image-question pair from dataset $\mathcal{D}$, the reference model $\pi_{\theta_{\text{old}}}$ samples N candidate outputs $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$. For each candidate o_i, we compute a format reward R_{fmt_i} and an accuracy reward R_{acc_i}, and define the total reward: $R_{base_i} = R_{fmt_i} + R_{acc_i}$. Within each group of N candidates, we estimate the advantage of o_i relative to the group average,

$$\hat{A}_{\text{base}_i} = \frac{R_{\text{base}_i} - \text{mean}\big(\{R_{\text{base}_j}\}_{j=1}^{N}\big)}{\text{std}\big(\{R_{\text{base}_j}\}_{j=1}^{N}\big)}, \tag{1}$$

Here, the mean and standard deviation are computed over the N candidates in the same group.

Given these advantages, we apply a single-update loss similar to standard GRPO without KL penalty or clipping:

$$\mathcal{L}_{\text{base}_i}(\theta) = -\mathbb{E}_m \left[\frac{\pi_\theta(o_{i,m}|x, o_{i,<m})}{\pi_{\theta_{\text{old}}}(o_{i,m}|x, o_{i,<m})} \right] \hat{A}_{\text{base}_i}, \tag{2}$$

where $o_{i,<m}$ is the prefix up to token m. Building on this objective, we extend GRPO to endow the model with confidence estimation for multimodal inputs before producing long responses, which supports both self-evaluation and cross-model evaluation.

3.2 Draft Rating Learning (DRL)

As shown in Fig. 3(left), we introduce a formatted prompt P_{dft} that instructs the draft model $\mathcal{M}_{d_{dft}}$ to produce a draft score before the think phase begins. This score reflects the model's self-assessed confidence for the input x. The output of $\mathcal{M}_{d_{dft}}$, denoted o_i, contains two parts: d_i, the draft score, and r_i, the remaining segment that includes the format markers, the reasoning, and the final answer.

To train this capability effectively, we define a reward for draft rating learning, adopt a dynamic score substitution strategy to prevent advantage vanishing, and apply task-specific optimization to preserve overall performance.

Draft Rating Reward. To train the draft model to rate both the input x and its own capability, we define the Draft Rating Reward. Given N sampled outputs $\mathcal{O} = \{o_i\}_{i=1}^{N}$, we first estimate the accuracy on this set. Specifically, we compute the mean draft accuracy as,

$$\mathrm{Acc}_{\mathrm{dft}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\{\mathrm{correct}(o_i, \mathcal{A})\}, \tag{3}$$

where $correct(o_i, \mathcal{A})$ denotes that the answer extracted from o_i is consistent with the answer $\mathcal{A}$ of (I, Q).

We then utilize Acc_{dft} as the ground-truth label for the draft rating d_i. In this way, for inputs that are simple and within the model's competence, d_i should approach 1; conversely, for inputs that are difficult or consistently produce incorrect solutions, d_i should approach 0. Accordingly, we define a simple yet effective draft rating reward:

$$R_{\mathrm{dft}_i} = \begin{cases} 1 - |d_i - \mathrm{Acc}_{\mathrm{dft}}|, & \text{if } 0 \leqslant d_i \leqslant 1 \\ -1, & \text{otherwise} \end{cases} \tag{4}$$

To keep rewards comparable across samples and to prioritize cases with larger score discrepancies, we compute the advantage $\hat{A}_{d_i}$ by mean-centering without standard-deviation scaling,

$$\hat{A}_{d_i} = R_{\mathrm{dft}_i} - \frac{1}{N}\sum_{j=1}^{N} R_{\mathrm{dft}_j} \tag{5}$$

Dynamic Score Substitution. During training, we observe that the draft scores $\{d_1, \ldots, d_N\}$ produced by the model often collapse to nearly identical values. As a result, both the reward and advantage diminish, and the model stops learning a meaningful rating. We define *Advantage Vanishing* to occur when $|mean(\{d_i\}_{i=1}^{N}) - Acc_{dft}| > \mu$ and $std(\{d_i\}_{i=1}^{N}) < \sigma$, where μ and σ are hyperparameters. In this regime, the model tends to overuse locally high-probability tokens when predicting scores. To counteract this, we propose Dynamic Score Substitution. Specifically, we construct a sampling interval $\mathcal{S}$ as,

$$\begin{aligned} \mathcal{S} \in [\,&\max(0, \min(d_N, \mathrm{Acc}_{\mathrm{dft}}) - \left|\mathrm{Acc}_{\mathrm{dft}} - d_N\right|), \\ &\min(\max(d_N, \mathrm{Acc}_{\mathrm{dft}}) + \left|\mathrm{Acc}_{\mathrm{dft}} - d_N\right|, 1)]. \end{aligned}$$

Within $\mathcal{S}$, we randomly sample $\{d'_1, \ldots, d'_{N-1}\}$ to substitute $\{d_1, \ldots, d_{N-1}\}$ in $\mathcal{O}$. By construction, each $d_i^{'}$ lies closer to Acc_{dft} than d_i, thereby yielding a higher reward $R_{dft_i}^{'}$. To amplify the advantage of the substituted scores while reducing the advantage of d_N, we replace $\hat{A}_{d_i}$ with $\hat{A}_{d_i}^{'}$ as

$$\hat{A}'_{d_i} = \begin{cases} R'_{\text{dft}_i} - R_{\text{dft}_N}, & i \in \{1, \ldots, N-1\} \\ -\sum_{j=1}^{N-1} \hat{A}'_{d_j}, & i = N \end{cases} \tag{6}$$

Task-specific Optimization. Since $\hat{A}_{base_i}$ is to optimize format and correctness, and $\hat{A}_{d_i}$ is to accurately assess the existing model's capabilities, directly mixing $\hat{A}_{base_i}$ and $\hat{A}_{d_i}$ to optimize all output tokens o_i can lead to damage to the original performance and inefficiency in rating learning. Therefore, we propose Task-specific Optimization. Specifically, we optimize the corresponding token parts, d_i and r_i, using $\hat{A}_{d_i}$ and $\hat{A}_{base_i}$, respectively, with weight α:

$$\begin{aligned} \mathcal{L}_{\text{dft}_i}(\theta) &= \mathcal{L}_{\text{base}_i}(\theta) + \alpha \mathcal{L}_{d_i}(\theta) \\ &= -\mathbb{E}_m \frac{\pi_\theta(r_{i,m}|x, d_{i,<m}, r_{i,<m})}{\pi_{\theta_{\text{old}}}(r_{i,m}|x, d_{i,<m}, r_{i,<m})} \hat{A}_{\text{base}_i} \\ &\quad - \alpha \mathbb{E}_m \frac{\pi_\theta(d_{i,m}|x, d_{i,<m}, r_{i,<m})}{\pi_{\theta_{\text{old}}}(d_{i,m}|x, d_{i,<m}, r_{i,<m})} \hat{A}_{d_i}. \end{aligned} \tag{7}$$

3.3 Joint Rating Learning with Target model (JRL)

As shown in Fig. 3(right), P_{joint} requires the model to output a target score t_i prior to draft score d_i and think phase, which denotes the target model's assessment of the input x. Different from Eq. (3), which calculates Acc_{dft} during training, we perform M sampling iterations on each sample in $\mathcal{D}$ using the target model before training the draft model, and record the average accuracy of M outputs in $Dict_{Acc_{tgt}}$. During training, we retrieve the corresponding Acc_{tgt} as the ground-truth label of t_i. We apply formulas similar to Eq. (4) and Eq. (5) to obtain R_{tgt_i} and $\hat{A}_{t_i}$, and employ Dynamic Score Substitution to avoid advantage vanishing, along with task-specific optimization with weight β for target rating training:

$$\begin{aligned} \mathcal{L}_{\text{joint}_i}(\theta) &= \mathcal{L}_{\text{base}_i}(\theta) + \alpha \mathcal{L}_{d_i}(\theta) + \beta \mathcal{L}_{t_i}(\theta) \\ &= -\mathbb{E}_m \frac{\pi_\theta(r_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})}{\pi_{\theta_{\text{old}}}(r_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})} \hat{A}_{\text{base}_i} \\ &\quad - \alpha \mathbb{E}_m \frac{\pi_\theta(d_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})}{\pi_{\theta_{\text{old}}}(d_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})} \hat{A}_{d_i} \\ &\quad - \beta \mathbb{E}_m \frac{\pi_\theta(t_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})}{\pi_{\theta_{\text{old}}}(t_{i,m}|x, t_{i,<m}, d_{i,<m}, r_{i,<m})} \hat{A}_{t_i}. \end{aligned} \tag{8}$$

4 Proactive Routing Method

In this section, we first establish the correlation between the predicted ratings and accuracy. Building on this, we then design a three-stage proactive routing

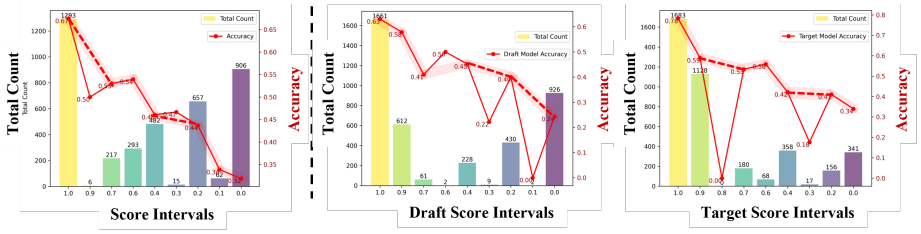

Fig. 4: Left: The per-score instance count and accuracy of $\mathcal{M}_{d_{dft}}$ on MathVerse. Right-1: $\mathcal{M}_{d_{joint}}$'s draft score distribution on MathVerse and the draft's corresponding accuracy. Right-2: The target score distribution and $\mathcal{M}_t$'s corresponding accuracy. In all plots, bars denote the number of samples in each score bin, and lines indicate the accuracy within each bin.

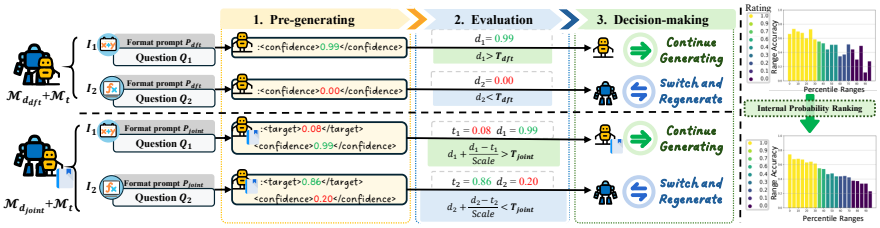

Fig. 5: Left: Three-stage Score-based Proactive Routing. Right: Internal Probability Ranking exhibits a strong correlation with accuracy.

scheme that delivers efficient acceleration while preserving performance. Moreover, we introduce a fine-grained partitioning method based on the token-level probabilities of predicted ratings, thereby enabling a more precise evaluation stage. We further analyze the superiority and practicality of our routing method.

4.1 Correlation between Ratings and Accuracy

As shown in Fig. 4, we analyze the ratings produced by $\mathcal{M}_{d_{dft}}$ (trained in Sec. 3.2) and $\mathcal{M}_{d_{joint}}$ (trained in Sec. 3.3) on MathVerse [48]. The left panel reports the relationship between the ratings from $\mathcal{M}_{d_{dft}}$ during inference and accuracy. The two right panels, under $\mathcal{M}_{d_{joint}}$, respectively show (i) the draft model's score bins and its corresponding accuracies, and (ii) $\mathcal{M}_t$'s score bins and the target model's accuracies. Across all settings, we observe a clear trend: higher score bins correspond to higher correctness. Minor deviations in a few bins arise primarily from limited sample counts within those intervals. Overall, the yellow bins exhibit substantially higher accuracy than the purple bins, demonstrating that our approach learns meaningful, discriminative ratings that correlate strongly with accuracy. We further demonstrate the advantages of our ratings by the correlation of interval accuracy and ratings in Tab. 1.

4.2 Three-stage Score-based Proactive Routing

Our Proactive Routing strategy involves $\mathcal{M}_{d_{dft}}$ and $\mathcal{M}_{d_{joint}}$ as routing models and follows the rating ranking in Sec. 4.1. It proceeds in three stages:

- **Pre-generating:** Before the model starts thinking and answering, we prompt $\mathcal{M}_{d_{dft}}$ or $\mathcal{M}_{d_{joint}}$ to generate a short prefix until a complete rating snippet is obtained. From this snippet, we extract: (i) the self-assessed confidence d of $\mathcal{M}_{d_{dft}}$; (ii) both the confidence d of $\mathcal{M}_{d_{joint}}$ and the predicted confidence t of $\mathcal{M}_t$.
- **Evaluation:** When $\mathcal{M}_{d_{dft}}$ serves as the routing model, we directly employ d as the score. When $\mathcal{M}_{d_{joint}}$ serves as the routing model, we combine d and t to reflect two priorities: (i) routing instances that the draft model can solve correctly to the draft model; (ii) preferring the draft model when $\mathcal{M}_t$ is likely to fail due to capability mismatch. Concretely, for datasets where easy instances can be solved with high confidence, we adopt a draft-rating-first scoring strategy rule and set $s = d + \frac{d-t}{\tau}$; whereas for more challenging datasets-where the system has generally low confidence, we adopt target-rating-first scoring rule and set $s = t + \frac{t-d}{\tau}$, where τ controls the priority given to capability gap between $\mathcal{M}_{d_{joint}}$ and $\mathcal{M}_t$.
- **Decision-making:** With $\mathcal{M}_{d_{dft}}$ as the router, if $d > T_{dft}$, we route the instance to $\mathcal{M}_{d_{dft}}$ to continue generating a full response; otherwise, we route it to $\mathcal{M}_t$. With $\mathcal{M}_{d_{joint}}$ for routing, challenging but solvable samples have lower scores compared to overly difficult ones, which means we prioritize assigning more suitable hard samples to the target model.If $s{>}T_{joint}$, $\mathcal{M}_{d_{joint}}$ is confident, and we route to $\mathcal{M}_{d_{joint}}$ and continue generating. If $s{<}T_{joint}$, $\mathcal{M}_{d_{joint}}$ is uncertain with lower expected accuracy, and we route to $\mathcal{M}_t$.

4.3 Internal Probability Ranking

Relying solely on the rating score to partition samples is too coarse; many examples cluster within the same score band, leaving no principled way to prioritize them. As shown in the upper-right of Fig. 5, we first sort MathVerse samples by score and measure range accuracy across percentiles. Although the high-score (yellow) region covers a large proportion, its internal accuracy is highly heterogeneous. To resolve this, we propose Internal Probability Ranking, which refines prioritization within the same score band by using the logits of the rating tokens.

Concretely, considering a score such as "0.98", we examine the token position corresponding to "9" in the rating string and extract its logits vector z over the full vocabulary. Let $D = \{0, 1, \ldots, 9\}$ denote the digit-token set, and let p_i be the probability assigned to digit i under the full-vocabulary softmax at that position, i.e., $p_i = \text{softmax}(z)_i$ for $i \in D$. Let $p_{max} = max_{i \in D} p_i$. We then define the within-band priority score as:

$$p = \begin{cases} \frac{1}{|D|} p_{max} \sum_{i \in D} (p_i - \frac{1}{|D|} \sum_{j \in D} p_j)^2, \text{if } d > 5 \\ -\frac{1}{|D|} p_{max} \sum_{i \in D} (p_i - \frac{1}{|D|} \sum_{j \in D} p_j)^2, \text{else} \end{cases} \tag{9}$$

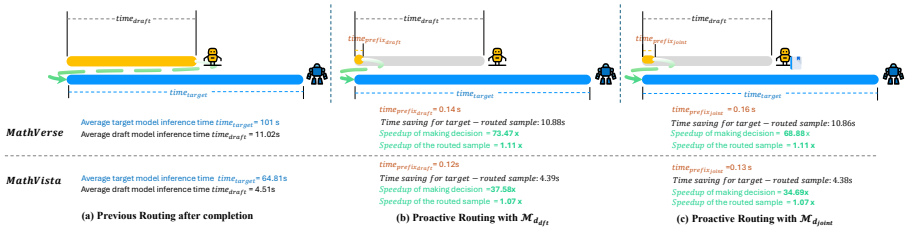

Fig. 6: Comparison of decision latency between traditional routing-after-completion and our proactive routing schemes ($\mathcal{M}_{d_{dft}}$ and $\mathcal{M}_{d_{joint}}$).

Intuitively, when scores are high, a more concentrated digit distribution together with a larger p_{max} indicates greater certainty, which yields a larger p and correlates with higher accuracy; for low scores, the trend reverses. Finally, we sort samples within each score band by p. As shown in the lower-right of Fig. 5, the interal accuracy then exhibits a clear stepwise pattern, which enables precise and flexible routing. We implement this by augmenting the ratings with their within-band priorities, setting $d' = d + p_d$ and $t' = t + p_t$. Replacing the second-stage evaluation rating d and t with d' and t' yields finer-grained assessments and more reliable routing within score ranges.

4.4 Superiority and Practicality of PRP

Fig. 6 compares the end-to-end latency of routing schemes on routed-to-target cases. In the conventional "route-after-completion" setting, the total time is $time_{draft} + time_{target}$. With proactive routing, the draft model only generates a short prefix before making a routing decision, so the total time becomes $time_{prefix_{draft}} + time_{target}$. This saves $time_{draft} - time_{prefix_{draft}}$ per routed example. On both MathVista and MathVerse, both DRL and JRL save nearly the entire inference time of the draft model, thereby advancing the routing decision to the initial stage of inference. This enables the swift redirection of challenging inputs to a more powerful target model, meaning our decision lead time is dozens of times earlier compared to post-inference routing. We provide two easy strategies for threshold setting as examples to illustrate the practicality of the routing method: *(a) Task-Specific Calibration with Validation Set:* For specific domains with a validation set, such as mathematical reasoning in MathVista, we use the validation set routing curve to find the point that achieves the best trade-off between performance and efficiency. Then, we can obtain the rating of the sample at that point. We route samples based on this rating as a threshold to achieve similar performance and speedup on test data. In Fig. 8(f), we use 30% of MathVista for validation and 70% for testing. Similar trends in the curves indicate the effectiveness of this strategy. *(b) Dynamic Pre-scoring without Validation Set:* In the absence of a validation set, we process concurrent requests as a batch at the same time. For query batches, we perform a rapid proactive rating (taking approximately 1% of the total draft latency shown in Fig. 6). By sorting the ratings within a batch, we determine the switching threshold based on limited

computational resources, allowing us to dynamically allocate these examples to maximize performance within the constraints of available resources.

5 Experiments

5.1 Setup

Evaluation Tasks. We evaluate the effectiveness of PRP across two representative domains. We first validate it in a simple question-answering setting, using 15K ChartQA [31] training subset for training and 2.5K test data for in-domain evaluation. The second domain targets mathematical reasoning: we simply employ 15K MMK12 [32] for confidence learning. We adopt two widely used benchmarks that span a broad range of problem types and difficulty levels, MathVista [28] and MathVerse [48]. We also provide the results on M3CoT [7] in **??** to further illustrate our effectiveness and generalization.
Reasoning MLLMs. For QA domain, we directly train Qwen2.5-VL-3B [1] with DRL to obtain $M_{d_{dft}}$. We use Ovis-2.5-9B [29] as the target model. For math domain, we first perform GRPO to empower Qwen2.5-VL-7B [1] with strong reasoning ability, then perform DRL and JRL on the GRPO-trained model for better efficiency, while Skywork-R1V-38B [41] is adopted as $\mathcal{M}_t$.
Implementation Details. α and β are set to 1.5. Rollout N is set to 4 and sample number M for JRL is 16. $M_{d_{dft}}$ and $M_{d_{joint}}$ are trained for 3 epochs with batch-size of 140 and a learning rate of $5e-7$. For evaluation, the thresholds T_{dft} and T_{joint} are controlled by the fraction of samples routed to M_t (p_t%).
Baselines. To the best of our knowledge, we are the first to introduce instance-level routing for multimodal reasoning. Accordingly, we compare against a random routing baseline and two paradigms described in Sec. 2:

- **Random**. We use our trained $\mathcal{M}_{d_{dft}}$ or $\mathcal{M}_{d_{joint}}$ to inference and assign instances randomly to draft and target according to p_t%.
- **Token-probability-based** ($Paradigm_1$). On ChartQA, we adopt Qwen2.5-VL-3B as the baseline; on MathVerse and MathVista, we use the GRPO-trained model as the baseline. For each example, we compute s_1, sort instances by s_1, and perform routing based on this ranking.
- **Special-token-based** ($Paradigm_2$). On ChartQA, we initialize training from Qwen2.5-VL-3B; on MathVerse and MathVista, we initialize from the GRPO-trained model. For training, each question follows the same sampling budget as our main training setup: we sample N times per question and append `[CN]` for correct and `[UN]` for incorrect, to perform SFT for 3 epochs. At inference, we rank instances by s_2, and route accordingly.

5.2 Overall Rating and Routing Performance Analysis

As shown in Tab. 1, we evaluate the overall rating and routing performance of various paradigms across 3 benchmarks. **Overall Rating Performance:** We calculates the Spearman correlation coefficient between the ratings assigned by

Table 1: Spearman Correlations between rating rank and accuracy show the superiority of our ratings. Area Under the Curve(AUC) for routing demonstrates the overall routing effectiveness.

Dataset	ChartQA					MathVerse					MathVista				
Paradigm	Random	P_1	P_2	DRL	JRL	Random	P_1	P_2	DRL	JRL	Random	P_1	P_2	DRL	JRL
Spearman of Rank&Acc↓	0.00	-0.34	-0.25	**-0.49**	**-0.50**	-0.01	-0.01	-0.11	**-0.32**	**-0.30**	0.00	0.12	-0.19	**-0.34**	**-0.35**
Area Under Curve(%)	78.87	80.74	73.30	**81.97**	**83.61**	56.13	56.72	56.03	55.44	**56.97**	72.09	72.69	72.19	**73.29**	**73.36**

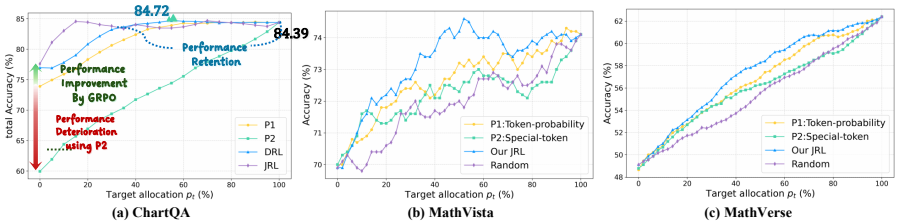

Fig. 7: Detailed comparison of our proposed routing with random and 2 previous paradigm on ChartQA, MathVista, and MathVerse.

the draft model and the actual accuracy (ranked from correct to incorrect). A stronger negative correlation indicates that the ratings more accurately reflects its actual performance. Our DRL and JRL consistently achieve higher correlations, proving that our ratings are more closely aligned with true model correctness. **Overall Routing Performance:** Area Under the Curve (AUC) is derived from the routing performance curve across various allocation ratios between the draft and target models (See Fig. 7). This reflects the overall effectiveness of the routing process across all possible target allocation p_t levels. We outperform alternative methods in nearly all scenarios, with the notable exception of DRL on MathVista. In this specific case, the DRL performance even falls below the random setting. This occurs because there are instances where the draft model is incapable of solving the task, but the target model also fails; conversely, in tasks where the draft model performs well, the target model exhibits even greater superiority. Because DRL focuses solely on assessing the draft model's capabilities while ignoring the target model's potential, it may incorrectly prioritize samples that would have provided a higher marginal contribution to overall accuracy if assigned differently. We provide a more granular analysis in **??**.

5.3 Detailed Evaluation Results on ChartQA

As shown in Fig. 7(a), we evaluate in-domain DRL-trained $\mathcal{M}_{d_{dft}}$'s and JRL-trained $\mathcal{M}_{d_{joint}}$'s routing capability using scores with Internal Probability Ranking. Compared to $Paradigm_1$ directly applied to the Qwen2.5-VL-3B model, our DRL-trained $\mathcal{M}_{d_{dft}}$ not only improves model performance but also produces reliable scores, enabling efficient routing. By contrast, $Paradigm_2$ exhibits a much more severe performance degradation. Notably, DRL can stably maintain, and

Table 2: Latency and Accuracy evaluations of our DRL and JRL along with baseline methods on MathVista and MathVerse benchmarks.

Model	MathVista				MathVerse			
	Draft $(1-p_t)\%$	Accuracy	Avg. Latency(s)	Speedup	Draft $(1-p_t)\%$	Accuracy	Avg. Latency(s)	Speedup
$\mathcal{M}_t$**(Skywork-R1V-38B)**	-	74.1%	64.8	-	-	62.4%	101.4	-
DRL-trained $\mathcal{M}_{d_{dft}}$ **(7B)**	100%	69.9%	10.2	-	100%	50.0%	11.8	-
	20%	75.4%	55.5	1.17x	20%	59.8%	86.4	1.17x
	30%	75.3%	53.5	1.21x	30%	58.7%	78.1	1.30x
	50%	74.3%	43.6	1.49x	40%	57.2%	69.1	1.47x
JRL-trained $\mathcal{M}_{d_{joint}}$ **(7B)**	100%	70.0%	4.9	-	100%	49.9%	10.9	-
	30%	74.4%	47.6	1.36x	20%	61.2%	82.1	1.24x
	40%	74.1%	40.1	1.62x	25%	60.77%	76.6	1.32x
	50%	74.0%	32.6	1.99x	30%	60.4%	71.5	1.42x
	60%	74.2%	26.9	2.41x	50%	58.5%	55.4	1.83x
Random	60%	71.6%	27.5	2.36x	50%	56.4%	56.0	1.81x
Token-probability-based ($Paradigm_1$)	60%	72.2%	29.5	2.19x	50%	56.9%	56.3	1.80x
Special-token-based ($Paradigm_2$)	60%	72.1%	26.9	**2.41x**	50%	56.3%	55.6	1.82x
Ours	60%	**74.2%**	**26.9**	**2.41x**	50%	**58.5%**	**55.4**	**1.83x**

even enhance, the target model performance from 84.39 to 84.72 when $p_t > 40$, which demonstrate the feasibility and reliability of our PRP on simple in-domain tasks. JRL can also stably maintain the target model performance when $p_t > 15$.

5.4 Detailed Evaluation Results on Math Domain

We present the detailed results of our Joint Rating Learning with Target Model (JRL) strategy compared with Random, $Paradigm_1$, and $Paradigm_2$ in Fig. 7. Our JRL consistently outperforms all other strategies across different target allocation ratios, with even larger gains when more queries are routed to the target model. This demonstrates that JRL is a more effective routing mechanism for MLLMs and directly addresses the limitations discussed in Sec. 2.2.

Efficiency Measurements. Table 2 summarizes the performance and speedup of our DRL and JRL strategies on the MathVista and MathVerse benchmarks. For Draft Rating Learning (DRL), we observe that the overall performance with DRL-based routing achieves lossless performance on MathVista with approximately 1.5× acceleration, demonstrating the effectiveness of confidence learning. With JRL, the routing performance improves substantially once the target model's capability is incorporated. For example, on MathVerse, $\mathcal{M}_{d_{joint}}$ maintains a competitive 60.4% accuracy with a 1.42× speedup. On MathVista, JRL performs even better: at a 60% routing ratio, $\mathcal{M}_{d_{joint}}$ surpasses $\mathcal{M}_t$ while achieving a 2.41× speedup.

5.5 Ablation Studies

Internal Probability Ranking (IPR). We validate the effectiveness of IPR on ChartQA and MathVista. Specifically, we compare (i) routing based solely on raw confidence ratings, and (ii) routing with additional Internal Probability Ranking within each score range. As shown in Fig. 8(a)(b), the latter consistently achieves better routing quality due to the more orderly score distribution within each bin. This demonstrates that incorporating internal probability provides finer-grained confidence calibration for the draft model, leading to improved performance.

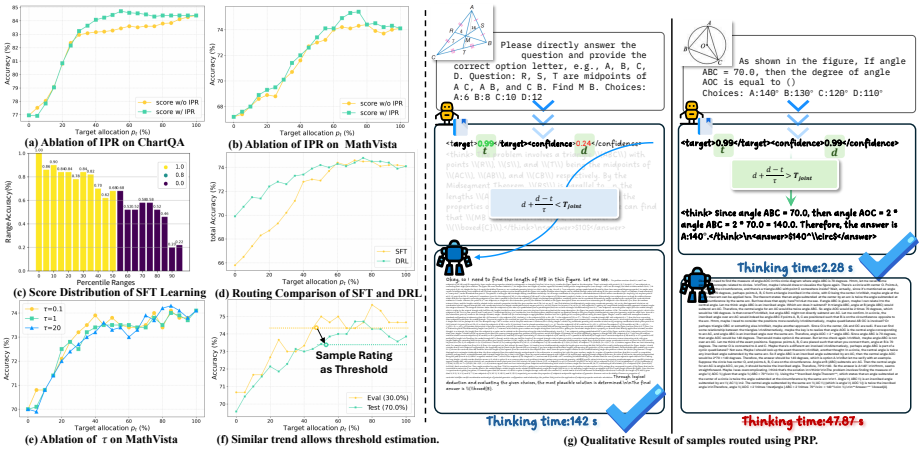

Fig. 8: Ablation Study: (a)(b) Ablation study of Internal Probability Ranking (IPR) on ChartQA and MathVista.(c) Score Distribution of SFT Rating Learning. (d) Routing Comparison of SFT and DRL. (e). $\mathcal{M}_{d_{joint}}$'s ablation of different τ for the scaling in signal $s = d + \frac{d-t}{\tau}$ on MathVista. (f) Similar accuracy trends across evaluation and test sets allows threshold estimation. (g) Qualitative Result of samples routed using PRP.

RL-based rating training vs SFT-based rating training. As a comparison, we use the accuracy within the group as the label and employ Supervised Fine-tuning for rating training. Compared to our rich scoring scenarios, the SFT model produces polarized scores (only 0/1) during inference (see Fig. 8(c)), resulting in a weaker correlation with actual performance than our RL-based method. Meanwhile, the draft accuracy on MathVista drops by 4.2%(see Fig. 8(d)).
Different scale τ for $\mathcal{M}_{d_{joint}}$. As discussed in Sec. 4.2, when employing $\mathcal{M}_{d_{joint}}$ for routing, the scaling coefficient τ controls the importance of the gap between $\mathcal{M}_{d_{joint}}$ and $\mathcal{M}_t$ during ranking. As shwon in Fig. 8(e),When τ is large (blue curve), routing focuses more on $\mathcal{M}_{d_{joint}}$'s intrinsic capability, thus preferentially assigning instances where $\mathcal{M}_{d_{joint}}$ excels to the $\mathcal{M}_{d_{joint}}$; under this setting, $p_t \in [80, 100]$ exhibits strong performance preservation. Conversely, when τ is small (yellow curve), routing emphasizes the discrepancy between $\mathcal{M}_{d_{joint}}$ and $\mathcal{M}_t$, preferentially preserving instances where $\mathcal{M}_t$ excels to $\mathcal{M}_t$; as a result, we achieve faster performance gains in $p_t \in [0, 50]$.
Qualitative Result. In Fig. 8(g), PRP routing proceeds once module $\mathcal{M}_{d_{dft}}$ produces a self-score d and an assessment t of $\mathcal{M}_t$. More results are in **??**.

6 Related Work

6.1 Reinforcement Learning for Reasoning Models

Reinforcement learning has been widely adopted to enhance the reasoning capabilities of LLMs. Methods such as PPO [38] and DPO [37] are commonly used

in post-training by maximizing reward functions. GRPO [18, 39] was proposed to improve reasoning by optimizing group advantages using reward differences across multiple completions, while subsequent works [24, 27, 32, 46] explore generalized optimization strategies. GRPO-like approaches have been extended to multimodal domains, where customized reward functions are designed for various vision-language tasks [5, 8, 13, 25, 26, 40, 47]. We leverage GRPO to preserve draft model's performance while equipping the model with proactive rating ability, thereby laying the foundation for flexible routing.

6.2 Efficient Reasoning

LLM reasoning often requires long chains of thought, leading to increased latency [36]. A common solution is to pair a small draft model with a larger target model. Speculative decoding [3,4,14,20,21,33] is one representative approach, but applying it to reasoning models introduces token-distribution mismatch across intermediate steps. Progress reward models (PRMs) [22, 42] attempt to assign reasoning steps to draft or target models, but they are difficult to train for MLLMs and incur additional cost. Router-based methods [2, 10, 15, 16, 34] rely on evaluating the model's confidence [17], and offer another direction by selecting models according to query difficulty or budget. For example, Self-REF [10] uses SFT to train a confidence token, while others [11] benchmark uncertainty estimation for reasoning LLMs. However, their effectiveness for MLLMs remains unclear; they typically rely on full responses, preventing early routing, and ignore the target model's solvability. In this work, we propose to estimate confidence *before* reasoning, enabling MLLMs to decide when and whom to ask.

7 Conclusion

We present a RL-based Proactive Routing Paradigm for efficient MLLM reasoning. By introducing Draft Rating Learning and Joint Rating Learning, we enables models to estimate both their own and the target model's solvability of the input before generation, making proactive and capability-aware routing possible. Building on robust rating model, our three-stage PRP enables fine-grained routing that prioritizes samples for which the target model is expected to excel, rather than simply forwarding the most difficult cases, yielding a favorable balance between performance and efficiency across the draft and target models.

Acknowledgements

This work is supported by National Key Research and Development Program of China (2023YFC3321600), the Early Career Scheme (No.CityU 21219323) and the General Research Fund (No.CityU 11220324) of the University Grants Committee (UGC), the NSFC Young Scientists Fund (No.9240127), and the Donation for Research Projects (No.9229164 and No.9229216).

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Barrak, A., Fourati, Y., Olchawa, M., Ksontini, E., Zoghlami, K.: Cargo: A framework for confidence-aware routing of large language models (2025), https://arxiv.org/abs/2509.14899
3. Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J.D., Chen, D., Dao, T.: Medusa: Simple llm inference acceleration framework with multiple decoding heads. arXiv preprint arXiv:2401.10774 (2024)
4. Chen, C., Borgeaud, S., Irving, G., Lespiau, J.B., Sifre, L., Jumper, J.: Accelerating large language model decoding with speculative sampling. arXiv preprint arXiv:2302.01318 (2023)
5. Chen, L., Gao, H., Liu, T., Huang, Z., Sung, F., Zhou, X., Wu, Y., Chang, B.: G1: Bootstrapping perception and reasoning abilities of vision-language model via reinforcement learning (2025), https://arxiv.org/abs/2505.13426
6. Chen, L., Li, L., Zhao, H., Song, Y., Vinci: R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V (2025), accessed: 2025-02-02
7. Chen, Q., Qin, L., Zhang, J., Chen, Z., Xu, X., Che, W.: M^3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In: Proc. of ACL (2024)
8. Chen, Y., Ge, Y., Wang, R., Ge, Y., Cheng, J., Shan, Y., Liu, X.: Grpo-care: Consistency-aware reinforcement learning for multimodal reasoning. arXiv preprint arXiv:2506.16141 (2025)
9. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024)
10. Chuang, Y.N., Sarma, P.K., Gopalan, P., Boccio, J., Bolouki, S., Hu, X., Zhou, H.: Learning to route llms with confidence tokens. arXiv preprint arXiv:2410.13284 (2024)
11. Chuang, Y.N., Yu, L., Wang, G., Zhang, L., Liu, Z., Cai, X., Sui, Y., Braverman, V., Hu, X.: Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. arXiv preprint arXiv:2502.04428 (2025)
12. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
13. Du, L., Meng, F., Liu, Z., Zhou, Z., Luo, P., Zhang, Q., Shao, W.: Mm-prm: Enhancing multimodal mathematical reasoning with scalable step-level supervision. arXiv preprint arXiv:2505.13427 (2025)
14. Elhoushi, M., Shrivastava, A., Liskovich, D., Hosmer, B., Wasti, B., Lai, L., Mahmoud, A., Acun, B., Agarwal, S., Roman, A., et al.: Layerskip: Enabling early exit inference and self-speculative decoding. arXiv preprint arXiv:2404.16710 (2024)
15. Feng, T., Shen, Y., You, J.: Graphrouter: A graph-based router for llm selections. arXiv preprint arXiv:2410.03834 (2024)
16. Fu, T., Ge, Y., You, Y., Liu, E., Yuan, Z., Dai, G., Yan, S., Yang, H., Wang, Y.: R2r: Efficiently navigating divergent reasoning paths with small-large model token routing. arXiv preprint arXiv:2505.21600 (2025)

17. Geng, J., Cai, F., Wang, Y., Koeppl, H., Nakov, P., Gurevych, I.: A survey of confidence estimation and calibration in large language models. In: Duh, K., Gomez, H., Bethard, S. (eds.) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 6577–6595. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024). https://doi.org/10.18653/v1/2024.naacl-long.366, https://aclanthology.org/2024.naacl-long.366/
18. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
19. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
20. Leviathan, Y., Kalman, M., Matias, Y.: Fast inference from transformers via speculative decoding. In: International Conference on Machine Learning. pp. 19274–19286. PMLR (2023)
21. Li, Y., Wei, F., Zhang, C., Zhang, H.: Eagle: Speculative sampling requires rethinking feature uncertainty. arXiv preprint arXiv:2401.15077 (2024)
22. Liao, B., Xu, Y., Dong, H., Li, J., Monz, C., Savarese, S., Sahoo, D., Xiong, C.: Reward-guided speculative decoding for efficient llm reasoning. arXiv preprint arXiv:2501.19324 (2025)
23. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023)
24. Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W.S., Lin, M.: Understanding r1-zero-like training: A critical perspective. arXiv preprint arXiv:2503.20783 (2025)
25. Liu, Z., Sun, Z., Zang, Y., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual-rft: Visual reinforcement fine-tuning. arXiv preprint arXiv:2503.01785 (2025)
26. Liu, Z., Zang, Y., Zou, Y., Liang, Z., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual agentic reinforcement fine-tuning (2025), https://arxiv.org/abs/2505.14246
27. Liu, Z., Meng, F., Du, L., Zhou, Z., Yu, C., Shao, W., Zhang, Q.: Cpgd: Toward stable rule-based reinforcement learning for language models. arXiv preprint arXiv:2505.12504 (2025)
28. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In: International Conference on Learning Representations (ICLR) (2024)
29. Lu, S., Li, Y., Xia, Y., Hu, Y., Zhao, S., Ma, Y., Wei, Z., Li, Y., Duan, L., Zhao, J., et al.: Ovis2. 5 technical report. arXiv preprint arXiv:2508.11737 (2025)
30. Mahaut, M., Aina, L., Czarnowska, P., Hardalov, M., Müller, T., Màrquez, L.: Factual confidence of llms: on reliability and robustness of current estimators. arXiv preprint arXiv:2406.13415 (2024)
31. Masry, A., Do, X.L., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In: Findings of the association for computational linguistics: ACL 2022. pp. 2263–2279 (2022)
32. Meng, F., Du, L., Liu, Z., Zhou, Z., Lu, Q., Fu, D., Han, T., Shi, B., Wang, W., He, J., Zhang, K., Luo, P., Qiao, Y., Zhang, Q., Shao, W.: Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. arXiv preprint arXiv:2503.07365 (2025)

33. Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Wang, Z., Zhang, Z., Wong, R.Y.Y., Zhu, A., Yang, L., Shi, X., et al.: Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3. pp. 932–949 (2024)
34. Ong, I., Almahairi, A., Wu, V., Chiang, W.L., Wu, T., Gonzalez, J.E., Kadous, M.W., Stoica, I.: Routellm: Learning to route llms with preference data. arXiv preprint arXiv:2406.18665 (2024)
35. Peng, Y., Zhang, G., Zhang, M., You, Z., Liu, J., Zhu, Q., Yang, K., Xu, X., Geng, X., Yang, X.: Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. arXiv preprint arXiv:2503.07536 (2025)
36. Qu, X., Li, Y., Su, Z., Sun, W., Yan, J., Liu, D., Cui, G., Liu, D., Liang, S., He, J., et al.: A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. arXiv preprint arXiv:2503.21614 (2025)
37. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems **36**, 53728–53741 (2023)
38. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
39. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
40. Shen, H., Liu, P., Li, J., Fang, C., Ma, Y., Liao, J., Shen, Q., Zhang, Z., Zhao, K., Zhang, Q., et al.: Vlm-r1: A stable and generalizable r1-style large vision-language model. arXiv preprint arXiv:2504.07615 (2025)
41. Shen, W., Pei, J., Peng, Y., Song, X., Liu, Y., Peng, J., Sun, H., Hao, Y., Wang, P., Zhang, J., Zhou, Y.: Skywork-r1v3 technical report (2025), https://arxiv.org/abs/2507.06167
42. Wang, Z., Wang, J., Pan, J., Xia, X., Zhen, H., Yuan, M., Hao, J., Wu, F.: Accelerating large language model reasoning via speculative search. arXiv preprint arXiv:2505.02865 (2025)
43. Yang, S., Zhou, Y., Zheng, Z., Wang, Y., Zhu, L., Wu, Y.: Towards unified text-based person retrieval: A large-scale multi-attribute and language search benchmark. In: Proceedings of the 31st ACM International Conference on Multimedia. p. 4492–4501. MM '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3581783.3611709, https://doi.org/10.1145/3581783.3611709
44. Yang, Y., Zhou, Y., Chen, Y., Zhang, Z., Ma, Z., Yuan, C., Li, B., Gao, J., Hu, W.: Beyond semantic search: Towards referential anchoring in composed image retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 31155–31165 (June 2026)
45. Yang, Y., Zhou, Y., Chen, Y., Zhang, Z., Ma, Z., Yuan, C., Li, B., Song, L., Gao, J., Li, P., Hu, W.: Detailfusion: A dual-branch framework with detail enhancement for composed image retrieval (2025), https://arxiv.org/abs/2505.17796
46. Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)
47. Zhang, J., Huang, J., Yao, H., Liu, S., Zhang, X., Lu, S., Tao, D.: R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. arXiv preprint arXiv:2503.12937 (2025)

48. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Gao, P., et al.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? arXiv preprint arXiv:2403.14624 (2024)
49. Zhou, Y., Chen, Y., Lin, H., Wu, Y., Yang, S., Qi, Z., Ma, C., Zhu, L.: Dogr: Towards versatile visual document grounding and referring. In: 2025 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3596–3606 (2025). https://doi.org/10.1109/ICCV51701.2025.00343
50. Zhou, Y., Wang, Y., Lin, H., Ma, C., Zhu, L., Zheng, Z.: Scale up composed image retrieval learning via modification text generation. IEEE Transactions on Multimedia **27**, 7510–7521 (2025). https://doi.org/10.1109/TMM.2025.3599088
51. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)