Optimal Policy Minimum Bayesian Risk

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inference scaling can help LLMs solve complex reasoning problems through extended runtime computation. On top of targeted supervision for long chain-of-thought (long-CoT) generation, purely inference-time techniques such as best-of-N (BoN) sampling, majority voting, or more generally, minimum Bayes risk decoding (MBRD), can further improve LLM accuracy by generating multiple candidate solutions and aggregating over them. These methods typically leverage additional signals in the form of reward models and risk/similarity functions that compare generated samples, e.g., exact match in some normalized space or standard similarity metrics such as Rouge. Here we present a novel method for incorporating reward and risk/similarity signals into MBRD. Based on the concept of optimal policy in KL-controlled reinforcement learning, our framework provides a simple and well-defined mechanism for leveraging such signals, offering several advantages over traditional inference-time methods: higher robustness, improved accuracy, and well-understood asymptotic behavior. In addition, it allows for the development of a sample-efficient variant of MBRD that can adjust the number of samples to generate according to the difficulty of the problem, without relying on majority vote counts. We empirically demonstrate the advantages of our approach on math (MATH-500) and coding (HumanEval) tasks using recent open-source models. We also present a comprehensive analysis of its accuracy-compute trade-offs.

1 Introduction

Recent progress in large language model (LLM) technologies has reignited interest in decoding methods, and in general in scaling laws for inference time compute. Reasoning models, such as OpenAI's O1, O3 and O4-mini [Jaech et al., 2024, OpenAI, 2025], Alibaba's Qwen with Questions[1] and DeepSeek's R1 [Guo et al., 2025], can learn to produce long chains of thought (long-CoT) that solve very hard problems by using reinforcement learning with verifiable rewards. At the same time, there is a general resurgence of interest in decoding methods beyond simple greedy decoding or sampling, see the recent NeurIPS tutorial [Welleck et al., 2024]. Among these are methods that rely on complex tree traversal, such as Monte Carlo tree search [Browne et al., 2012, Chen et al., 2024] and different variants of beam search. A second category includes best-of-N (BoN) decoding and self-ensembling techniques that can exploit additional signal such as process reward models [Uesato et al., 2022, Lightman et al., 2024] or some measure of consistency across multiple model outputs – these can be generally regarded as variants of minimum Bayesian risk decoding (MBRD) [Kumar and Byrne, 2004, Bertsch et al., 2023].

A large body of recent research has focused on long-CoT inference scaling. Monte Carlo tree search has also received significant attention due to its success in other machine learning areas such as games. In this paper, we focus on the remaining category of BoN/MBRD methods, which offer a

[1] https://qwenlm.github.io/blog/qwq-32b-preview/

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

unique and complementary set of advantages over other approaches. First, these methods can be used with any model, irrespective of whether the latter was trained for long-CoT reasoning. Second, their implementation is relatively simple, often requiring only parallel decoding and a final integration of results, which gives more precise control over inference scaling budget. Finally, they are modular, allowing for the use of off-the-shelf reward models to support multiple domains, even beyond those where long-CoT is particularly advantageous. On the flip side, they typically provide a lower gain in accuracy for the same amount of compute[2]. Whether BoN or MBRD is the superior option in a given scenario is dependent on the quality of the generator relative to that of the reward model. Finally, the simplicity of BoN and MBRD leaves less room for modification and improvement.

In this work, we propose an improvement to standard MBRD, termed Optimal Policy MBRD (OP-MBRD), with the following desirable properties:

- Robust response and performance across different scenarios, outperforming or closely matching the better of BoN and MBRD, even when there is a large performance gap between the two.
- Well-understood asymptotic behavior, converging to regular MBRD over a distribution that balances the contributions of a reward model and a reference generator.
- A sample-efficient version of the algorithm that can adjust the number of generated samples depending on the difficulty of the prompt, relying on general string matching instead of exact match counts.
- OP-MBRD retains most of the simplicity of BoN and MBRD, while introducing only a single new parameter and remaining compatible with standard MBRD (i.e., beyond the use of simple exact match as in ordinary majority voting).

2 Related work

The following approaches can be considered related to our work.

Learned Chain-of-Thought Techniques: These inference scaling methods achieve high performance by training the generator to produce chain of thought to solve difficult problems. These CoTs often contain spontaneously appearing instances of self-reflection, backtracking, option enumeration, summarization and others. The most successful models are trained with reinforcement learning and verifiable rewards [Jaech et al., 2024, OpenAI, 2025, Guo et al., 2025]. There are however more structured approaches that consider specific types of skills in the CoT [Kumar et al., 2024, Gandhi et al., 2025]. Compared with the approach presented here, these techniques require specific generator training but no special decoding, besides support for long context. They can be considered complementary to the technique presented here.

Majority Voting and Bayesian Risk: This concerns inference scaling methods that generate multiple outputs from a model and consolidate them into a final answer. Their main advantage is their simplicity, requiring only N independent generations, no specific generator training, or need of an external model such as reward models. Recent variants applied to LLMs include self-consistency [Wang et al., 2023], and conventional counts-based consensus in mathematical reasoning [Yang et al., 2024, Guo et al., 2025]. Minimum Bayes risk decoding [Kumar and Byrne, 2004] can be considered a generalization of majority voting (MV) that utilizes a risk or similarity function between pairs of outputs to select the output that has the lowest risk / highest similarity with regard to any other output. MBRD reduces to count-based MV by using exact match as similarity [Bertsch et al., 2023]. MBRD allows to extend MV to domains where exact match is not an option. A disadvantage of MV methods is that they often require large amount of samples to yield good gains.

Reward-weighted Post-Processing: These can be seen as an enhancement of the previous. They generate N independent sentences in the same way, but utilize a separate model to score the completed outputs e.g. a reward model. Best-of-N (BoN) [Charniak and Johnson, 2005] is a very common and simple method with proven success to enhance LLM performance [Yang et al., 2024]. These methods also include combinations of majority voting with reward models, which provide improved performance with respect to plain majority voting, e.g., voting verifiers [Li et al., 2023]. These methods can also be expressed as a form of MBRD where pair-wise risk/similarity is enhanced with

[2]Early DeepSeek-R1 release note `https://api-docs.deepseek.com/news/news1120`.

the output reward. Reward post-processing compensates some of the limitations of pure MV while adding only the overhead of a call to to an external reward model. The method here presented falls into this category. As shown in the next sections it retains the simplicity advantage of similar counterparts such as BoN or voting verifiers, while being derived from well understood principles, providing more robust performance and an efficient version with better performance/compute trade-offs.

Step-by-Step Decoding: This includes methods that generate N outputs in steps. After each step all partial completions are scored and combined together to produce the prefixes for the next step. This can be done through deterministic pruning of the worse options as in beam search [Graves, 2012], or stochastic re-sampling of candidates as in [Deng and Raffel, 2023]. Scorers can be reward models [Deng and Raffel, 2023], but also attribute classifiers [Yang and Klein, 2021]. With the rise of reasoning LLMs and process reward models (PRMs) [Lightman et al., 2024], able to score partial reasoning chains, steps have grown fro single tokens to multiple, although the basic results are maintained [3]. Compared to reward-weighted post-processing techniques, step-by-step additional complexity due to the need to synchronize intermediate steps and the extra communication overhead per step between generator and scorer. The technique introduced here could also be applied however to multi-step algorithms, but this is beyond the current scope of the manuscript.

Efficient Inference Scaling Optimal allocation of inference compute can enable inference-time methods to outperform simply using a larger model Snell et al. [2024]. Given the recent success of inference scaling methods, several approaches have been proposed in this area. One category of methods estimates task difficulty and performs budget allocation or input routing to different generators [Damani et al., 2024]. Other approaches focus on minimizing the number of samples generated or compared, based on the observed distribution of answers over multiple samples [Aggarwal et al., 2023] or pair-wise risk [Cheng and Vlachos, 2023].

Optimal Policy in Reinforcement Learning The method introduced here is also related in its mathematical background to recent works in the reinforcement learning. Methods like proximal policy optimization (PPO) [Schulman et al., 2017], GDC++ [Korbak et al., 2022] and BRAIn [Pandey et al., 2024] optimize the KL-controlled reward maximization objective through different approximations. The solution to this problem is the intractable optimal policy, which is the basis of the decoding algorithm presented here. Direct preference optimization (DPO) [Rafailov et al., 2023] is also derived from the optimal policy and can be proven to be a special case of BRAIn. Rejection sampling from the optimal posterior is also considered in statistical rejection sampling optimization [Liu et al., 2024] as a method similar to DPO.

3 Minimum Bayesian Risk Decoding

We define an LLM as a neural network parameterizing a distribution $p(y \mid x)$ over strings. Here $x, y \in V^+$ are input and output strings, respectively, and V^+ is the countably infinite set of all possible strings formed by concatenating tokens from a vocabulary V. Generating from an LLM generally corresponds to finding the most likely string

$$\hat{y} = \arg\max_{y \in V^+} \left\{ p(y \mid x) \right\} \tag{1}$$

which can only be approximately computed in practice using techniques such as greedy search.

In recent years, the increase in performance of LLMs has made sampling also a viable option. For auto-regressive models, this is usually done using ancestral sampling, often with some re-shaping of the token distribution by setting the temperature or nucleus size [Holtzman et al., 2019].

In this context, techniques that aggregate over multiple outputs of the same model have become a simple yet powerful way to further boost results. This is best exemplified by the resurgence of minimum Bayesian risk decoding (MBRD) and related methods applied to LLMs, such as self-consistency [Wang et al., 2023], as well as strong results for "consensus" in reasoning models such as OpenAI's O1, O3 or DeepSeek's R1, which can be interpreted as MBRD with exact match.

MBRD solves an alternative decoding problem, where the goal is to find the output that minimizes the expected risk with respect to the LLM distribution. In the remainder of this manuscript, we will

[3] For recent results combining step-by-step and post-processing see `https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute`

refer instead to the mathematically equivalent problem of maximizing the expected similarity, which makes notation simpler. This problem can be expressed as

$$\hat{y} = \arg\max_{y' \in V^+} \left\{ \mathbb{E}_{p(y|x)} \{ M(y, y', x) \} \right\} = \arg\max_{y' \in V^+} \left\{ Q(y', x) \right\} \tag{2}$$

where $M(y, y', x)$ is a similarity function between outputs $y, y' \in V^+$. [4]. Exact MBRD is doubly intractable since it requires the same search over V^+ as greedy decoding, but also the computation of the expectation over that same domain. For this reason, MBRD is often approximated through Monte Carlo estimation by using a set of samples $\mathcal{S}(N) = \{y_1 \cdots y_N\} \sim p(y \mid x)$, both as the search space and to compute the expectation:

$$Q(y', x) = \mathbb{E}_{p(y|x)} \{ M(y, y', x) \} \approx \frac{1}{N} \sum_{y_n \in \mathcal{S}(N)} M(y_n, y', x). \tag{3}$$

Often described as *consensus* or *majority voting*, MBRD using exact match similarity, henceforth referred to as MBRD (EM), is a well known and strong baseline

$$M(y, y') = \delta_{g(y,y')}. \tag{4}$$

Here $g()$ is a function that extracts an answer from each model output (which may contain CoT and other tokens), compares them using some normalization, e.g., a symbolic representation, and returns 1 if they are equal or 0 otherwise. This amounts to selecting the answer that occurs more often in this normalized space. Other forms of MBRD include using symbol-level distances such as Rouge [Lin, 2004]. In some fields like machine translation, evaluation metrics like BLEU [González-Rubio et al., 2011] or COMET [Guttmann et al., 2024] are also used. For LLMs, it is straightforward to incorporate a reward model into the risk/similarity computation as

$$M(y, y', x) = \delta_{g(y,y')} \cdot R(y, x), \tag{5}$$

which can substantially improve performance. This can also be viewed as an instance of a voting verifier [Li et al., 2023]. This setup will henceforth be referred to as MBRD (EM*R).

4 Optimal Policy Minimum Bayesian Risk Decoding

4.1 Definition

We here propose another way of combining $p(y \mid x)$ and $R(y, x)$ that represents a minimum increase in complexity, while providing interesting properties. Borrowing from Reinforcement Learning, one can define a distribution q that maximizes a expected reward $R(y, x)$ while being close to a reference distribution $p_R(y \mid x)$. This can be expressed as the objective

$$\mathcal{L}(q) = \mathbb{E}_{q(y|x)} \left\{ R(y, x) \right\} - \beta \cdot \mathrm{KL}(q(y \mid x) \mid\mid p_R(y \mid x)) \tag{6}$$

where β controls how much influence the reward has on q. This objective is the well known KL-controlled reward maximization, which is the basis for RL algorithms such as PPO [Schulman et al., 2017], GDC++ [Korbak et al., 2022], DPO [Rafailov et al., 2023] and BRAIn [Pandey et al., 2024]. It is easy to see that the solution to this is given by the optimal policy [5]

$$p^*(y \mid x) = \arg\max_q \{ \mathcal{L}(q) \} = \frac{1}{Z} \cdot p_R(y \mid x) \cdot \exp\left(\frac{1}{\beta} R(y, x) \right) \tag{7}$$

where the partition function Z requires an intractable sum over the space of sentences V^+. Assuming that we could sample from this distribution, it's trivial to do MBRD with this optimal posterior

$$\hat{y} = \arg\max_{y' \in V^+} \{ \mathbb{E}_{p^*(y|x)} \{ M(y, y', x) \} \} \tag{8}$$

This formulation provides a well defined way of integrating a reward $R(y, x)$, a similarity function $M(y, y', x)$, a reference model $p_R(y \mid x)$, and an available generator $p(y \mid x)$.

[4] For generality, we have also included the input in this function, since it does not alter the formulation.

[5] For a formulation, see for example Rafailov et al. [2023] Appendix A.1.

4.1.1 Computing Expectations with respect to the Optimal Policy

To approximate MBRD expectations we need to sample from an intractable energy model, in particular from the optimal policy. This has been addressed before in the literature but for the purpose of Reinforcement Learning training (DPG, GDC++, RSO [Liu et al., 2024], BRAIn). It can be shown that, given a sample from a proposal distribution, in this case assumed to be our generator $p(y \mid x)$, the probability of the sample $y_n \in \mathcal{S}(N)$ belonging to $p^*(y \mid x)$ is given by[6]

$$p(\text{accept } y_n) = \exp\left(\tilde{R}(y_n, x) - \max_{y'} \tilde{R}(y', x)\right) \tag{9}$$

where

$$\tilde{R}(y, x) = \frac{R(y, x)}{\beta} + \log \frac{p_R(y \mid x)}{p(y \mid x)} \tag{10}$$

It seems intuitive that just using the accepted samples to compute the expectation is the best option. However, it is well-known that the Rao-Blackwellized version [Casella and Robert, 1996][7] of this estimator can use all samples to provide a lower variance estimate. This can further be approximated via importance sampling to yield

$$\hat{Q}(y', x)^{\text{OP}} = \frac{1}{N} \sum_{y_n \in \mathcal{S}(N)} M(y_n, y', x) \cdot \frac{p(\text{accept } y_n)}{\sum_{y'_n \in \mathcal{S}(N)} p(\text{accept } y'_n)} \tag{11}$$

Since softmax is invariant to shifting the logits by a constant, Rao-Blackwellized rejection sampling in Eq. 11 coincides with self-normalized importance sampling (SNIS) [Bengio and Senécal, 2008] with unnormalized weights p_n. A bias subtracted version of SNIS is also used in BRAIn, but for the purpose of reinforcement learning. We term this last estimator Optimal Policy Minimum Bayesian Risk (OP-MBR) and its maximization OP-MBR Decoding (OP-MBRD).

4.2 OP-MBRD with a Process Reward Model

The method introduced here provides a well defined way to integrate a reward model $R(y, x)$, a reference model $p_R(y \mid x)$, and a similarity function $M(y_n, y', x)$ into a decoding strategy for a generator $p(y \mid x)$. It does not prescribe which values should these take. In the case of inference scaling, PRMs estimate the odds that a given partial reasoning leads to the correct answer. For these, the acceptance probability can thus be defined as the product of acceptance of every step, leading to

$$\prod_{t=1}^{T} p(\text{accept } y^n_{<t+1}) = \exp\left(\sum_{t=1}^{T} \frac{\text{PRM}(y^n_{<t+1}, x)}{\beta} + \log \frac{p_R(y_n \mid x)}{p(y_n \mid x)} - M\right) \tag{12}$$

where M is the sum of maximum $\tilde{R}$ for each step that ensures we are multiplying probabilities. In practice we normalize the sum of PRM scores by the number of steps T. Note that this does not require step-by-step decoding. The outputs are fed to the PRM at the end of generation with appropriate markers i.e. double end of line, and the PRM returns scores for what it considers steps. Another possible interpretation of this formula is ancestral importance sampling of the optimal policy.

4.3 Efficient OP-MBRD

Since OP-MBRD Rao-Blackwellized rejection sampling and the importance sampling estimators coincide, it may seem that the rejection sampling formulation is redundant. Nevertheless, one can still derive a useful metric from it, the number of expected optimal policy samples for a sample set $\mathcal{S}(N)$

$$\hat{N}^{\text{OP}} = \sum_{y_n \in \mathcal{S}(N)} p_n = \sum_{y_n \in \mathcal{S}(N)} \exp\left(\tilde{R}(y_n, x) - \max_{y'} \tilde{R}(y', x)\right). \tag{13}$$

Under the rejection sampling interpretation of our estimator, this gives us a measure of how successful our sampling round was, with a higher $\hat{N}^{\text{OP}}$ indicating more samples belong to p^*. We can use this to derive an efficient version of OP-MBRD, where we fix a desired number of optimal policy samples N^{OP} and we sample repeatedly until $\hat{N}^{\text{OP}} \geq N^{\text{OP}}$. As it will be shown in the experimental setup, these yields a good prediction of task difficulty for generator-PRM pairs. We will describe those pairs has being *well calibrated*. We will henceforth refer to this proposed method as OPE-MBRD.

[6] See e.g. RSO [Liu et al., 2024] Appendix A.1

[7] See `https://andrewcharlesjones.github.io/journal/rao-blackwellization.html`

4.4 Formal Guarantees

Unlike other methods that combine majority voting and reward models, like MBRD (EM*R), OP-MBRD has a clearly defined asymptotic behavior, trivially following from the properties of self-normalized importance sampling[8]. The proposed OP-MBRD estimator converges to the true MBRD with respect to the optimal policy with probability 1.

$$p\Big(\lim_{N\to\infty} \hat{Q}(y', x)^{\text{OP}} = \mathbb{E}_{p^*(y|x)}\{M(y, y', x)\}\Big) = 1 \tag{14}$$

Furthermore, we can examine in detail Eqs 9,10 to study what sampling from the optimal policy entails. For cases in which $R = 0$ only the log-ratio term remains and OP-MBRD reduces to MBRD from $p_R(y \mid x)$, as approximated by self-normalized importance sampling. If we use our generator as reference $p_R(y \mid x) = p(y \mid x)$, only the reward term R remains. For a PRM this now will represent MBRD with respect to an energy model proportional to the odds of reaching the correct answer. For an oracle PRM this would assign zero weight to any sample in $S(N)$ not reaching the correct answer, which in the limit guarantees that OP-MBRD would always choose the right answer [9].

5 Experimental Setup

5.1 Models and Datasets

To evaluate the methods proposed, we test small and medium LLMs on math and coding tasks. For reproducibility and completeness, we select recent open source models in the 1-20 billion parameter range. For math, we select Alibaba's Qwen-2.5-math models [Abdin et al., 2024] sizes 1.5b and 7b as high performing math-specific models. These have a matching process reward model – Qwen-2.5-PRM-7b [Zhang et al., 2025] – that we also use in our experiments. We also select IBM's Granite 3.3[10] models sizes 2b and 8b. These are general models that also exhibit strong math performance compared to, e.g., LLaMa models of the same size [Grattafiori et al., 2024]. For the Granite models, we train our own PRM from Granite-3.3 for math. The model was trained with synthetically generated data. The training data consists of step-level correctness annotations, generated using the binary search method of Luo et al. [2024]. The input prompts are sampled from MathInstruct [Yue et al., 2023], MetaMathQA [Yu et al., 2023] and NuminaMath [Li et al., 2024] datasets – the responses are sampled from Granite-3.x, Mixtral-8x22B and Phi4-instruct models. After initial training of the PRM with this data, we use the trained PRM to further filter out low-quality step annotations. We discard samples where step-level correctness annotations do not match the PRM's assessment of step quality. We then perform a second iteration of PRM training with this higher-quality filtered data. As the upper tier in size we select again Phi-4-instruct [Abdin et al., 2024] (14b) as an additional generator. Finally, we pair Phi4-instruct also with a Phi4-PRM trained the same way. We do not include long-CoT models since we are focusing here on approaches leveraging BoN and MBRD, which leverage independent samples rather than long contexts.

We evaluate all models and methods on MATH-500 [Hendrycks et al., 2021, Lightman et al., 2024] and HumanEval [Chen et al., 2021]. MATH-500 is a collection of 500 math competition problems that require detailed step-by-step reasoning to solve. HumanEval consists of 164 programming problems, each asking to complete a standalone Python function from requirements specified in a docstring. Unit tests are included for each example for automatic evaluation. We use pass@1 scores to assess performance on both datasets.

5.2 Baselines and Methods

As inference scaling baselines we focus on well established single-step algorithms that ensemble multiple samples. We consider BoN using the average PRM score across steps, which was observed to be more performant than other aggregations like minimum in these datasets. We use also two variants of MBRD. First, variants not making use of a PRM or any other parametric scoring function. For MATH-500 we use exact match similarity, $M(y, y') = \delta_{g(y,y')}$. This is often also described as majority voting or consensus and is here referred as MBRD (EM). As text normalizer $g()$ we

[8] For a derivation see `https://www.tuananhle.co.uk/notes/is.html`

[9] Assuming a perfect PRM, $\lim_{N\to\infty}$ and model assigning non zero probability to the solution

[10] `https://huggingface.co/ibm-granite/granite-3.3-8b-instruct`

extract the answer inside the a boxed command, including a prompt to force the model to adhere to such format. For code, exact match performs very poorly, we use Rouge [Lin, 2004] instead, $M(y, y') = \text{rouge}(y, y')$. This performed better in our initial test than other alternatives such as Python's `difflib.SequenceMatcher` [Ratcliff and Metzener, 1988, Wei et al., 2025]. We refer to this as MBRD (rouge). As parametric MBRD baseline, we used Voting Verifier [Li et al., 2023], which can be expressed as $M(y, y') = \delta_{g(y,y')} \cdot \text{meanPRM}(y, x)$ here referred to as MBRD (EM*R).

As methods proposed, we introduce Optimal Policy variants of the non-parametric MBRD i.e. OP-MBRD (EM) and OP-MBRD (rouge). These use the Rao-Blackwellized rejection sampling (or equivalently importance sampling) to sample from the optimal posterior (see Section 4). All our experiments use $p_R(y \mid x) = p(y \mid x)$, which in practice nullifies the effect of the log-ratio term of OP-MBRD. Although the log-ratio term can be proved to equate rejection/importance sampling of p_R (see Section 4.4), initial experiments did not show big advantages when using strong teachers for p_R and we leave further exploration for future work. This also allowed to estimate the maximum reward in Eq. 9 as the maximum PRM value 1.0. In addition to the normal variants, we also used the efficient version proposed in Section 4.3, termed OPE-MBRD, which uses the expected amount of accepted samples to decide when to stop sampling. For this we iteratively sampled outputs one by one until a target budget of $N^{\text{OP}} = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ optimal samples was met. These experiments are designed to measure the gains in throughput and not in wall-clock time. For the latter, a schedule would have to be designed that uses the observed probability of success to guess a fixed number of samples to be generated next. We leave this for future work.

5.3 Hyperparameters and Variance Reduction

For hyperparameter tuning, we construct a development set out of NuminaMath[11]. We include a random subset of 500 question-answer pairs in this set, discarding their CoTs, and making sure they (a) pass simple format check, and (b) are not in MATH-500. We set the KL-term weight β, representing the relative weight of the generator versus the PRM in this set. A value of $\beta = 0.1$ was found to be robust across many scenarios and was selected for Qwen and Granite models both for math and code tasks. The only clear exception was Qwen-15b/Qwen-7B-PRM. Results on the dev set indicated that, for this pair, the PRM is much stronger than the generator, and a value of $\beta = 0.001$ was selected. Similarly a value of $\beta = 0.01$ was selected for Phi-4/Phi-4-PRM. For the OPE-MBRD a maximum number of samples was set as a $\times 10$ multiplier of the number of optimal samples selected. For e.g. if we solicited 2 optimal samples, no more than 20 real samples would be sampled. This was a simple compromise that helped with badly calibrated generator-PRM pairs, that tend to have spikes in the number of samples solicited. We include the full dev details in Appendix B.

To cover a wide range of inference scaling cases, we produce $N = 256$ samples for each input. To reduce variance of results, we always make use of the pool of 256 samples for all experiments, either for ensembling or experiment repetition. For example, for MBRD (EM) with 16 samples, we can repeat the experiment $256/16 = 16$ times. We show the average performance for these 16 repetitions. Note that for the efficient version of OP-MBRD, OPE-MBRD, the number of samples that constitutes an experiment changes, since the algorithm can select a different number of samples for different generations. We consume blocks of samples of variable size until exhausting the sample pool to construct experiment repetitions. No sample is ever shared across experiments. Standard deviation across all conditions ranged from 0.3 to 0.5.

5.4 Results Analysis

Figure 1 shows the comparison of the different generator-PRM pairs. The left shows pass@1 performance as a function of the real number of samples generated, averaged over all dataset examples. The right side shows study cases for specific optimal policy budgets of OPE-MBRD, signaled with a cyan star on the left side of the plot. Each marker on the right represents an example in the MATH-500 dataset, sorted from lower to higher difficulty. The difficulty is assessed by using the pass@1 of the normal generator $p(y \mid x)$ and the full 256 samples. On the vertical axis we display the real number of samples N used by OPE-MBRD averaged over experiment repetitions. We color as green instances for which the OPE-MBRD attains higher pass@1 than OP-MBRD, red if lower, black if both match (typically both reach 1.0) and gray if both fail (0.0). We consider generator-PRM

[11] `https://huggingface.co/datasets/AI-MO/NuminaMath-CoT`

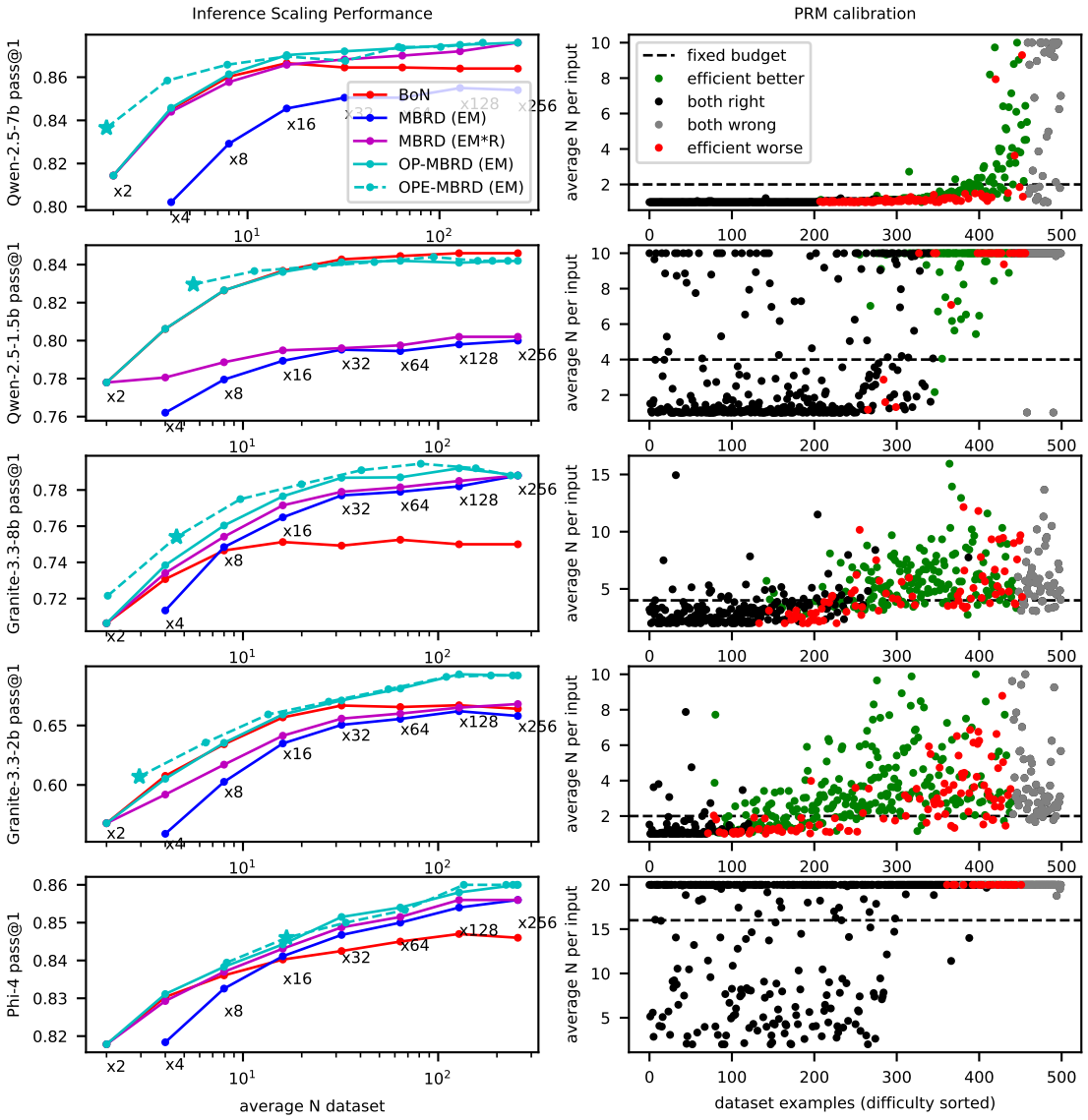

Figure 1: MATH-500 test results. Left: pass@1 score as a function of the number of samples per input. Every dot marker is an average over multiple experiment repetitions. A star marker denotes the efficient OP-MBRD represented on the right side. Right: Number of samples OPE-MBRD selects for every example in the test set, sorted from easy to difficult by regular decoding pass@1. Largest standard deviation among all experiments was less than 0.5.

pair as calibrated if the number of samples used increases with problem difficulty and this leads to performance improvements (green dot).

Looking at Figure 1, left: In terms of pass@1 performance, OP-MBRD performs robustly across scenarios and is mostly above or equal to the best baseline, which alternates between BoN or MBRD (EM*R). For the stronger Qwen-7b-math/Qwen-PRM-7B, results match or slightly outperform the baseline MBRD (EM*R). For the efficient version OPE-MBRD, large gains in performance at attained at low numbers of samples- this is consistent with the excellent calibration where the

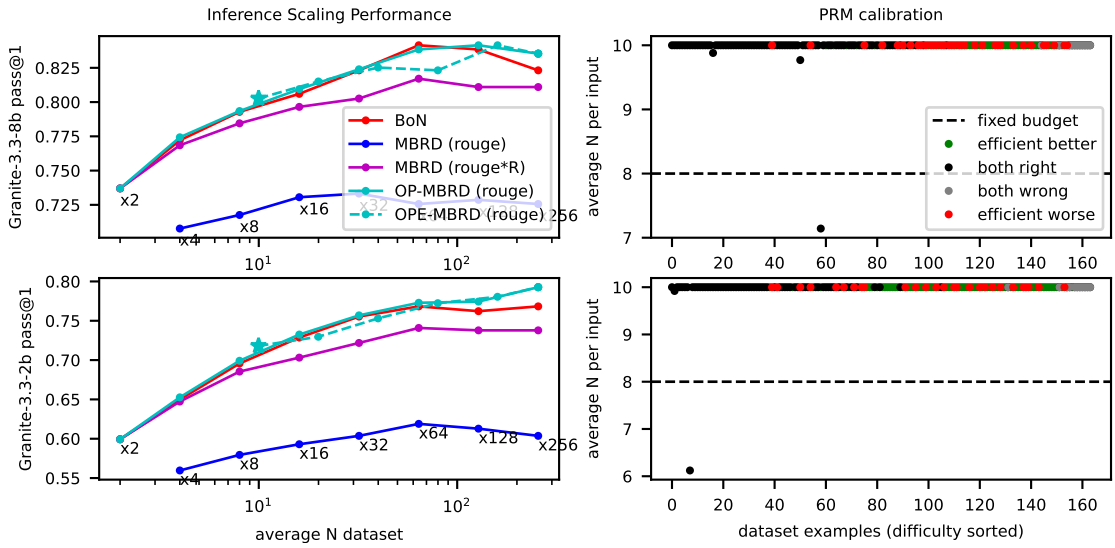

Figure 2: HumanEval test results: Left: pass@1 score as a function of the number of samples per input. Every dot marker is an average over multiple experiment repetitions. A star marker denotes the efficient OP-MBRD represented on the right side. Right: Number of samples OPE-MBRD selects for every example in the test set, sorted from easy to difficult by regular decoding pass@1. Largest standard deviation among all experiments was less than 0.5.

OPE-MBRD version select one real sample for all but the hardest 15% of all examples. In the smaller Qwen case the BoN baseline attains much better performance than MBRD (EM*R) baseline, but OP-MBRD closely matches it. Despite the worse calibration OPE-MBRD still provides a good advantage. For the Granite/Granite-PRM pairs, which are weaker at math, OP-MBRD provides an advantage over the baseline MBRD (EM*R), with particularly strong gains for the smaller model and high number of samples. Both results show reasonably good, but noisier, calibration which leads to OPE-MBRD providing gains over OP-MBRD. Phi-4/Phi-4-PRM presents the worst calibration, which leads to OPE-MBRD just matching OP-MBRD, but overall gaining a small advantage against the best baseline MBRD (EM*R). The lack of Phi-4/Phi-4-PRM calibration may stem from the fact that PRM development was mostly centered around the Granite models.

Figure 2 shows additional results for Granite/Phi-4-PRM pairs[12] on the the HumanEval coding task. All metrics and symbol meanings are same as before. Overall, OP-MBRD remains close to the best performing baseline, in this case BoN. Calibration in this case is non-existent, which can be explained by the fact that we use a PRM tuned on math data to judge a coding task, resulting in very low overall PRM scores and very pessimistic (high) number of samples solicited. As with MATH-500, OP-MBRD still provides an advantage for the smaller model and at higher sample counts.

6 Conclusions

We present Optimal Policy Minimum Bayesian Risk Decoding (OP-MBRD), a simple alternative to BoN and MBRD with rewards that performs more robustly across different generator-PRM combinations. OP-MBRD also has well-defined asymptotic behavior interpolating, in an interpretable way, between rejection/importance sampling from a target generator and sampling from an energy model associated to a reward model. Finally, the proposed formulation also yields an additional useful signal that can suggest a variable number of samples based on input difficulty. For well-calibrated generator-PRM pairs, this results in large gains in throughput for the same compute budget, without relying on answer counts or risk/value functions. Future work can expand on the role of the reference generator and look into efficient multi-step algorithms, for which the properties of the presented method are well-suited.

[12]We paired Granite with Phi-4-PRM since it showed better overall performance on the coding task.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024. URL https://arxiv.org/abs/2412.08905. pages 6

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/. pages 3

Yoshua Bengio and Jean-Sébastien Senécal. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *IEEE Transactions on Neural Networks*, 19(4):713–722, 2008. pages 5

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In Yanai Elazar, Allyson Ettinger, Nora Kassner, Sebastian Ruder, and Noah A. Smith, editors, *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.bigpicture-1.9. URL https://aclanthology.org/2023.bigpicture-1.9/. pages 1, 2

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012. pages 1

George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83 (1):81–94, 1996. URL http://www.jstor.org/stable/2337434. pages 5

Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, 2005. URL https://aclanthology.org/P05-1022/. pages 2

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: process supervision without process. *arXiv preprint arXiv:2405.03553*, 2024. pages 1

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374. pages 6

Julius Cheng and Andreas Vlachos. Faster minimum bayes risk decoding with confidence-based pruning. *arXiv preprint arXiv:2311.14919*, 2023. pages 3

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of lm computation. *arXiv preprint arXiv:2410.04707*, 2024. pages 3

Haikang Deng and Colin Raffel. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.721. URL https://aclanthology.org/2023.emnlp-main.721/. pages 3

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025. URL https://arxiv.org/abs/2503.01307. pages 2

Jesús González-Rubio, Alfons Juan, and Francisco Casacuberta. Minimum bayes-risk system combination. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 1268–1277, 2011. pages 4

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783. pages 6

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012. URL https://arxiv.org/abs/1211.3711. pages 3

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948. pages 1, 2

Kamil Guttmann, Mikołaj Pokrywka, Adrian Charkiewicz, and Artur Nowakowski. Chasing comet: Leveraging minimum bayes risk decoding for self-improving machine translation. *arXiv preprint arXiv:2405.11937*, 2024. pages 4

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe. pages 6

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019. pages 3

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. URL https://arxiv.org/abs/2412.16720. pages 1, 2

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67496dfa96afddab795530cc7c69b57a-Paper-Conference.pdf. pages 3, 4

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024. URL https://arxiv.org/abs/2409.12917. pages 2

Shankar Kumar and Bill Byrne. Minimum bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, 2004. URL https://aclanthology.org/N04-1022/. pages 1, 2

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath,

2024. URL https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf. pages 6

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.291. URL https://aclanthology.org/2023.acl-long.291/. pages 2, 4, 7

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*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi. pages 1, 3, 6

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. URL https://aclanthology.org/W04-1013. pages 4, 7

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xbjSwwrQOe. pages 3, 5

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision, 2024. URL https://arxiv.org/abs/2406.06592. pages 6

OpenAI. Openai o3 and o4-mini system card. 2025. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf. pages 1, 2

Gaurav Pandey, Yatin Nandwani, Tahira Naseem, Mayank Mishra, Guangxuan Xu, Dinesh Raghu, Sachindra Joshi, Asim Munawar, and Ramón Fernandez Astudillo. Brain: Bayesian reward-conditioned amortized inference for natural language generation from feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024. URL https://arxiv.org/abs/2402.02479. pages 3, 4

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. In *Thirty-seventh Conference on Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023. URL https://arxiv.org/abs/2305.18290. pages 3, 4

John W. Ratcliff and David E. Metzener. Pattern matching: The Gestalt approach. *Dr. Dobb's Journal*, 1988. URL https://www.drdobbs.com/database/pattern-matching-the-gestalt-approach/184407970. pages 7

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347. pages 3, 4

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. pages 3

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. pages 1

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw. pages 2, 3

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. SWE-RL: Advancing LLM reasoning via reinforcement learning on open software evolution, 2025. URL https://arxiv.org/abs/2502.18449. pages 7

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=eskQMcIbMS. Survey Certification. pages 1

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. URL https://arxiv.org/abs/2409.12122. pages 2

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276/. pages 3

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023. URL https://arxiv.org/abs/2309.12284. pages 6

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023. URL https://arxiv.org/abs/2309.05653. pages 6

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025. URL https://arxiv.org/abs/2501.07301. pages 6

A Limitations

The work presented here has the following limitations: Although we cover a diverse set of 3 generators and 3 PRMs covering 5 different sizes across the range of 1-20B parameters, this is not fully representative of all LLMs. In particular, larger models that are likely to have better generation capabilities would be interesting to look at. In this setting, MBRD could be expected to have additional advantages over BoN. Unfortunately, due to compute limitations, it was not possible to cover all such cases. Although we cover both math and coding tasks, we had to keep the scope limited due to both time and compute constraints. In particular, a separate development set for coding and a larger experimental setup would have provided better opportunity to explore the methods better. Other domains where MBRD is also well established, such as machine translation, could also have added value.

B Development Set Results

As stated in Section 5.3, we created a dev set for hyperparameter tuning based on NuminaMath[13] of the same size as MATH-500. We report full results on the development set under the same conditions as the MATH-500 test set in Figure 3. These results were used to tune $beta$ for different generator/PRM pairs. As it can be observed from the results, this dataset is harder than Math-500, but model/PRM calibration is similar. Overall improvements with OP-MBRD are also larger, but this can be due to tuning effects.

[13] https://huggingface.co/datasets/AI-MO/NuminaMath-CoT

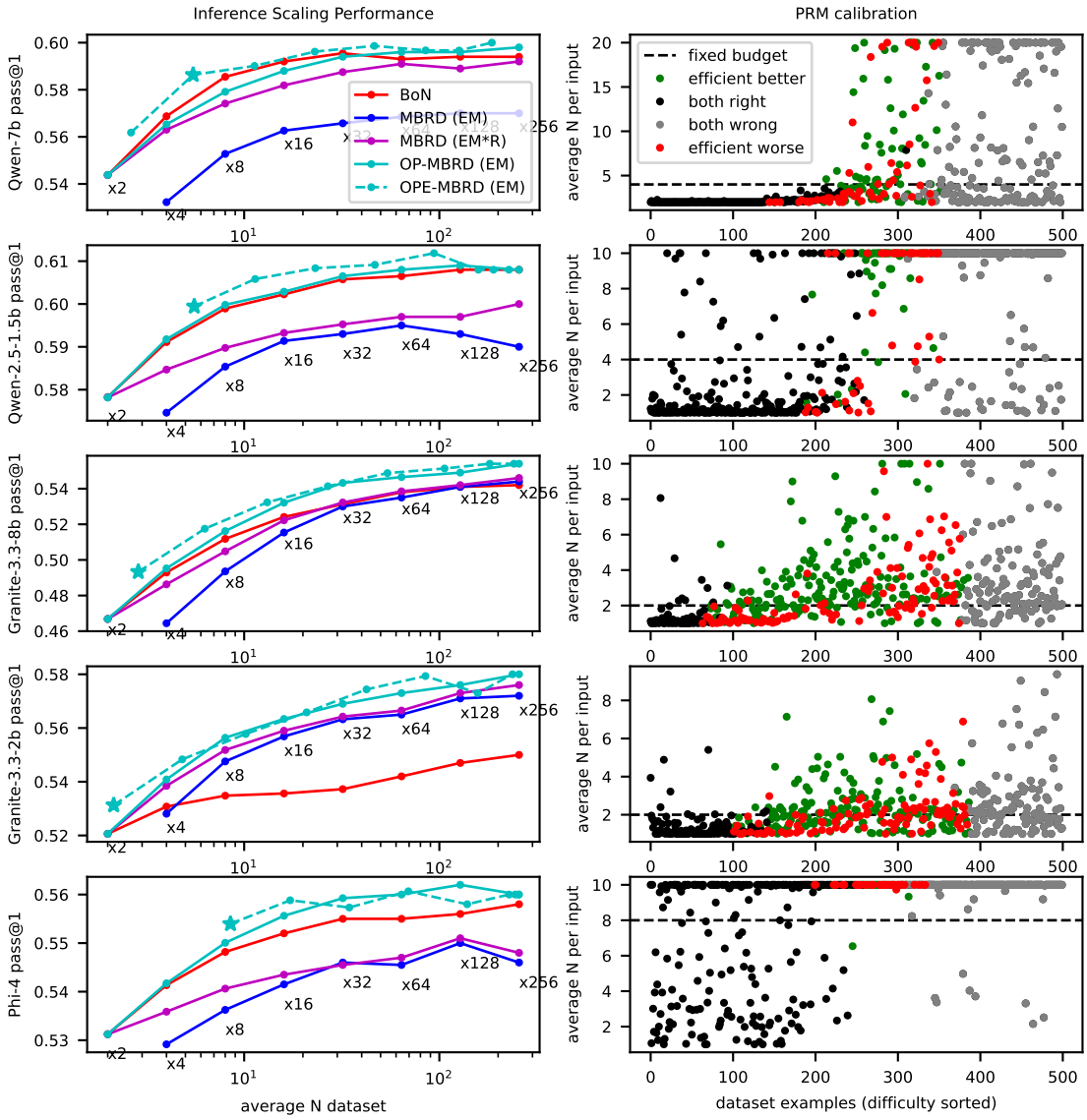

Figure 3: Development set (Numinamath-500) results.

C Hardware

Runtime experiments were carried out on a private H100 cluster. The code was a fork of math-eval-harness, concretely the one in [14]. Models were served using VLLM[15]. Some steps like computation of MBRD similarity were carried out on standard CPUs. The Phi-4 PRM model training was carried

[14] https://github.com/QwenLM/Qwen2.5-Math
[15] https://github.com/vllm-project/vllm

out on 8 H100 GPUs in a private cluster, and inference was done on a single H100 GPU using the Hugging Face `Transformers` library.

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: The claims in the Abstract and Section 1 third paragraph have been substantiated in Sections 4 and 5.

   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: Please see the Limitations section A.

   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: [Yes]

Justification: The paper itself does not introduce new theoretical results. All mathematical results are direct consequences of well-established formulations, which we refer appropriately to in each section.

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 have provided all details needed to reproduce our experimental results in Section 5.

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: [Yes]

Justification: The paper includes information necessary to implement the algorithms, which are generally simple. We will release the code upon acceptance of the manuscript. All datasets used to train the Phi-4 PRM will also be listed.

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: Please see Section 5.3.

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: [Yes]

Justification: We do not report error bars in the plots, as they would make the results hard to read. But we do report the range of standard deviations across experiments in Section 5.3. This is a narrow 0.3 to 0.5 across all conditions. If we take worst case two standard deviations are 1 point difference. As described in the section, every experiment is repeated a large number of times to reduce the effect of variance, up to 256 times. We are also careful not to claim superiority or inferiority of the proposed method except in situations where differences are well above two standard deviations.

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: Answer: Runtime experiments were carried out on commonly used hardware devices. An additional description of hardware has been included in Appendix C.

   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 have reviewed the NeuRIPS code of Ethics and are in compliance with it.

   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: [NA]

Justification: We propose a set of methods that attempt to find a high-quality answer from among a model's output samples, not otherwise affecting its response in any way or along any specific dimensions.

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: [NA]

Justification: The methods proposed here are not a specific source of harm beyond the usual potential harms of LLMs, against which the used models already have been post-trained. There is no indication that the techniques used in this paper increase the risks associated.

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: All used datasets and evaluation benchmarks are cited.

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: [Yes]

    Justification: We plan to release our code upon acceptance.

    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:

    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:

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: [Yes]

Justification: Our usage of LLMs – both generators and PRMS – has been described throughout the paper.

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.