Adaptive Inference Scaling via Monte Carlo Sampling

Joseph Boen
Stanford University
tboen@stanford.edu

James Zou
Stanford University
jamesz@stanford.edu

Abstract

LLM inference time scaling has emerged as an important paradigm for training-free alignment of LLMs using external reward signals. However, central questions regarding practical deployment, such as answer selection methods and optimal compute allocation, remain poorly understood, with advancements primarily driven by empirical heuristics. To address this, we provide a principled framework for analyzing inference time scaling via Monte Carlo (MC) sampling. This framework treats inference scaling as a statistical estimation problem over a reward weighted posterior, and introduces principled choices for response selection and compute allocation strategies. Experiments on mathematical reasoning benchmarks show that (i) our MC derived inference scaling methods outperform baseline strategies, (ii) our adaptive inference scaling strategy dynamically adjusts compute on per-query basis, allocating more compute to challenging prompts.

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance across a variety of tasks. Yet, LLMs still struggle on tasks requiring complex reasoning or nuanced user preferences. In these domains, performance is often improved by incorporating external reward signals such as human feedback, verifier scores, or trained reward models into the generation process. This procedure, broadly referred to as LLM "alignment", has often been accomplished through model post-training. While effective, finetuning approaches are expensive as they require retraining large models, and inflexible to changes in reward signals [Rafailov et al., 2023, Ouyang et al., 2022].

An emerging alternative is *inference-time scaling*, which improves quality by allocating more computation at generation without modifying model parameters. Applying inference time strategies to small models can substantially boost performance, even rivaling much larger models. However, several open questions remain for inference-time scaling: When is the amount of compute dedicated to a particular problem "enough", and how should one select the final response? How should we allocate limited compute to multiple queries without sacrificing performance? [Snell et al., 2024]

In this work, we introduce a Monte Carlo (MC) perspective on inference scaling for LLM alignment. We treat alignment as sampling from a reward weighted posterior distribution, using the base LLM's output distribution as the prior. Within this view, inference time scaling corresponds to improving an MC estimate of a target statistic under the posterior distribution. This observation motivates the introduction of estimator quality diagnostics, which can be used to terminate sampling when the estimator is sufficiently robust. Concretely, we make the following contributions:

- We derive a principled statistical framework for analyzing inference time scaling. We recast inference-time alignment as MC estimation under a reward-weighted posterior and identify the Minimum Bayes Risk (MBR) estimator as the natural target statistic, and introduce approximation diagnostics that allow for online, per-prompt compute allocation.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

- We empirically validate our methods using Importance Sampling for MBR estimation (IS-MBR) by (1) demonstrating that fixed IS-MBR yields superior compute–accuracy tradeoffs compared to BoN, and (2) when combined with our diagnostics, adaptive IS-MBR implicitly captures question difficulty by allocating more compute to challenging prompts.

2 A Monte Carlo View of LLM Inference-Time Alignment

2.1 Alignment as Posterior Inference

In a language model (LM), generation corresponds to sampling a response y from a distribution over sequences conditioned on a prompt x. In LLM alignment, responses are evaluated through a *reward model* $r(x, y)$, that assigns a reward reflecting the quality of the response, and the objective is to bias generation towards producing responses with higher rewards.

One common approach to LM alignment is to finetune the LM's parameters so as to maximize the expected reward over a training set of prompts $x \in \mathcal{D}$, while regularizing the deviation from its pretrained distribution. This leads to the KL-regularized RL objective,

$$\pi^* = \max_{\pi} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi(\cdot|x)}\big[r(x,y)\big] - \beta\, D_{\mathrm{KL}}\big(\pi(\cdot \mid x) \,\|\, p_{\mathrm{LM}}(\cdot \mid x)\big), \tag{1}$$

Here π^* denotes the model after alignment fine-tuning, and the optimization problem is typically solved via policy gradient methods like PPO [Ouyang et al., 2022].

Alternatively, we can view alignment through a Bayesian lens. We can regard $p_{\mathrm{LM}}(y \mid x)$ as a prior over responses, and the reward model as providing *likelihood* information about the event that y is "aligned" for prompt x. Formally, we define a binary variable $\phi(y, x)$ indicating "preference" or "verification", such that $p(\phi = 1 \mid y, x) \;\propto\; \exp\big(r(x, y)/\beta\big)$. Bayes' rule then gives the *aligned posterior*:

$$\pi(y \mid x) \;\triangleq\; p(y \mid \phi = 1, x) \;=\; \frac{1}{Z(x)}\, p_{\mathrm{LM}}(y \mid x)\, \exp\big(r(x,y)/\beta\big), \tag{2}$$

where $Z(x)$ is the normalizing constant. It can be shown that $\pi(y \mid x)$ as defined in Equation 2 is the optimal policy for the KL-regularized RL objective in Equation 1 [Korbak et al., 2022]. From this perspective, alignment reduces to a *posterior inference problem*: given a prior p_{LM} and evidence from r, our goal at inference time is to produce high-quality samples from the posterior π.

2.2 Test Time Scaling via Monte Carlo Sampling

Inference scaling is an alternate method for LM alignment that avoids model finetuning, which can be expensive and inflexible. In inference scaling, LMs improve response quality by increasing the amount of computation allocated to a given prompt. This is typically accomplished by *searching* against the RM by repeatedly generating responses, and selecting the one with the highest reward.

In our work, we reframe inference time scaling as a *sampling* problem for given function over a target distribution. In classical MC, the goal is to estimate some function or statistic f over a target distribution π by drawing samples and computing the expectation $\mathbb{E}_{y\sim\pi}[f(y)]$, such that the quality of this estimate improves with the number of samples. For LLM alignment, our target distribution is simply the aligned posterior $\pi(y \mid x)$ defined in Equation 2.

When choosing our statistic, we aim to produce a *single* high-quality response over π. Instead of selecting the maximum reward response, a principled alternative is the *Minimum Bayes Risk* (MBR) estimator [Kumar and Byrne, 2004], which selects the candidate y that maximizes expected utility (or "alignment score") under π:

$$y^* = \arg\max_{y\sim\pi(y|x)} \mathbb{E}_{y'\sim\pi(y|x)} \texttt{align}(y, y'), \tag{3}$$

here $\texttt{align}(\cdot,\cdot)$ is a task-specific similarity measure (e.g., mathematical equivalence). When both the candidates y and references y' are both drawn from π, MBR decoding is analogous to the "centroid" of the reward weighted posterior, thus balancing response reward and robustness.

From an MC perspective, `align` is simply a function, and MBR its expectation over π. With a small sample budget, this estimate will be noisy; with more samples, it will converge toward the true MBR

response. In this view, inference scaling is precisely the act of increasing our MC budget to improve this estimator. This connection reframes inference scaling as a *statistical estimation problem*, giving a principled way to reason about the trade-off between compute and alignment quality.

3 Methodology

3.1 From Best-of-N to Importance Sampling

One of the most popular inference scaling strategies is *Best-of-N* (BoN) sampling. Given a prompt x, BoN draws N independent responses from the base model p_{LM} and returns the highest-scoring candidate under a reward model: $y^*_{\text{BoN}} = \arg\max_{y_i \sim p_{LM}(y|x)} r(x, y_i)$. As N increases, the output distribution is increasingly biased toward high-reward responses, and can be remarkably effective with strong reward models. However, due to its greedy selection process, BoN is vulnerable to reward hacking, which can lead to diminishing or even negative returns for large N [Gao et al., 2023].

From our MC perspective, the natural analogue of BoN is *Importance Sampling* for MBR estimation (IS-MBR). IS-MBR draws N samples $\mathcal{Y} = \{y_i \sim q(y \mid x)\}_{i=1}^N$ from a proposal distribution q, and reweights them to match the target π before computing the MBR:

$$y^*_{\text{IS}-\text{MBR}} = \arg\max_{y \sim q(y|x)} \mathbb{E}_{y' \sim q(y|x)} \left[w'(y') \cdot \texttt{align}(y, y') \right] = \arg\max_{y \in \mathcal{Y}} \sum_{y_i \in Y} \left[w_i \cdot \texttt{align}(y, y_i) \right]$$

$$\tilde{w}_i = \frac{\pi(y_i \mid x)}{q(y_i \mid x)}, \quad w_i = \tilde{w}_i / \sum_{j=1}^{N} \tilde{w}_j \tag{4}$$

When $q = p_{LM}$, $\tilde{w}$ simplifies to the reward scores. In this special case, IS-MBR is procedurally identical to BoN: both perform parallel sampling from the LM, followed by RM scoring, and select a candidate. However, as demonstrated in the following sections, IS-MBR is far more robust than BoN.

3.2 Adaptive Inference Scaling via MC Diagnostics

While IS is guaranteed to converge to the optimal MBR solution as $N \to \infty$, in practice, sampling is costly, and users can only reasonably afford tens to hundreds of LLM inference calls. Thus, instead of consistently scaling compute across all queries, it is desirable to *adapt* the inference budget to the relative *difficulty* of the problem x. From a MC perspective, this corresponds to evaluating how close our finite-sample estimate is to the true statistic under $\pi(y \mid x)$, for each individual prompt x.

Formally, let f denote our function (e.g. MBR), such that $f(\pi)$ denotes our statistic over the true target distribution, and $f(\hat{\pi}_N)$ the estimator based on N samples. Our task is to decide when $f(\hat{\pi}_N)$ is a sufficiently accurate proxy for $f(\pi)$. Naively, one could recompute $f(\hat{\pi_N})$ as each new sample is added and monitor its consistency. In practice this strategy has two important drawbacks. Firstly, recomputing $f(\hat{\pi}_N)$ incurs significant overhead, especially if f is expensive. Secondly, *apparent* stability of $f(\hat{\pi}_N)$ is not a reliable measure of convergence. For MBR, a single late-arriving, high-weight sample can "flip" the selected candidate, even if $f(\hat{\pi}_N)$ was previously consistent.

This motivates diagnostics that monitor the quality of our sampled target distribution, rather than the estimator's value. Importance weights are a natural, low-cost basis for such diagnostics: the weights must be computed regardless, so inspecting them incurs negligible extra cost, and they summarize two properties that matter for downstream reliability: *concentration*, (are a few samples dominating the mass?) and *relevant-coverage* (is there non-trivial posterior mass in the high-reward region that the statistic cares about?) Concretely, we employ two complementary weight-only diagnostics.

Effective Sample Size (ESS): $N_{eff} = 1/\sum_i w_i^2$. A low ESS warns that a small number of samples dominate and that further sampling is needed to reduce estimator variance.

Relevant Support Mass (RSM): $\hat{P}_\pi(r \geq \tau) = \sum_i \tilde{w}_i \mathbb{I}(r(y_i) \geq \tau)$ where τ is an RM specific threshold for "high" rewards. RSM estimates how much of the sampled posterior mass lies in the high-reward region that the MBR objective depends on.

The combination of ESS and RSM allows us to capture the reliability of $f(\hat{\pi}_N)$. While ESS measures the *consistency* of our samples, RSS measures their *quality*. Concretely, when both RSM and ESS are sufficiently high, this indicates that IS-MBR is both successfully and consistently sampling high

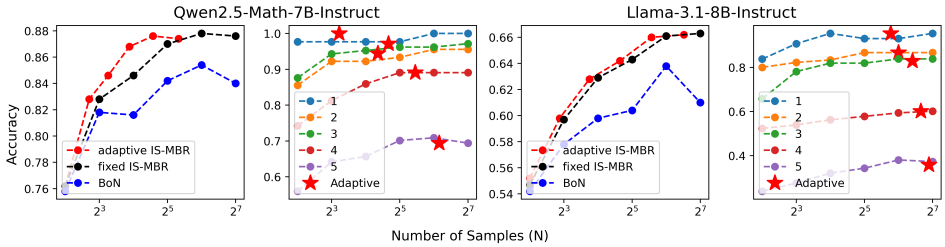

Figure 1: We illustrate the performance of adaptive and fixed IS-MBR vs BoN on the `MATH500` dataset. As shown, both adaptive and fixed IS-MBR outperforms BoN, while adaptive IS-MBR dynamically adjusts the sample size based on the problem difficulty. Here the stars indicate the sample sizes and performance of adaptive IS-MBR with a maximum budget of $N = 128$, with the dashed lines indicating baselines from the fixed IS-MBR.

reward regions of the response space, and the estimator is sufficiently robust enough to terminate sampling early. While these diagnostics are not sufficient conditions for true convergence, they provide indirect evidence about the robustness of $\hat{\pi}_N$ for the reliable computation of $f(\hat{\pi}_N)$. Both diagnostics integrate naturally with IS-MBR and are cheap to compute, thus enabling *online* decision making when allocating inference time scaling budgets for an individual prompt.

4 Results

We evaluate our method on the mathematical reasoning benchmark `MATH500` [Lightman et al., 2023], using Llama 3.1-8B-Instruct and Qwen2.5-Math-7B-Instruct as p_{LM}, and Qwen2.5-Math-PRM-7B as the reward model r [Dubey et al., 2024, Yang et al., 2024]. Although the Qwen PRM is trained for process supervision, in our experiments we assign a single reward to each complete response by performing step-wise reward aggregation using the `last` operator. We evaluate our method by performing IS-MBR with increasing compute budgets $N = \{4, 8, 16, 32, 64, 128\}$. In the non-adaptive or "fixed" setting, we perform standard IS-MBR over N samples. In the "adaptive" setting, we compute our ESS and RSM diagnostics after each sample, and terminate sampling when ESS and RSM are sufficiently high. We compare both to fixed sample size BoN. (Appendix B)

We observe that both fixed and adaptive IS-MBR strongly outperforms BoN for both the Llama and Qwen base models, where the decline in performance at large N for BoN is presumably due to reward over-optimization. For Qwen, we observe that adaptive IS-MBR enables signficant compute savings while sacrificing minimal accuracy. For Llama, these gains are still apparent, though far less pronounced. In addition, the questions in the `MATH500` dataset are annotated with difficulty levels ranging from 1 to 5, where 5 is the hardest. For both models, we observe that the sampling budget for our adpative IS-MBR model increases as the question difficulty increases (Figure 1).

5 Conclusion and Limitations

This paper introduces a framework for analyzing inference time scaling by recasting it as an MC estimation problem under a reward-weighted posterior, and empirically demonstrates the implications and improvements of this approach. Specifically, we motivate the selection of the MBR estimator as a target statistic, IS as the sampling procedure, and derive approximation diagnostics based on the importance weights for determining the reliability of the finite sample size estimator. We observe improved performance in both efficiency and accuracy on a mathematical reasoning benchmark.

While promising, we caution that our techniques require further evaluation across more LLMs, reward models, and domains to ensure its broad applicability. Given the discrepancies between the Qwen and Llama base model performance when using a Qwen reward model, we hypothesize that reward model calibration may have a significant influence on our convergence diagnostics. Furthermore, it would be interesting to explore more powerful MC sampling algorithms.

References

Edward Beeching, Lewis Tunstall, and Sasha Rush. Scaling test-time compute with open models. *URL https://huggingface. co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute*, 13, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Shengyu Feng, Xiang Kong, Shuang Ma, Aonan Zhang, Dong Yin, Chong Wang, Ruoming Pang, and Yiming Yang. Step-by-step reasoning for math problems via twisted sequential monte carlo. *arXiv preprint arXiv:2410.01920*, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. Rl with kl penalties is better viewed as bayesian inference. *arXiv preprint arXiv:2205.11275*, 2022.

Shankar Kumar and William Byrne. Minimum bayes-risk decoding for statistical machine translation. 2004.

Hynek Kydlíček. Math-verify: Math verification library. `https://github.com/huggingface/math-verify`, 2025. Apache-2.0 License.

James Lesatod, Jonathan Rivera, Lucas Kowalski, Matthew Robinson, and Nicholas Ferreira. An adaptive compute approach to optimize inference efficiency in large language models. 2024.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier. *arXiv preprint arXiv:2206.02336*, 2022.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-n decoding via speculative rejection. *Advances in Neural Information Processing Systems*, 37:32630–32652, 2024.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024.

A Additional Related Work

LLM Inference Scaling A variety of strategies have been employed for LLM inference scaling. Most techniques employ search based strategies, where responses are repeatedly generated and modified to maximize some reward signal. Representative algorithms include Best of N [Snell et al., 2024], Monte Carlo Tree Search [Guan et al., 2025], and Diverse Verifier Tree Search [Beeching et al., 2024]. Self-refinement strategies like Textgrad [Yuksekgonul et al., 2025] and React [Yao et al., 2023] can also be viewed as inference time strategies, albeit through repeated refinement over a single response "chain". In our work, we reframe inference time scaling as a sampling problem over the reward weighted LLM posterior. Our decision is motivated by the observation that aggressive search against imperfect reward models can lead to over-optimization [Gao et al., 2023], and that Bayesian inference naturally mitigates this by incorporating the LLM prior.

Adaptive Inference Scaling Optimizing the allocation of inference time compute has often been approached through the context of balancing model *size* with the amount of additional *tokens* generated at inference time. In this setting, the decision is whether to apply inference time scaling to a smaller model, or *switch* to a single-shot response from a larger model. Several works have explored this tradeoff and developed so-called "inference time scaling laws" that capture the relationship between model size, inference compute, and performance. While the exact tradeoff varies from strategy to strategy, most works reliably demonstrate that smaller models equipped with sufficient inference time compute can be more efficient and potentially even more performant, than larger models without inference compute [Wu et al., 2024, Snell et al., 2024]. Other works have explored adaptive inference scaling from a model architecture perspective, that apply "early exits" or "shortcuts" through the model layers depending on the complexity of the inputs [Lesatod et al., 2024]. These architectural strategies have even been successfully been adopted for standard inference time scaling methods, for example through speculative decoding for best of N [Sun et al., 2024], which automatically halts the generation of low reward sequences. In our setting, the adaptive sampling tradeoff is between the success probability and computational cost for a *fixed* model on a *single* prompt, or more generally, how to allocate a fixed computational budget for a given model on multiple prompts with varying difficulties. To the best of our knowledge, prior works have not proposed adaptive computation budgets in this context.

Monte Carlo Algorithms for LLMs Monte Carlo (MC) methods have been applied to LLMs in several contexts. Twisted Sequential Monte Carlo (TSMC) has been applied to both to token-level sequence generation [Zhao et al., 2024] and step-level reasoning [Feng et al., 2024]. In these works however, the focus is on *learning* optimal twist functions that can effectively guide sampling beyond outcome reward models. While promising, our work focuses on applying MC to LLMs with fixed outcome reward models, in the context of test time alignment. Regarding the choice of MBR, we note that IS-MBR with `align(.,.)` defined as an exact final-answer match recovers the "weighted" best-of-N [Li et al., 2022] heuristic, where the answer is chosen by a reward-weighted majority vote. While weighted BoN was originally proposed as a practical heuristic, our formulation shows that it is in fact a special case of a more general principle: answer selection as computing the expectation of a test statistic under a reward-weighted posterior. This perspective both explains the empirical success of weighted BoN and offers a natural path to our analysis of finite sample approximations and per-prompt adaptive inference scaling.

B Experiment Details

B.1 Prompts

We use the following system prompt for all LLMs on the `MATH500` dataset.

System Prompt

```
Solve the following math problem efficiently and clearly:

- For simple problems (2 steps or fewer):
Provide a concise solution with minimal explanation.

- For complex problems (3 steps or more):

Use this step-by-step format:
## Step 1: (Concise description)
[Brief explanation and calculations]

## Step 2: (Concise description)
[Brief explanation and calculations]

Regardless of the approach, always conclude with:

Therefore, the final answer is: $\boxed{answer}$. I hope it is correct.

Where [answer] is just the final number or expression that solves the problem.
```

B.2 Parsing, Scoring, and MBR Calculations

For the mathematical benchmarks explored in our experiments, we compute the MBR solution with $\texttt{align}(y_i, y_j)$ defined as a binary function denoting the mathematical equivalence between two solutions, implemented using the HuggingFace library `math-verify` [Kydlíček, 2025]. Briefly, `math-verify` parses the LLM responses text to extract the boxed expression, before converting into SymPy expressions to check for algebraic or numeric equivalency. We use the same functionality to compute the accuracy between our MBR solution and the ground truth solution.

B.3 Model Information

We use Llama 3.1-8B-Instruct and Qwen2.5-Math-7B-Instruct as our base LLMs, and Qwen2.5-Math-PRM-7B as our reward model. For our base LLMs, we perform inference using `vllm` with tensor parallelism on 2 Nvidia L40S GPUs, each with 48GB VRAM. We sample responses with $\texttt{temperature} = 1.0$ and $\texttt{max\_tokens} = 2048$.

For our reward model, we deploy Qwen2.5-Math-PRM-7B on 1 Nvidia L40S GPU using the HuggingFace `transformers` library. Although the Qwen PRM is trained for process supervision, following Snell et al. [2024] we assign a single reward to each complete response by performing step-wise reward aggregation using the `last` operator. Concretely, if $y_{1:t} = [y_0, y_1, ...y_t]$ denotes a response broken into t reasoning "steps"

$$r(x, y) = \texttt{last}(\text{PRM}(y_{1:t}, x)) = \texttt{last}([r_1, r_2, \cdots r_t]) = r_t$$

B.4 Adaptive Inference Scaling Algorithm Details

We set the reward temperature $\beta = 1.0$, and the minimum number of samples $N_{min} = 4$. The estimator robustness hyperparameters for ESS (ESM_{upper}) and RSM (τ, RSM_{upper}) are model specific, and are related to the calibration of the Reward Model r to the distribution of responses from p_{LM}. Intuitively, the choice of diagnostic parameter thresholds for ESS is related the estimator variance, while the RSM thresholds are related to the distribution of rewards assigned to correct vs incorrect responses.

Algorithm 1 Adaptive IS-MBR

Initialize `ESS_check = False`, `RSS_check = False`, `continue_sampling = True`

while `continue_sampling` **do**
  `# Perform Importance Sampling`
  Sample a response $y_i \sim p_{LM}(y \mid x)$
  Compute importance weights $\tilde{w}_i = \exp(r(y_i, x)/\beta), \;\; w_i = \tilde{w}_i / \sum_j \tilde{w}_j$
  Construct empirical distribution $\hat{\pi}_N = \sum_{i=1}^{N} w_i \delta(y_i)$

  `# Computing diagnostics`
  Compute ESS: N_{eff}, and ESS ratio: N_{eff}/N
  Compute the $(1-\alpha)$ CI boostrap for RSM: $P_\pi(r \geq \tau) \in [\hat{P}_\pi(r \geq \tau)_{\alpha/2}, \hat{P}_\pi(r \geq \tau)_{1-\alpha/2}]$

  `# Check estimator robustness`
  if $N_{eff} \geq N_{min}$ AND $N_{eff}/N \geq \text{ESS}_{upper}$ **then**
    `ESS_check = True`
  end if
  if $\hat{P}_\pi(r \geq \tau)_{\alpha/2} > \text{RSM}_{upper}$ **then**
    `RSM_check = True`
  end if

  `# Make decision`
  if (`RSM_check` AND `ESS_check`) OR $N > N_{max}$ **then**
    `continue_sampling = False`
  end if
end while
return MBR over $\hat{\pi}_N$ using Equation 4

In practice, we have found it helpful to tune these hyperparameters over a small subset of sample questions ($n = 10$ questions) before deployment. In the results shown in Figure 1, for Qwen2.5-Math-7B-Instruct, we set $\text{ESM}_{upper} = 0.75, \tau = 0.5, \text{RSM}_{upper} = 0.5$, and for Llama-3.1-8B-Instruct we set $\text{ESM}_{upper} = 0.75, \tau = 0.75, \text{RSM}_{upper} = 0.75$. When estimating the RSM value, we compute a 95% confidence interval (i.e. $\alpha = 0.05$) over 1000 repetitions.

Note that our decision strategy in Algorithm 1 is *optimistic*, if the upper thresholds of our diagnostic criteria ESS_{upper} and RSM_{upper} are not satisfied, we continue to sample until we have reached N_{max}. Alternatively, we could introduce *lower* thresholds for early termination. For example, if our most optimistic estimate for our RSM, $\hat{P}_\pi(r \geq \tau)_{1-\alpha/2}$ was *less* than our minimum acceptable RSM after a certain number of iterations, this would be strong evidence that $q = p_{LM}$ is incapable of successfully sampling from π, and should thus be terminated.

C Additional Results

C.1 ESS and MBR Variance

As before, let $f(\hat{\pi}_N)$ denote our finite-sample estimate of statistic f over π based on N samples. In order to show that our weight-only diagnostic ESS is a relevant proxy for the variance of $f(\hat{\pi}_N)$, we recompute the MBR over multiple sets of responses sampled from p_{LM} and plot the variance of the final response as a function of N_{eff}.

Concretely, for a fixed number of repetitions K, we perform importance sampling with sample budget N and compute the MBR as outlined in equation 4. We keep track of both the ESS and the final response, and define the variance as the proportion of unique responses over the total number of repetitions,

$$\hat{\text{Var}}(y_{\text{IS-MBR}}) = \frac{1}{K} |\{y_{\text{IS-MBR}}\}_{i=1}^{K}|$$

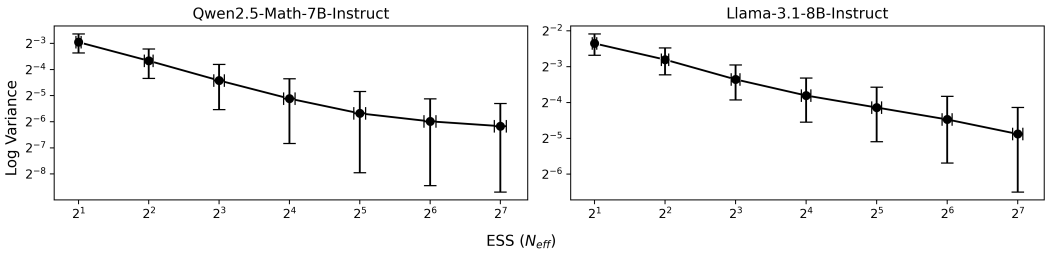

Figure 2: Test statistic variance (MBR) as a function of N_{eff}. Wihle the variance is higher for Llama-3.1-8B-Instruct than for Qwen2.5-Math-7B-Instruct, we observe that variance decays in a power law relationship with N_{eff}.

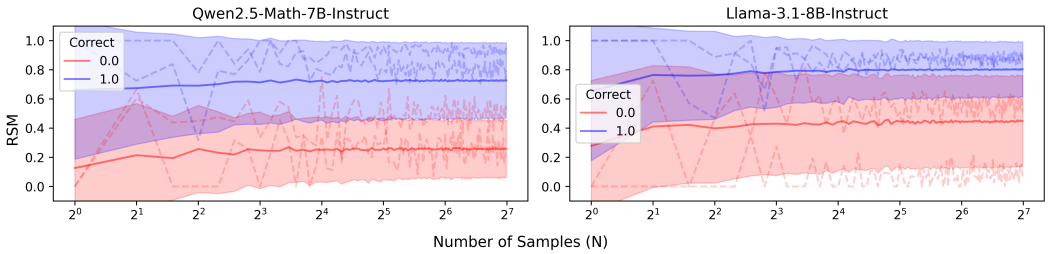

Figure 3: RSM $\hat{P}_\pi(r \geq \tau)$ convergence as a function of sample size N. The solid blue/red lines denote the mean trajectory of the empirical RSM over correct and incorrect responses respectively. The shaded regions correspond 1 standard deviation, and the dashed lines show individual IS-MBR trajectories from select questions. Note that the separation between correct/incorrect responses for Llama-3.1-8B-Instruct is less pronounced than for Qwen2.5-Math-7B-Instruct, indicating that the Qwen2.5-Math-PRM-7B reward model is less well calibrated for Llama.

Intuitively, if the IS-MBR solution is different for each repetition, the variance will be high. Conversely, if each reptition generates the same or similar solutions, the variance approaches $1/K \to 0$ with increasing K.

We perform this experiment for $K = 100$ repetitions over the `MATH500` dataset for both Qwen2.5-Math-7B-Instruct and Llama-3.1-8B-Instruct, again using Qwen2.5-Math-PRM-7B as our reward model following the fixed IS-MBR procedure. To reduce the computational cost, we first sample with $N = 128$, and perform bootstrap resampling to generate the repetitions for different N. In Figure 2, we observe that for both Qwen2.5-Math-7B-Instruct and Llama-3.1-8B-Instruct, $\hat{\text{Var}}(y_{\text{IS-MBR}})$ exhibits a power law relationship with N_{eff}, an observation that aligns with the theoretical $\mathcal{O}(1/N_{eff})$ asymptotic variance of Monte Carlo estimators.

C.2 Relevant Support Mass Convergence

The RSM diagnostic is designed to approximate the relevant coverage of our support with respect to the MBR estimator. Concretely, it measures the amount of posterior mass in our approximation of the target distribution that is located in high-reward response regions. This too is a statistical estimation problem that improves with the number of samples drawn. For $\hat{\pi}_{N=1}$, $\hat{P}_\pi(r \geq \tau)$ is trivially 0 or 1. However, as $N \to \infty$, our estimate of $\hat{P}_\pi(r \geq \tau)$ improves. In Figure 3, we capture this uncertainty by showing the empirical estimates of $\hat{P}_\pi(r \geq \tau)$ as a function of N over each question in the `MATH500` dataset, using the same models and parameters as in the main experiments. While the mean $\hat{P}_\pi(r \geq \tau)$ over all questions is relatively constant for all N, the variance is not, and shrinks as N increases. This is also apparent through the sample trajectories, which oscillate for small N, before converging. Intuitively, for very easy or very hard problems, IS can consistently sample very good/bad responses, and thus quickly "push" $\hat{P}_\pi(r \geq \tau)$ to converge to either a very

high or low RSM. In contrast, for "borderline" problems, $\hat{P}_\pi(r \geq \tau)$ will oscillate for longer as better and worse responses are iteratively sampled. This separation is also impacted by calibration. For a well calibrated reward model, the variance between correct and incorrect responses ceases to overlap relatively quickly, while for a non-calibrated reward model, this overlap persists for longer, though still shrinks. Regardless, for a given model, we can assess the quality of our test static (e.g. final MBR response) via $\hat{P}_\pi(r \geq \tau)$ with increasing certainty as N increases, and use this as a signal for compute allocation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our methods, experiments, and analyses reflect the claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the relevant limitations several times throughout our paper. Please see the conclusion and methods section for more details.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We make no novel theoretical claims in our paper. For any theoretical guarantees we ensure an appropriate reference to the literature.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include full experimental details in our methods and results section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to author confidentiality, we do not currently include code. However, we aim to completely open source our code upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include full experimental details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to computational limitations, we were not able to rerun each experiment multiple times to produce error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include relevant details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We can confirm that our research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: To the best of our knowledge, our work does not entail any negative societal impacts. Our work has several positive impacts, namely it improves the availability and capacity of LLM inference time scaling for users with limited compute budgets.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We do not release any data or models of our own, and we only use off-the-shelf open source models. Therefore there are no possibilities of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We ensure proper citations to the data, code, and models we use. We only use open-source artifacts, so the license and terms are respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets are introduced in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not include any human subject experiments or crowdsourcing experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not include any human subjects, and we did not need IRB approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA] .

    Justification: Our research is on LLMs, but LLMs were not used in any part of the core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.