Confidence-Guided Stepwise Model Routing for Cost-Efficient Reasoning

Sangmook Lee*, **Dohyung Kim***, **Hyukhun Koh**, **Nakyeong Yang**, **Kyomin Jung**†

Seoul National University
{helmsman, kimdohyung, hyukhunkoh-ai, yny0506, kjung}@snu.ac.kr

Abstract

Recent advances in Large Language Models (LLMs) - particularly model scaling and test-time techniques - have greatly enhanced the reasoning capabilities of language models at the expense of higher inference costs. To lower inference costs, prior works train router models or deferral mechanisms that allocate easy queries to a small, efficient model, while forwarding harder queries to larger, more expensive models. However, these trained router models often lack robustness under domain shifts and require expensive data synthesis techniques such as Monte Carlo rollouts to obtain sufficient ground-truth routing labels for training. In this work, we propose Confidence-Guided **Ste**pwise Model Routing for Cost-**E**fficient **R**easoning (STEER), a domain-agnostic framework that performs fine-grained, step-level routing between smaller and larger LLMs *without* utilizing external models. STEER leverages confidence scores from the smaller model's logits prior to generating a reasoning step, so that the large model is invoked only when necessary. Extensive evaluations using different LLMs on a diverse set of challenging benchmarks across multiple domains such as Mathematical Reasoning, Multi-Hop QA, and Planning tasks indicate that STEER achieves competitive or enhanced accuracy while reducing inference costs (up to $+20\%$ accuracy with 48% less FLOPs compared to solely using the larger model on AIME), outperforming baselines that rely on trained external modules. Our results establish model-internal confidence as a robust, domain-agnostic signal for model routing, offering a scalable pathway for efficient LLM deployment.

1 Introduction

Recent LLMs demonstrate remarkable abilities in complex, non-trivial tasks such as mathematical reasoning, multi-hop reasoning, and code generation. Their rapid progress can be largely attributed to scaling laws (Kaplan et al. 2020; Sardana et al. 2024), in which language models obtain power-law improvements with increased model parameters and data. Exploiting such phenomena has pushed current LLMs into the billion-parameter regime.

However, such exploitation of scaling laws also led to expensive inference costs, as FLOPs per inference token scale with parameter count (Kaplan et al. 2020; Sardana et al. 2024). Furthermore, in the context of LLM reasoning, the problem of expensive deployment is exacerbated by test-time techniques such as Self-Consistency (Wang et al. 2023), Tree Search (Yao et al. 2023), and more recently, extended test-time computation using RL techniques (Guo et al. 2025) - all of which substantially increase the number of tokens generated per reasoning query. For sustainability and scalability of LLM deployment, addressing the escalating cost of LLM inference has become a timely and pressing challenge (Faiz et al. 2024; Poddar et al. 2025).

To mitigate such issues, a promising research direction in the domain of cost-efficient inference is adaptive inference using multiple models, where simple questions are allocated to small, cheaper models, and complex questions are allocated to larger, more expensive models. Previous works (Chen, Zaharia, and Zou 2023; Gupta et al. 2024; Ong et al. 2024; Ding et al. 2024; Damani et al. 2025) train router models or deferral mechanisms that adaptively allocate an appropriate model at the question-level. While such methods are effective in providing a good quality-cost balance, the routers, being lightweight transformers trained on a specific dataset, show limited performance in out-of-domain inputs (Ong et al. 2024; Ding et al. 2024). Furthermore, generating the data to train the router typically requires expensive Monte Carlo rollouts to obtain golden router labels (Damani et al. 2025; Ding et al. 2024), or large-scale data-augmentation methods using LLMs (Ong et al. 2024). We argue that, although the router may be lightweight, the computational cost of data collection required to train the router is substantial and should not be dismissed as a sunk cost.

In the narrower domain of cost-efficient LLM reasoning, recent works utilize a more fine-grained, step-level signal: RSD (Liao et al. 2025) leverages Process Reward Models (PRMs) to guide step-level routing through rejection sampling. Although such works offer fine-grained signals that are well-aligned with the step-by-step reasoning paradigm of LLMs, the reliance on trained PRMs limits applicability predominantly to the domain of mathematical reasoning (Zeng et al. 2025) and to the LLMs that the PRM is trained on (Zhu et al. 2025). Such drawbacks of previous works that rely on trained external models motivate an external model-free framework that is robust to domain shifts.

In this paper, we propose Confidence-Guided **Ste**pwise

---

*These authors contributed equally.

†Corresponding author.

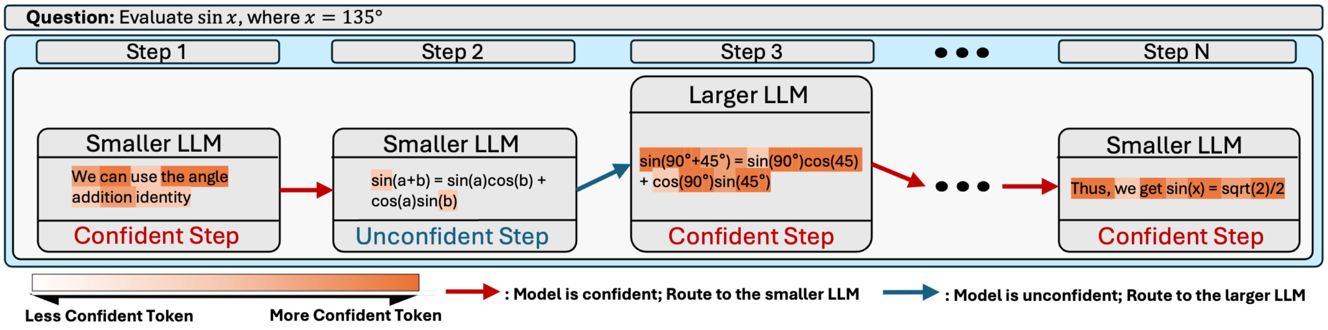

Figure 1: **STEER** uses model confidence to route between the smaller LLM and the larger LLM in a step-wise fashion to achieve a good balance between performance and inference costs. More specifically, the larger LLM is invoked for the generation of the next step only if the smaller model is unconfident in generating the next step.

Model Routing for Cost-**E**fficient **R**easoning (STEER), a novel domain-agnostic, external model-free framework. STEER leverages logit-based confidence scores to dynamically route between smaller and larger models at the step level, reducing inference costs while preserving the reasoning performance of the larger model. This is motivated by the observation that model confidence is strongly correlated with output correctness (Wang and Zhou 2024; Ma et al. 2025c,a). To assess each model's reliability throughout the reasoning process, we calibrate these confidence scores using a Gaussian Mixture Model (GMM). As illustrated in Figure 1, STEER evaluates model confidence at every step to determine whether the smaller model is sufficiently capable of handling the current reasoning step, or whether the complexity of the task necessitates invoking the larger model. Notably, in contrast to prior works, STEER provides fine-grained, step-level routing signals *without* relying on external supervision or the training of auxiliary models such as PRMs. This enables broad applicability across diverse reasoning tasks, without the need for additional data collection or fine-tuning.

Through extensive experiments, we demonstrate that STEER achieves significantly reduced inference cost with minimal performance degradation across diverse domains. Furthermore, STEER is robust to changes in the backbone LLM family, allowing it to generalize effectively across different architectures, substantially broadening its practical utility compared to prior works.

Contributions Our contributions are as follows:

- We propose STEER, a domain-agnostic, external model-free framework for cost-efficient reasoning. Our novel framework leverages logit-based confidence estimation to adaptively route between a larger model and a smaller model at each reasoning step, at test time.
- We propose a posterior-based routing strategy leveraging mixture models further to enhance the effectiveness of confidence estimation for routing.
- We conduct extensive experiments across a diverse set of benchmarks, including MATH500, AIME, Omni-Math, ACPBench, MuSiQue, and KOR-Bench. Across the benchmarks, compared to the larger model, STEER reduces inference costs by 10% to 48%, and on AIME, STEER achieves +20% accuracy with 48% less FLOPs.

2 Preliminaries

2.1 Step-Level Model Switching Formulation

In the context of reasoning, an LLM is fed an input question x and iteratively generates a sequence of reasoning steps $s_1, \ldots, s_n$, which we denote as $s_{1:n}$. Each step s_i is composed of k_i tokens. Formally, the step-by-step generation of the reasoning steps using autoregressive models can be described by the following equation:

$$s_i \sim p_\theta(\cdot \mid x, s_{1:i-1}), \tag{1}$$

where p_θ represents the probability distribution from an LLM with parameter θ.

For computational efficiency, works such as RSD (Liao et al. 2025) and SpecReason (Pan et al. 2025) utilize a step-level model switching formulation. These works involve adaptively assigning an larger, more capable model or a smaller, less capable model for the generation of each step. Formally, under this formulation, the generation of the reasoning steps can be described by:

$$s_i \sim p_{\theta_i}(\cdot \mid x, s_{1:i-1}), \tag{2}$$

where $\theta_i \in \{\theta_M, \theta_m\}$ defines the model chosen for the generation of step i, and θ_M and θ_m represents the larger model and the smaller model, respectively. In this work, we make use of a GMM that take in as input model confidence scores to choose between θ_M and θ_m at every step, invoking the larger model only when necessary in a step-wise fashion.

2.2 Gaussian Mixture Models

A two-component Gaussian mixture has density

$$\begin{aligned} &P(r \mid \boldsymbol{\Theta}) = \pi_1 \mathcal{N}(r; \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(r; \mu_2, \sigma_2^2), \\ &\pi_1 + \pi_2 = 1, \end{aligned} \tag{3}$$

where r is a scalar variable, $\boldsymbol{\Theta} = \{\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$ with $\pi_1, \pi_2 \geq 0$ and $\sigma_1^2, \sigma_2^2 > 0$. The parameters of the GMM can be obtained using the EM Algorithm. Details on the EM algorithm are in Appendix E.

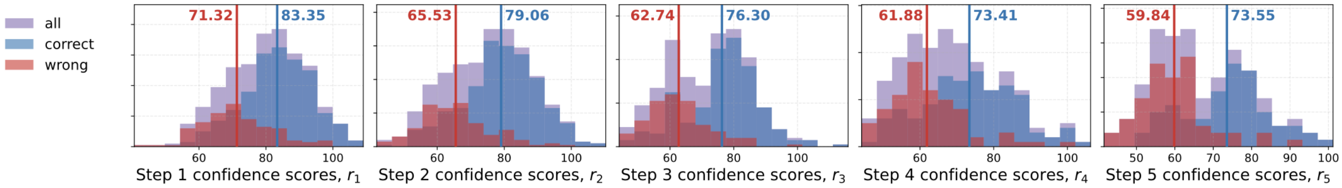

Figure 2: Distributions of the confidence scores on correct and wrong reasoning trajectories for MATH500 on Gemma-3-Instruct 4B. The means of correct and wrong cases are shown with respective colors.

Posterior assignment Let $z \in \{1, 2\}$ denote the latent membership indicator variable. For a new observation r, the posterior probability of membership in component 1 is:

$$P\big(z=1 \mid r, \boldsymbol{\Theta}\big) = \frac{\pi_1 \mathcal{N}\big(r; \mu_1, \sigma_1^2\big)}{\pi_1 \mathcal{N}\big(r; \mu_1, \sigma_1^2\big) + \pi_2 \mathcal{N}\big(r; \mu_2, \sigma_2^2\big)} \tag{4}$$

The posterior for component 2 is obtained similarly.

3 STEER

We propose a step-level routing framework that adaptively distributes computation between the smaller and larger models. Let θ_M and θ_m be parameters of the smaller and the larger models, respectively, and $r(x, s_{1:i})$ denote the reward of a sequence of steps $s_{1:i}$, given a question x. Following previous works (Liao et al. 2025), we assume that the larger model outperforms the small model in expected reward:

$$\mathbb{E}_{s_i \sim p_{\theta_M}}\left[r(x, [s_{<i}; s_i])\right] \geq \mathbb{E}_{s_i \sim p_{\theta_m}}\left[r(x, [s_{<i}; s_i])\right] \tag{5}$$

Under this assumption, routing to the larger model at any given step increases the expected reward. However, under the setting of cost-efficient reasoning, such a routing decision incurs additional computational costs that may be unnecessary. Therefore, an effective routing algorithm must selectively invoke the larger model only when the small model's expected reward is low, aiming to optimize the trade-off between performance and inference cost.

3.1 Logit-based Confidence Estimation

Following previous works demonstrating that the correctness of a model is closely related to its confidence (Xie et al. 2023; Wang and Zhou 2024; Wang et al. 2023), we introduce a confidence-based method to estimate whether a model possesses sufficient capability to generate a correct next step. Specifically, we leverage logit values as a measure of confidence (Wang and Zhou 2024; Ma et al. 2025a). To quantify the confidence of a reasoning step, we first define the token-level confidence score for a single token, and then define the step-level confidence by aggregating token-level scores.

Token-Level Confidence. Let t_{ij} be the j-th token of the i-th reasoning step. Let $z_{ij} \in \mathbb{R}^V$ denote the vector of vocabulary logits output by a language model when it generates the token t_{ij}, where V is the dimension of the vocabulary space of the model. We define the token-level confidence score $\phi_{ij} \in \mathbb{R}^1$ of the token t_{ij} under the language model as follows:

$$\phi_{ij} := \max(z_{ij}) \tag{6}$$

Step-Level Confidence. The stepwise confidence score for step i, denoted as $\Phi_i \in \mathbb{R}^1$, is then computed by applying an aggregation function g over the sequence of tokens and their confidence scores associated with step i:

$$\Phi_i := g\left(\phi_{i1}, \phi_{i2}, \ldots, \phi_{iL_i}\right), \tag{7}$$

where L_i denotes the token length of the i-th reasoning step.

The choice of the aggregation function g depends on the reasoning domain. We evaluate STEER on two domains: mathematical reasoning and broader reasoning tasks, such as multi-hop QA and planning. In the mathematical domain, we define g as taking the average of the token-level confidence scores over tokens corresponding to mathematical expressions, as mathematical symbols are more critical for solution correctness (Lin et al. 2025) in mathematical reasoning. For broader reasoning tasks, we define g as taking the average token-level confidence over all tokens in a step.

3.2 Adaptive Routing via Mixture Models

Having defined stepwise confidence scores, we now present our method for adaptive routing between smaller and larger models. Interestingly, distributions of step-level confidence scores for step i for a group of problems, $\mathbf{P}_{\Phi_i}$ (we omit the step index i hereafter in this section for notational simplicity), exhibit a bi-modal distribution, implying that the overall distribution can be decomposed into a mixture two simpler distributions (See Figure 2). Following this observation, we model the distribution of confidence scores as a mixture of two distributions, namely the confident distribution $\mathbf{P}_c$ and the unconfident distribution $\mathbf{P}_u$.

More formally, given a group of ongoing responses, at each step, we model the distribution of stepwise confidence scores $\mathbf{P}_\Phi$ as a mixture of two underlying distributions:

$$\mathbf{P}_\Phi = \pi_c \mathbf{P}_c + \pi_u \mathbf{P}_u \tag{8}$$

where π_c and π_u are the corresponding mixture weights satisfying $\pi_c + \pi_u = 1$. We also assume that $\mathbf{P}_c$ and $\mathbf{P}_u$ follow Gaussian distribution, parametrized by μ_c, σ_c and μ_u, σ_u, respectively. The formulation separates the reasoning steps into two categories: those that the model can handle confidently, and those that require routing to the larger model to solve it correctly. The parameters $\pi_u, \pi_c, \mu_c, \sigma_c, \mu_u$, and σ_u can be estimated by applying the EM algorithm to a group of step-level confidence scores at each step.

Using the fitted $\mathbf{P}_u$ and $\mathbf{P}_c$, we can calculate the posterior probability that an observed step-level confidence score Φ originates from the confident distribution $\mathbf{P}_c$ over the unconfident distribution $\mathbf{P}_u$. In this work, following the observations on the confidence distribution exemplified by Figure

2, we assume that $\mu_u < \mu_c$. We can then obtain a closed-form expression for the posterior probability as follows:

$$\Pr(c|\Phi) = \frac{\pi_c \mathbf{P}_c(\Phi;\mu_c,\sigma_c)}{\pi_c \mathbf{P}_c(\Phi;\mu_c,\sigma_c) + \pi_u \mathbf{P}_u(\Phi;\mu_u,\sigma_u)} \tag{9}$$

3.3 The STEER Algorithm

After estimating the underlying distributions $\mathbf{P}_c$ and $\mathbf{P}_u$, STEER routes the ongoing reasoning traces based on the posterior probability of each confidence score. To be precise, we adopt a posterior threshold γ as a hyperparameter for making routing decisions. Given a confidence score of step i, Φ_i, the model parameter θ_{i+1} to be used for generating the next step is determined as follows:

$$\theta_{i+1} = \begin{cases} \theta_m & \text{if} \quad \Pr(c|\Phi_i) \geq \gamma \\ \theta_M & \text{otherwise} \end{cases} \tag{10}$$

A reasoning trace is routed to a small model if the current step is expected to originate from confident generations. Otherwise, it routes to the larger model. The procedure distinguishes between the initial step and subsequent steps, as no prior confidence score is available for the former.

1. **Initial Generation**: At the first step ($i = 0$), the small model θ_m generates an output for all questions.
2. **First-Step Refinement**: Any initial output flagged as unconfident is re-generated with the larger model θ_M.
3. **Iterative Routing**: For steps $i > 1$, each reasoning trace is dynamically routed following the thresholding rule in Equation(10).
4. **Generation & Update Confidence**: The selected model generates the current step. Update the step-level confidence scores Φ_i for the current generation.
5. **Iteration**: Repeat steps 3-4 until all questions are solved or reach the maximum iteration limit.

More detailed and formal description of STEER and routing mechanism can be found in Algorithm 1 and in Appendix B.

4 Experiments

4.1 Experimental Setup

Benchmarks. To evaluate our method on the mathematical reasoning task, we employ (1) MATH500 (Hendrycks et al. 2021; Lightman et al. 2023), (2) Omnimath (Gao et al. 2025), and (3) AIME questions collected from 2022 through 2025. MATH500 and OmniMath provide human-annotated difficulty labels, enabling deeper analysis.

Additionally, to evaluate our approach on broader reasoning tasks, we utilize (1) MuSiQue (Trivedi et al. 2022), a multi-hop QA benchmark, and (2) the MCQ split of ACP-Bench (Kokel et al. 2025), a benchmark to evaluate LLMs' ability to reason about action and planning. We also use (3) KOR-Bench (Ma et al. 2025b), and evaluate on *Cipher*, *Counterfactual*, and *Logic* reasoning subsets[1]. More details on benchmarks and evaluation are in Appendix A.

Algorithm 1: STEER

Input: smaller model m and larger model M, their parameters θ_m and θ_M, questions $\mathcal{X}$, max steps N, threshold γ
Output: final solutions S

```
Generate step and its confidence $s_{0,m}, \Phi_{0,m} \sim p_{\theta_m}(\mathcal{X})$
Classify prompts $\mathcal{Q}_M$ to refine with larger model
        $(\mathcal{Q}_m, \mathcal{Q}_M) \leftarrow \texttt{classify}(\mathcal{X}, \Phi_{0,m}, \gamma)$
for each question $x \in \mathcal{X}$ do
    if $x \in \mathcal{Q}_M$ then
        Refine with larger model $s_{0,M}, \Phi_{0,M} \sim p_{\theta_M}(\mathcal{Q}_M)$
        Add generated steps to prompt $\mathcal{Q}_M \leftarrow \mathcal{Q}_M \oplus s_{0,M}$
    else
        Use the step by smaller model $\mathcal{Q}_m \leftarrow \mathcal{Q}_m \oplus s_{0,m}$
    end if
end for
for $i = 1$ to $N-1$ do
    Route prompts based on confidence
        $(\mathcal{Q}_m, \mathcal{Q}_M) \leftarrow \texttt{route}(\mathcal{Q}_m, \mathcal{Q}_M, \Phi_{i,m}, \Phi_{i,M}, \gamma)$
    Generate step and step confidence
        $s_{i,m}, \Phi_{i,m} \sim p_{\theta_m}(\mathcal{Q}_m)$; $s_{i,M}, \Phi_{i,M} \sim p_{\theta_M}(\mathcal{Q}_M)$
    $\mathcal{Q}_m \leftarrow \mathcal{Q}_m \oplus s_{i,m}$;   $\mathcal{Q}_M \leftarrow \mathcal{Q}_M \oplus s_{i,M}$
    if $\mathcal{Q}_m$ and $\mathcal{Q}_M$ is all complete then
        break
    end if
end for
$S \leftarrow [\,\mathcal{Q}_m;\ \mathcal{Q}_M\,]$
return $S$
```

Unless otherwise stated, STEER uses all questions in a benchmark to model $\mathbf{P}_{\Phi_i}$. Specifically, STEER models the distribution from all ongoing reasoning traces across all questions, excluding traces that have already terminated.

Models. We perform experiments on general-purpose and math-tuned models. Specifically, we evaluate the general-purpose Gemma3-Instruct (Team et al. 2025) and Qwen2.5-Instruct (Team 2024) models, and the math-tuned Qwen2.5-Math-Instruct model (Yang et al. 2024), which was tuned for mathematical reasoning with GRPO (Shao et al. 2024). For mathematical reasoning benchmarks, we exclude the general-purpose Qwen2.5-Instruct models, as their performance on the AIME benchmark is lacking for meaningful evaluation ($< 1\%$ accuracy). For broader reasoning benchmarks, we exclude Qwen2.5-Math-Instruct models, as we observed text degradation on non-math tasks. All experiments were performed using vLLM (Kwon et al. 2023) with `temperature=0.7`. We define a reasoning step as a text segment separated by `\n\n`.

Baselines. We adopt two groups of baseline frameworks: *External models*, and *No external models*.

For the ***External models*** baselines, we adopt (1) RSD (Liao et al. 2025) and the question-level model allocation framework by Damani et al. (2025). In RSD, each reasoning step is first generated using θ_m, after which a PRM assigns a scalar reward to the step. θ_M is called to regenerate the step only if the reward is below a pre-defined threshold. We use Skywork-o1-PRM 7B(o1 Team 2024), which

---

[1] We exclude *Operation* due to its similarity to math reasoning, and *Puzzles* due to low performance ($< 8\%$) across all models.

Method	Requires Trained Ext. Models	Allocation Granularity	MATH500			OmniMath			AIME			Average		
			Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑
Gemma3-Instruct														
4B Only	–	–	73.4	8.12	9.05	24.5	12.3	1.98	10.8	11.9	0.907	36.2	10.8	3.35
12B Only	–	–	84.4	19.4	4.35	33.0	35.4	0.93	17.5	33.8	0.517	44.9	29.4	1.53
SpecReason	✗	Step	79.8	12.3	6.45	29.6	27.0	1.10	15.0	33.6	0.447	41.4	24.2	1.71
Damani et al. (2025)	✓	Query	85.8	17.3	4.96	32.5	29.3	1.11	15.8	34.8	0.45	44.7	27.1	1.65
RSD	✓	Step	82.8	15.1	5.50	32.9	38.6	0.85	15.8	34.4	0.460	43.8	29.2	1.50
STEER	✗	Step	85.8	15.0	5.70	33.2	26.6	1.25	15.8	30.6	0.515	**44.9**	**24.0**	**1.87**
Qwen2.5-Math-Instruct														
1.5B Only	–	–	73.0	1.82	40.1	26.4	2.72	9.7	6.67	2.94	2.28	35.3	2.50	14.1
7B Only	–	–	79.6	9.10	8.75	28.2	13.56	2.08	8.33	16.6	0.500	38.7	13.10	2.96
SpecReason	✗	Step	77.4	7.74	10.0	28.0	12.8	2.18	10.0	15.5	0.645	38.5	12.0	3.21
Damani et al. (2025)	✓	Query	74.8	3.08	24.3	26.8	3.33	8.05	8.33	3.10	2.69	36.3	**3.17**	**11.5**
RSD	✓	Step	79.8	4.56	17.5	30.0	8.54	3.51	11.7	16.9	0.692	**40.5**	10.0	4.05
STEER	✗	Step	79.6	6.38	12.4	28.3	9.16	3.09	10.0	8.64	1.16	39.3	8.06	4.88

Table 1: Results on mathematial reasoning benchmarks. *Acc* denotes accuracy. *Requires Trained Ext. Models* denote if the method involves employing an external trained model. Best measures are in **bold**, and the second best underlined.

is the best-performing PRM in the RSD setting. Note that Skywork-o1-PRM models are trained on Qwen2.5-Math-Instruct models. To the best of our knowledge, there are no PRMs trained on Gemma3-Instruct. (2) Damani et al. (2025) train a lightweight difficulty estimation model to allocate models to input questions efficiently. The router takes in the question as input and allocates either θ_m or θ_M to generate the entire reasoning trajectory for the query using a thresholding rule. For the ***No-External Models*** baselines, we adopt SpecReason (Pan et al. 2025). SpecReason works similarly to RSD, but with the reward assignment done by prompting the larger LLM, using LLM-Judge. For all baseline methods and STEER, we perform a grid search with a gap of 0.1 over threshold values, selecting the best-performaning value for a fair comparison.

Efficiency Metrics Following previous works (Liao et al. 2025; Sardana et al. 2024; Kaplan et al. 2020), as a measure of inference cost, the standard FLOPs estimation for transformers with N parameters - $2N$ per inference token - is used, and we report the average FLOPs spent per query. We report FLOPs in units of 10^{12} for readability. We also report Accuracy-per-FLOPs (denoted as A/F for brevity) adapted from Ma et al. (2025c) to better convey the performance–inference cost balance.

4.2 Results

The results for mathematical reasoning and broader reasoning benchmarks are presented in Tables 1 and 2, respectively.

STEER exhibits strong performance across task domains and models. In the mathematical reasoning task, STEER effectively preserves or improves upon the reasoning ability of the larger model, while reducing up to 48% of the inference cost. Our method remains effective in broader reasoning domains, achieving the best accuracy-cost trade-off across benchmarks. When applied to Gemma3-Instruct models, STEER achieves the highest accuracy with the least FLOPs. With Qwen2.5-Instruct models, our method attains competitive accuracy (within 1% of the highest-scoring baseline) while using the least (18% less) computation, further validating the effectiveness of our framework.

External models have limited robustness With Qwen2.5-Math-Instruct models, RSD attains the highest accuracy in the mathematical reasoning domain. However, with Gemma3-Instruct models, RSD yields lower accuracy on mathematical reasoning benchmarks compared to the larger model, with minimal reduced inference costs. As Skywork-o1-PRM is trained on Qwen2.5-Math-Instruct, the reasoning traces from Gemma3-Instruct present an out-of-distribution problem to the PRM (Zhu et al. 2025), thus limiting its ability. Moreover, RSD exhibits sub-optimal performance on broader reasoning tasks, highlighting the limitations of PRMs (Zeng et al. 2025) when applied to out-of-domain tasks. We also note that in some cases, RSD incurs more FLOPs than solely querying the larger model. The root cause is the low acceptance rate for the smaller model's draft steps and the post-hoc evaluation nature of the RSD framework. As the step reward is evaluated post-hoc, rejecting a step leads to invoking *both* the smaller and the larger model. If the acceptance rate is significantly low, both models are invoked for most steps, resulting in FLOPs that match or exceed the sum of FLOPs by the smaller and larger models.

The query-level router Damani et al. (2025) reduces inference costs on Qwen2.5-Math-Instruct, albeit with visible performance degradation. The performance is sub-optimal with Gemma3-Instruct models across domains, and when run on Qwen2.5-Instruct for broader reasoning, it routes most queries to the larger model, leading to good accuracy at the cost of heavier inference costs compared to STEER.

Method	Requires Trained Ext. Models	Allocation Granularity	ACPBench			MuSiQue			KOR-Bench			Average		
			Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑	Acc↑	FLOPs↓	A/F↑
Gemma3-Instruct														
4B Only	–	–	53.5	5.48	9.75	43.2	0.80	54.0	46.8	2.98	15.7	47.8	3.08	15.5
12B Only	–	–	70.7	13.5	5.20	62.6	3.80	16.4	57.0	8.96	6.35	63.4	8.78	7.20
SpecReason	✗	Step	62.3	12.5	4.96	55.2	2.98	18.5	52.2	8.80	5.95	56.6	8.10	7.00
Damani et al. (2024)	✓	Query	67.5	11.8	5.70	47.6	1.90	25.0	52.5	6.64	7.90	55.9	6.80	8.20
RSD	✓	Step	58.7	9.68	6.05	55.8	2.54	22.0	52.6	8.78	6.00	55.7	7.00	7.95
STEER	✗	Step	64.5	9.10	7.10	54.8	2.38	23.0	54.6	6.72	8.15	**58.0**	**6.06**	**9.55**
Qwen2.5-Instruct														
1.5B Only	–	–	33.1	1.20	27.6	20.0	0.76	26.3	30.5	1.06	28.7	27.9	1.00	27.9
7B Only	–	–	58.4	5.92	9.85	50.6	3.88	13.0	53.2	5.80	10.5	54.0	5.20	10.9
SpecReason	✗	Step	50.3	4.36	11.5	37.8	3.22	11.7	41.4	4.62	8.95	43.2	4.06	10.6
Damani et al. (2024)	✓	Query	57.8	5.98	9.65	37.4	2.52	14.8	43.7	4.64	9.40	**46.3**	4.38	10.5
RSD	✓	Step	55.3	5.08	10.9	38.4	2.90	13.2	42.6	5.20	8.20	45.4	4.40	10.3
STEER	✗	Step	52.6	4.16	12.6	38.6	2.72	14.2	45.2	4.02	11.2	45.5	**3.64**	**12.5**

Table 2: Results on broader reasoning benchmarks. *Acc* denotes accuracy. *Requires Trained Ext. Models* denote if the method involves employing an external trained model. Best measures are in **bold**, and the second best underlined.

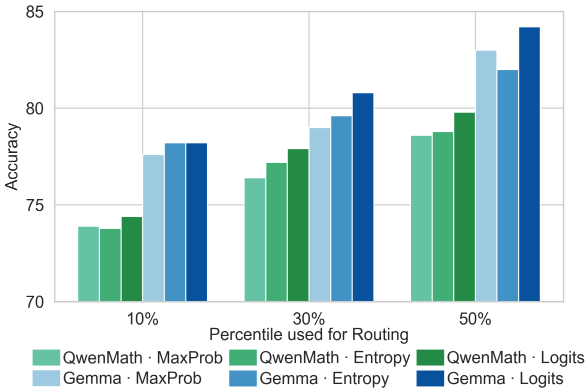

Figure 3: MATH500 accuracy for percentile routing with other measures of token confidence. *QwenMath* denotes Qwen2.5-Math-Instruct. *Gemma* denotes Gemma3-Instruct.

Comparison with LLM-Judge STEER outperforms SpecReason across most benchmarks, achieving higher average accuracy while requiring less computation costs. This result demonstrates that prompting is sub-optimal compared to logit-based confidence in providing a performance-inference cost trade-off.

4.3 Analysis

Confidence Measure Selection. We validate our approach against other confidence metrics, such as entropy and maximum probability. Our aim is to evaluate how effectively each confidence signal identifies wrong reasoning traces. Here we adopt a percentile-based routing strategy. At each reasoning step, we compute the confidence using each metric, and route the traces falling below the p-th percentile

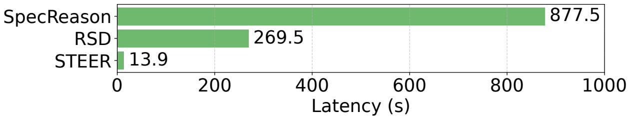

Figure 4: Overall latency induced by different methods, measured on the MATH500 benchmark.

to the larger model for subsequent step generation. As shown in Figure 3, maximum logit emerges as the most effective signal, yielding the highest accuracy overall. Notably, the superiority of logit-based confidence becomes more apparent at higher values of p, indicating its strength in estimating uncertainty in borderline scenarios.

GMM Routing Latency Advantage. As shown in Figure 4, the routing latency of GMM employed by STEER is over a magnitude lower than that of the baseline approaches. This can be attributed to the fact that RSD and SpecReason require a full forward pass through an LLM to obtain the routing signals, while STEER utilizes a lightweight GMM.

Robustness to Group Size Variations. We conduct an ablation study where we vary the number of samples used to fit the GMM. In this experiment, a benchmark with D questions is decomposed into K equally sized groups, each with size D/K. We solve each group independently using STEER, each using D/K questions to model $\mathbf{P}_{\Phi_i}$. We report the accuracy and FLOPs aggregated across the groups. Table 3 shows the variation in performance with respect to changes in the group size. Results show that STEER remains effective even when applied to a small-sized group of questions, demonstrating its robustness under limited-sample conditions for confidence distribution estimation.

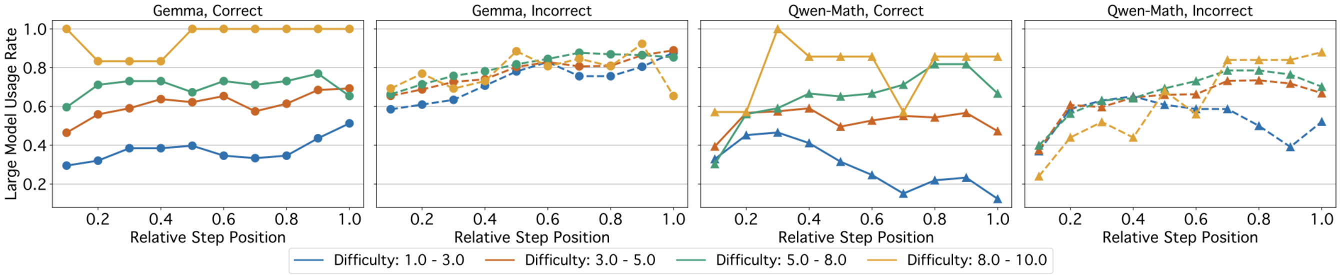

Figure 5: Ratio of steps using the larger model, by relative step position and difficulty in OmniMath. Relative step 0.1 denotes the first step, and 1.0 denotes the final step. Difficulty of 1.0 denotes the easiest, and 10.0 the hardest questions respectively.

		Number of Groups K			
		10	5	2	1
MATH500	Acc	83.9	83.5	83.5	85.8
	FLOPs	7.42	7.33	7.25	7.51
	A/F	11.3	11.4	11.5	11.4
ACPBench	Acc	63.6	64.3	64.5	64.5
	FLOPs	5.04	4.80	4.53	4.55
	A/F	12.6	13.4	14.2	14.2

Table 3: Results on MATH500 and ACPBench using STEER when divided in to K equally sized groups.

Larger Model Usage Pattern. We also analyze larger model usage rates by step position and difficulty on the OmniMath benchmark, which has human-annotated difficulty labels ranging from 1 to 10. Figure 5 shows that for both model families, the larger model usage increases with question difficulty in *correct* solutions, implying that STEER correctly leverages more computation when needed. Moreover, large model usage rates are higher in *incorrect* solutions, suggesting that STEER recognizes inability to solve problems correctly and routes more steps to the large model in an effort to solve them. We also note that the usage pattern differs between model families, particularly for correct solutions. Gemma3-Instruct models exhibit relatively consistent usage rates across reasoning steps, while Qwen2.5-Math-Instruct models show divergent usage patterns in later reasoning steps, depending on question difficulty. Still, STEER effectively leverages confidence signals for both model families, validating its robustness across backbone models.

5 Related Works

5.1 Cost-Efficient Adaptive Inference using Multiple Models

To reduce inference costs of LLMs, a body of work involves training an external model that allocates models of different sizes at test time. Works on Model Cascading (Dohan et al. 2022; Narasimhan et al. 2025) and Model Routing (Damani et al. 2025; Ong et al. 2024) utilize trained models that allocate simple queries to the smaller model and complex queries to the larger model. In the domain of cost-efficient reasoning (Sui et al. 2025), Liao et al. (2025) propose RSD, which leverages step-level reward values from a trained PRM to perform step-level Model Cascading. However, trained on task-specific and model-specific data, the external models used in these frameworks are not robust to changes in task domains and base models (Zeng et al. 2025; Zhu et al. 2025; Ong et al. 2024; Ding et al. 2024).

While STEER focuses on reducing inference costs, other works focus on reducing latency. Speculative decoding (Leviathan, Kalman, and Matias 2023; Li et al. 2025) involves first drafting a set of inference tokens using the smaller model, and then verifying them in parallel by assessing the likelihoods of the draft tokens under the larger model. Although such pipelines can reduce end-to-end latency, they offer no savings in inference compute (FLOPs), as every draft token still undergoes a full forward pass through the larger model during the verification process.

5.2 Measuring Confidence in Language Models

Prior works introduce probability-based approaches to measure confidence in generations of a language model, suggesting measures such as maximum probability (Arora, Huang, and He 2021; Kim, Yang, and Jung 2024), entropy (Kadavath et al. 2022), and semantic entropy (Farquhar et al. 2024). Recently, Ma et al. (2025a) argues that probability-based measures fail to capture model confidence when equally likely responses exist, and proposes modeling logit-based uncertainty with the Dirichlet distribution for uncertainty quantification. Other works focus on assessing response-level confidence. Duan et al. (2024) argues that not all tokens contribute equally to the semantics of the generated outputs. Similarly, Lin et al. (2025) find that certain tokens play more critical roles than others in math reasoning.

6 Conclusion

We propose STEER, a novel domain-agnostic, external model-free framework that performs step-level routing for cost-efficient reasoning. STEER leverages logit-based confidence estimation together with GMMs for step-level routing, striking a balance between performance and inference cost. Extensive evaluation across reasoning domains and models validate the effectiveness of STEER. By eliminating the need for trained external modules, STEER achieves superior robustness over existing baselines, presenting a practical solution to the challenge of efficient LLM deployment.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) [No.RS-2023-00229780, Development of Artificial Intelligence Technology for Process-focused Evaluation(Student's Learning Diagnosis)]. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No.RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics & No.RS-2021-II212068, Artificial Intelligence Innovation Hub (Artificial Intelligence Institute, Seoul National University) & No.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)]. In addition, we thank LYWAY for providing math test sets and for the experimental support via API, which contributed to the development of this work. K. Jung is with ASRI, Seoul National University, Korea.

References

Arora, U.; Huang, W.; and He, H. 2021. Types of Out-of-Distribution Texts and How to Detect Them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10687–10701. Association for Computational Linguistics.

Chen, L.; Zaharia, M.; and Zou, J. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research*.

Damani, M.; Shenfeld, I.; Peng, A.; Bobu, A.; and Andreas, J. 2025. Learning How Hard to Think: Input-Adaptive Allocation of LM Computation. In *The Thirteenth International Conference on Learning Representations*.

Ding, D.; Mallick, A.; Wang, C.; Sim, R.; Mukherjee, S.; Rühle, V.; Lakshmanan, L. V.; and Awadallah, A. H. 2024. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. In *The Twelfth International Conference on Learning Representations*.

Dohan, D.; Xu, W.; Lewkowycz, A.; Austin, J.; Bieber, D.; Lopes, R. G.; Wu, Y.; Michalewski, H.; Saurous, R. A.; Sohl-Dickstein, J.; et al. 2022. Language model cascades. *International Conference on Machine Learning*.

Duan, J.; Cheng, H.; Wang, S.; Zavalny, A.; Wang, C.; Xu, R.; Kailkhura, B.; and Xu, K. 2024. Shifting Attention to Relevance: Towards the Predictive Uncertainty Quantification of Free-Form Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5050–5063.

Faiz, A.; Kaneda, S.; Wang, R.; Osi, R. C.; Sharma, P.; Chen, F.; and Jiang, L. 2024. LLMCarbon: Modeling the End-to-End Carbon Footprint of Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Farquhar, S.; Kossen, J.; Kuhn, L.; and Gal, Y. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017): 625–630.

Gao, B.; Song, F.; Yang, Z.; Cai, Z.; Miao, Y.; Dong, Q.; Li, L.; Ma, C.; Chen, L.; Xu, R.; et al. 2025. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark for Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Gupta, N.; Narasimhan, H.; Jitkrittum, W.; Rawat, A. S.; Menon, A. K.; and Kumar, S. 2024. Language Model Cascades: Token-Level Uncertainty And Beyond. In *The Twelfth International Conference on Learning Representations*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; et al. 2022. Language Models (Mostly) Know What They Know. *CoRR*.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kim, J.; Yang, N.; and Jung, K. 2024. Persona is a Double-edged Sword: Mitigating the Negative Impact of Role-playing Prompts in Zero-shot Reasoning Tasks. arXiv:2408.08631.

Kokel, H.; Katz, M.; Srinivas, K.; and Sohrabi, S. 2025. Acpbench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26559–26568.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Leviathan, Y.; Kalman, M.; and Matias, Y. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 19274–19286. PMLR.

Li, Y.; Wei, F.; Zhang, C.; and Zhang, H. 2025. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*.

Liao, B.; Xu, Y.; Dong, H.; Li, J.; Monz, C.; Savarese, S.; Sahoo, D.; and Xiong, C. 2025. Reward-Guided Speculative Decoding for Efficient LLM Reasoning. In *Forty-second International Conference on Machine Learning*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Lin, Z.; Liang, T.; Xu, J.; Liu, Q.; Wang, X.; Luo, R.; Shi, C.; Li, S.; Yang, Y.; and Tu, Z. 2025. Critical Tokens Matter: Token-Level Contrastive Estimation Enhances LLM's Reasoning Capability. In *Forty-second International Conference on Machine Learning*.

Ma, H.; Chen, J.; Zhou, J. T.; Wang, G.; and Zhang, C. 2025a. Estimating LLM Uncertainty with Evidence. arXiv:2502.00290.

Ma, K.; Du, X.; Wang, Y.; Zhang, H.; ZhoufutuWen; Qu, X.; Yang, J.; Liu, J.; minghao liu; Yue, X.; Huang, W.; and Zhang, G. 2025b. KOR-Bench: Benchmarking Language Models on Knowledge-Orthogonal Reasoning Tasks. In *The Thirteenth International Conference on Learning Representations*.

Ma, X.; Wan, G.; Yu, R.; Fang, G.; and Wang, X. 2025c. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6025–6035. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Narasimhan, H.; Jitkrittum, W.; Rawat, A. S.; Kim, S.; Gupta, N.; Menon, A. K.; and Kumar, S. 2025. Faster Cascades via Speculative Decoding. In *The Thirteenth International Conference on Learning Representations*.

o1 Team. 2024. Skywork-o1 Open Series. https://huggingface.co/Skywork. Accessed: 2025-12-06.

Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.-L.; Wu, T.; Gonzalez, J. E.; Kadous, M. W.; and Stoica, I. 2024. RouteLLM: Learning to Route LLMs from Preference Data. In *The Thirteenth International Conference on Learning Representations*.

Pan, R.; Dai, Y.; Zhang, Z.; Oliaro, G.; Jia, Z.; and Netravali, R. 2025. Specreason: Fast and accurate inference-time compute via speculative reasoning. *arXiv preprint arXiv:2504.07891*.

Poddar, S.; Koley, P.; Misra, J.; Ganguly, N.; and Ghosh, S. 2025. Towards Sustainable NLP: Insights from Benchmarking Inference Energy in Large Language Models. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 12688–12704. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.

Sardana, N.; Portes, J.; Doubov, S.; and Frankle, J. 2024. Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws. In *ICML*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sui, Y.; Chuang, Y.-N.; Wang, G.; Zhang, J.; Zhang, T.; Yuan, J.; Liu, H.; Wen, A.; Zhong, S.; Chen, H.; et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Wang, X.; and Zhou, D. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37: 66383–66409.

Xie, Y.; Kawaguchi, K.; Zhao, Y.; Zhao, J. X.; Kan, M.-Y.; He, J.; and Xie, M. 2023. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36: 41618–41650.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; Lu, K.; Xue, M.; Lin, R.; Liu, T.; Ren, X.; and Zhang, Z. 2024. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. arXiv:2409.12122.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Zeng, T.; Zhang, S.; Wu, S.; Classen, C.; Chae, D.; Ewer, E.; Lee, M.; Kim, H.; Kang, W.; Kunde, J.; Fan, Y.; Kim, J.; Koo, H. I.; Ramchandran, K.; Papailiopoulos, D.; and Lee, K. 2025. VersaPRM: Multi-Domain Process Reward Model via Synthetic Reasoning Data. In *Forty-second International Conference on Machine Learning*.

Zhu, J.; Zheng, C.; Lin, J.; Du, K.; Wen, Y.; Yu, Y.; Wang, J.; and Zhang, W. 2025. Retrieval-Augmented Process Reward Model for Generalizable Mathematical Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 8453–8468. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.