Majority of the Bests: Improving Best-of-N via Bootstrapping

Amin Rakhsha[1,2,3*] **Kanika Madan**[3] **Tianyu Zhang**[3]

Amir-massoud Farahmand[4,5,1] **Amir Khasahmadi**[3]

[1]University of Toronto [2]Vector Institute [3]Autodesk

[4]Polytechnique Montréal [5]Mila - Quebec AI Institute

Abstract

Sampling multiple outputs from a Large Language Model (LLM) and selecting the most frequent (Self-consistency) or highest-scoring (Best-of-N) candidate is a popular approach to achieve higher accuracy in tasks with discrete final answers. Best-of-N (BoN) selects the output with the highest reward, and with perfect rewards, it often achieves near-perfect accuracy. With imperfect rewards from reward models, however, BoN fails to reliably find the correct answer and its performance degrades drastically. We consider the distribution of BoN's outputs and highlight that, although the correct answer does not usually have a probability close to one under imperfect rewards, it is often the most likely outcome. This suggests that the mode of this distribution can be more reliably correct than a sample from it. Based on this idea, we propose Majority-of-the-Bests (MoB), a novel selection mechanism that estimates the output distribution of BoN via bootstrapping and selects its mode. Experimental results across five benchmarks, three different base LLMs, and two reward models demonstrate consistent improvements over BoN in 25 out of 30 setups. We also provide theoretical results for the consistency of the bootstrapping. MoB serves as a simple, yet strong alternative to BoN and self-consistency, and more broadly, motivates further research in more nuanced selection mechanisms.[2]

1 Introduction

Scaling the inference-time computation of language models has led to a significant improvement of their performance on a variety of tasks (Brown et al., 2024; Snell et al., 2025; Wu et al., 2024; OpenAI, 2024; DeepSeek-AI, 2025). A growing number of methods have been introduced in this paradigm, such as generating long chains-of-thought (Wei et al., 2022; Muennighoff et al., 2025), asking the model to evaluate and improve its own outputs (Madaan et al., 2023), and tree search (Yao et al., 2023; Hao et al., 2023; Zhang et al., 2024). Another family of such algorithms, termed *sample-and-marginalize* by Wang et al. (2022), generate multiple outputs from the model and then aggregate them into a final answer. Examples include Self-consistency (Wang et al., 2022), Best-of-N (Lightman et al., 2023), and Weighted Best-of-N (Li et al., 2022). These methods have gained popularity due to their simplicity and scalability.

[*] Work done during internship at Autodesk. Correspondence to: aminr@cs.toronto.edu.

[2] Code and data available at https://github.com/arakhsha/mob

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

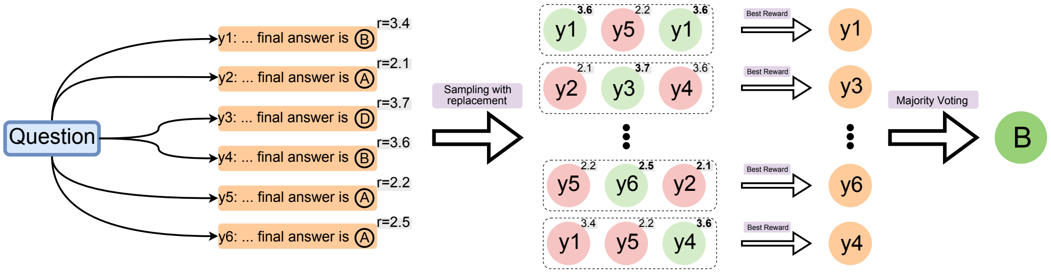
Question
y1: ... final answer is B
r=3.4
y2: ... final answer is A
r=2.1
y3: ... final answer is D
r=3.7
y4: ... final answer is B
r=3.6
y5: ... final answer is A
r=2.2
y6: ... final answer is A
r=2.5
Sampling with replacement
3.6 y1 2.2 y5 3.6 y1
2.1 y2 3.7 y3 3.6 y4
2.2 y5 2.5 y6 2.1 y2
3.4 y1 2.2 y5 3.6 y4
Best Reward
y1
y3
y6
y4
Majority Voting
B

Figure 1: Majority-of-the-Bests: first, N outputs are generated for the given question. Then, we create a large number of subsets of size $m < N$ by sampling with replacement from the generated outputs. From each subset, we choose the output with the highest reward. The most frequent answer among these chosen outputs is reported as the final answer.

Self-consistency (SC) (Wang et al., 2022), also referred to as "majority voting", is a widely used algorithm in this paradigm. It samples multiple outputs from the model and selects the final answer that appears most frequently among them. SC improves the performance by leveraging a key property of the model's output distribution: on difficult problems, the probability of generating the correct answer is often far from 1, making single-sample predictions unreliable. SC capitalizes on the fact that, even if the model's output distribution is imperfect, it may still favor the correct answer and generate it more frequently than incorrect ones.

Best-of-N (BoN) (Lightman et al., 2023) uses a reward model to evaluate the generated outputs and chooses the final answer in the highest-scoring output. With an ideal reward model, BoN succeeds as long as one of the generated outputs is correct. In this paper, we highlight that in the realistic setting of an imperfect reward model, the success of BoN is no longer (nearly) guaranteed. In such cases, BoN exhibits stochastic behavior akin to the underlying generative model. While the reward model improves the likelihood of selecting the correct answer, it often falls short of ensuring certainty. This is the same property that underlies the effectiveness of SC. Motivated by this observation, we show that applying a similar principle—aggregating multiple samples to identify the most probable answer—leads to a better performance over BoN.

We introduce Majority-of-the-Bests (MoB), a method that leverages bootstrapping to improve upon BoN *by approximating the most probable output of BoN.* As illustrated in Figure 1, after obtaining multiple (parallel) solution samples for a given question and computing their rewards, we apply bootstrapping: we create subsets of size m by sampling with replacement from the generated outputs. For each subset, we select the sample with the highest reward. This results in a new set of high-reward samples, over which we perform majority voting to determine the final answer. Just like BoN and SC, MoB can be applied independent of the output generation procedure. It only modifies the selection of the final answer with marginally extra computation on the CPU. We provide a procedure to adaptively select m, eliminating any critical hyperparameters from the algorithm. We show the consistency of the algorithm theoretically, and empirically show significant improvements over BoN on 25 out of 30 tested setups.

2 Background

In this section, we formulate BoN and bootstrapping and provide some background for the algorithm and its theoretical grounds. Given a prompt x, in the standard procedure with LLMs, we sample an output $Y \sim p_{\text{ref}}$ from a base model p_{ref}. This output yields a corresponding final answer $Z = f(Y)$ after applying a post-processing or evaluation function f. For example, for a multiple choice problem, Z is the chosen option and Y is the whole output containing both Z and its justification. We denote the distribution of the final answer in this procedure as π_{ref}, that is, $Z \sim \pi_{\text{ref}}$. The goal is to find the correct final answer z^*. We define the success probability for this given problem as the probability of selecting the correct final answer. If the algorithm's final answer is Z, the success probability is defined as $\mathbb{P}(Z = z^*)$. Given a dataset of questions, the average of the success probabilities over all questions is referred to as the accuracy. For the standard procedure, the success probability is equal to $\pi_{\text{ref}}(z^*)$ and the corresponding accuracy is called the *pass@1* accuracy. We assume access to a

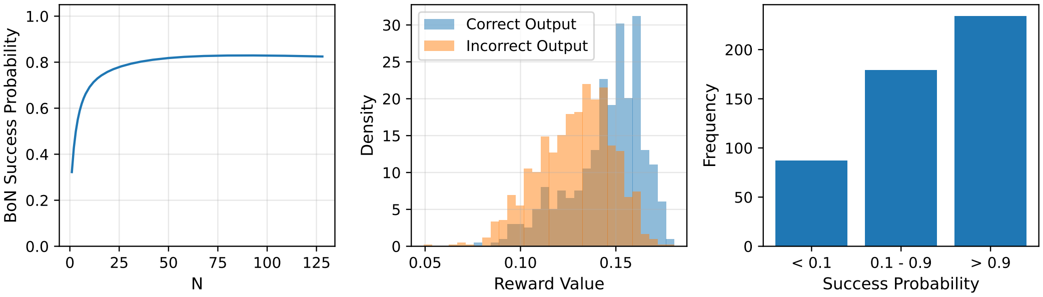

Figure 2: *(Left)* BoN's success probability as a function of N for question 647 from MMLU-Pro-Math. The success probability remains below 80%. *(Middle)* Distribution of the reward for correct and incorrect outputs for the same question. A separation between the two distributions is ideal. *(Right)* Histogram of Best-of-64 success probabilities over 500 questions.

reward model r that assigns a reward $R = r(Y)$ to the output Y, reflecting its accuracy, coherence, or alignment with human preferences (Uesato et al., 2022; Lightman et al., 2023). For a given budget N, sample-and-marginalize algorithms generate N independent outputs $Y_1, \ldots, Y_N \sim p_{\text{ref}}$ and select the final answer reached by one of these outputs. BoN selects the final answer from the output with the highest reward, that is,

$$Z_N^{\text{Best}} = f\big(\underset{y \in \{Y_1, \ldots, Y_N\}}{\text{argmax}} \; r(y)\big).$$

Regularized versions of BoN have also been introduced to address its reward hacking issues in the presence of inaccurate rewards (Jinnai et al., 2024; Ichihara et al., 2025). Alternatively, self-consistency or majority voting selects the final answer that occurs most frequently among $Z_1, \ldots, Z_N$, where $Z_i = f(Y_i)$ is the final answer for output Y_i. If N is large enough, this most frequent answer will be the mode of the final answer distribution π_{ref}. Li et al. (2022) suggested the Weighted Best-of-N (WBoN) selection method. For each final answer, WBoN sums the rewards of all outputs that lead to it. Then, it selects the final answer with the highest total reward.

Bootstrapping. Bootstrapping is a powerful and widely used non-parametric resampling technique for estimating the distribution of a statistic by repeatedly drawing samples with replacement from the original dataset (Efron, 1992; Efron and Tibshirani, 1994). The core idea is to generate multiple "bootstrap samples", by sampling observations uniformly and with replacement. For each bootstrap sample, the statistic of interest is computed. The collection of these computed statistics from the many bootstrap samples forms an empirical approximation of the statistic's true distribution. We use this technique to approximate the distribution of BoN's output.

3 Motivation: Output Distribution of Best-of-N

To motivate our algorithm, we highlight the behavior of BoN's final answer distribution. We denote this distribution by π_N. It means,

$$Z_N^{\text{Best}} \sim \pi_N.$$

Assume among the N sampled outputs, N_c outputs $\{Y_1^c, \ldots, Y_{N_c}^c\} \subseteq \{Y_i\}_{i=1}^N$ yield the correct final answer: $f(Y_i^c) = z^*$. Conversely, $N_w = N - N_c$ outputs $\{Y_1^w, \ldots, Y_{N_w}^w\} \subseteq \{Y_i\}_{i=1}^N$ lead to an incorrect solution. Then, BoN's output is correct if the highest reward among the correct outputs is larger than the highest reward among the incorrect ones. Formally, we can express this condition as:

$$\max\big(r(Y_1^c), \ldots, r(Y_{N_c}^c)\big) > \max\big(r(Y_1^w), \ldots, r(Y_{N_w}^w)\big). \tag{1}$$

There are two factors that influence the probability of this event. First, note that each side of (1) is the maximum of some random variables. As the number of random variables increases, the probability distribution of their maximum shifts towards higher values. Therefore, larger values of N_c and smaller values of N_w, make condition (1) more likely. The values of N_c and N_w depend on $\pi_{\text{ref}}(z^*)$, the probability of the correct answer z^* in the base model's final answer distribution π_{ref}. For large enough n, we will have

$$N_c \approx N \cdot \pi_{\text{ref}}(z^*) \quad , \quad N_w \approx N \cdot (1 - \pi_{\text{ref}}(z^*)).$$

It means that if the base model has a higher chance of solving the problem, BoN is also more likely to select the correct answer.

The second factor is the distribution of $r(Y_i^c)$ and $r(Y_i^w)$ on each side of (1). The reward of a correct output follows the conditional distribution $\mathcal{P}_c \triangleq \mathbb{P}(r(Y)|f(Y) = z^*)$ while the reward of an incorrect output follows the conditional distribution $\mathcal{P}_w \triangleq \mathbb{P}(r(Y)|f(Y) \neq z^*)$. We hope that the reward model assigns higher rewards to correct outputs, and $r(Y_i^c) \sim \mathcal{P}_c$ on the left side of (1) generally be larger than $r(Y_i^w) \sim \mathcal{P}_w$ on the right side.

Therefore, the success probability of BoN heavily depends on the separation between $\mathcal{P}_c$ and $\mathcal{P}_w$. A perfect reward model would always assign a higher value to a correct output than to an incorrect one. In that case, as long as at least one correct output is generated (which is highly likely for large enough N), condition (1) is satisfied. The resulting success probability is close to 1, indicating a nearly deterministic final answer. On the other hand, consider the case where $\mathcal{P}_c$ and $\mathcal{P}_w$ are identical. In this case, the reward of an output becomes independent of its correctness, and choosing according to the reward model will be no better than a random choice. Consequently, the success probability of BoN will be the same as the base model, i.e. $\pi_N(z^*) = \pi_{\text{ref}}(z^*)$. In practice, our reward models exhibit a middle ground between these two extremes. They might not be perfect for BoN to succeed with a single correct output, but they can still be somewhat informative to increase the success probability of BoN compared to the base model.

In Figure 2, we show an example of these dynamics for Question 647 of the MMLU-Pro-Math benchmark (Wang et al., 2024b) with base model Qwen2.5-3B (Qwen Team, 2024) and reward model ArmoRM (Wang et al., 2024a). We approximate the output distribution p_{ref} with a large pool of 1400 samples. In Question 647 (Figure 2), the two distributions $\mathcal{P}_c$ and $\mathcal{P}_w$ are overlapping, and even with large values of N, the success probability remains below 80%. Nonetheless, BoN still outperforms the base model, which is equivalent to Best-of-1 and has a success probability of 30% in this case.

We expect the stochasticity of BoN's output to depend on the difficulty of the question relative to the base and reward models' capabilities. For more difficult questions, the base model generates fewer correct outputs, and the reward model is less likely to distinguish the correct outputs from the incorrect ones. Through the two factors discussed above, BoN is not able to pick the correct answer with high certainty. The right plot in Figure 2 shows the histogram of the success probability of Best-of-64 among 500 randomly selected MMLU-Pro-Math problems. We see that for approximately 175 problems, BoN has a success probability between 0.1 and 0.9. That means, BoN has a significant chance of returning the correct answer but fails to do so reliably. *The idea behind our introduced method, MoB, is that if we can find the most probable output of the BoN distribution, we may reliably pick the correct answer even if its probability is well below* 1.

4 Majority-of-the-Bests

In Section 3, we showed that BoN's final answer is stochastic, and this stochasticity might remain true even with a very large budget N. In this section, we introduce Majority-of-the-Bests (MoB). MoB can select the correct answer with high probability as long as the correct answer is the most probable output of BoN, even if its probability is well below 1. We first showcase this idea in the hypothetical case where BoN's output distribution π_N is given by an oracle. Later, we show how to estimate this distribution using bootstrapping.

4.1 MoB with Oracle Access to BoN's Output Distribution

Suppose the distribution of BoN's final answer π_N is known through an oracle. Instead of sampling from this distribution, which is equivalent to BoN and is a noisy decision, we propose selecting the mode of this distribution. That is

$$z_N^{\text{OracleMoB}} = \operatorname*{argmax}_z \pi_N(z). \tag{1}$$

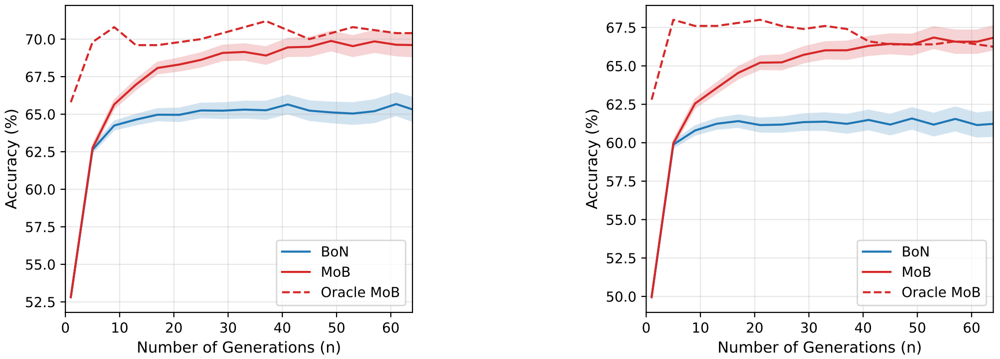

Figure 3: Final answer accuracy comparison of BoN, MoB, and Oracle MoB on MMLU-Pro-Math using Qwen2.5-3B *(Left)* and Llama3.1-8B *(Right)* as the base model, and ArmoRM as the reward model. Results are averaged across all problems and multiple runs. Shaded area indicates the standard error.

We refer to this algorithm as *Oracle MoB* as it relies on an oracle. By selecting the mode, if the correct answer has a higher probability than any of the other answers, it will be selected without any randomness that would reduce the success probability. Since $\pi_{\text{ref}} = \pi_1$, we can say SC for a large N is equivalent to Oracle MoB with $N = 1$. It has been extensively shown that SC improves the LLM's original accuracy. As we will also empirically show, MoB similarly increases the accuracy of BoN by selecting the mode of its output distribution.

In Figure 3, we compare the accuracy of Oracle MoB with BoN on MATH500 (Lightman et al., 2023; Hendrycks et al., 2021) and math problems of MMLU-Pro (Wang et al., 2024b). We use the same output pool, base model, and reward model as Figure 2. We can see that depending on the value of N, Oracle MoB provides 5 to 10 percentage points improvement in accuracy. Oracle MoB unrealistically requires an oracle access to π_N. Next, we will show how π_N can be estimated via bootstrapping and remove the oracle dependence.

4.2 MoB with Estimated BoN's Output Distribution

We now discuss how, without the oracle access to the BoN's output distribution π_N, one can approximately find its most probable output. The most obvious approach is to follow the same procedure as SC. For some $k \geq 1$, we can run k independent BoN procedures, each with m outputs. Then, out of the k resulting answers, we select the final answer that appears the most number of times. The answer of the BoN procedures let us approximate π_m, and selecting the most frequent answer among them will approximate Oracle MoB (1) with budget m. We refer to this algorithm as "BoN+SC" due to its simple combination of BoN and SC. To keep the generation budget fixed at N, we are forced to use a smaller budget m for each of the BoN runs. For now, we treat the choice of m as a hyperparameter, but will return to this choice later. Assume $m < N$ and $k = \lfloor N/m \rfloor$. Formally,

$$Z_m^{\text{Best},(i)} = f\left(\operatorname*{argmax}_{y \in \{Y_{im}, \ldots, Y_{(i+1)m-1}\}} r(y) \right) \qquad (i = 1, \ldots, k), \tag{2}$$

$$Z_{m,n}^{\text{BoN + SC}} = \operatorname*{argmax}_{z} \sum_{i} \mathbb{I}\left[Z_m^{\text{Best},(i)} = z \right]. \tag{3}$$

The main problem with BoN+SC is that it is too expensive. We would like to have a large value for m to get the benefits offered by BoN. To have a fairly accurate estimation of π_m, we need a reasonably large value for k. Together, this requires a large budget $N \approx mk$.

The deficiency of BoN+SC comes from the fact that each sample Y_i only contributes to generating one BoN output. To address this deficiency, we propose estimating π_m not by generating independent samples from it, but by bootstrapping. To do that, we first note that the distribution π_m of Z_m^{Best} is a function of the unknown distribution p_{ref}. Bootstrapping suggests to estimate π_m with the BoN's output distribution under a known approximation $\hat{p}_{\text{ref}} \approx p_{\text{ref}}$. The typical non-parametric approach is

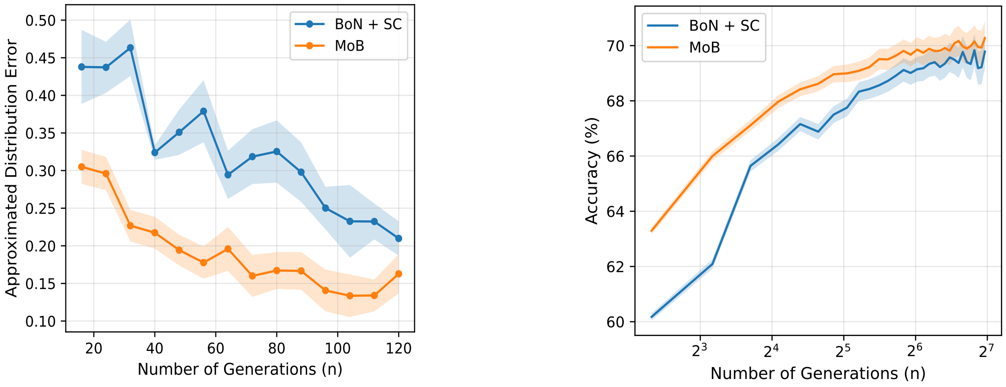

Figure 4: Comparison of MoB and BoN+SC using Qwen2.5-3B as the reference model and ArmoRM as the reward model. *(Left)* π_m approximation error in ℓ_1-norm for $m = 8$. *(Right)* Average accuracy on MMLU-Pro-Math dataset. Shaded area indicates the standard error.

to set $\hat{p}_{\text{ref}}$ to be the empirical distribution of the generated samples $\{Y_1, \ldots, Y_N\}$. Since $\hat{p}_{\text{ref}}$ is known, we can cheaply sample from it. For any arbitrarily large value B, we generate B approximately sampled BoN outputs. We first create B datasets of size m from $\hat{p}_{\text{ref}}$. That is

$$D_i = \{\hat{Y}_{i,1}, \hat{Y}_{i,2}, \ldots, \hat{Y}_{i,m}\} \sim \hat{p}_{\text{ref}}, \qquad\qquad (i = 1, \ldots, B).$$

This is equivalent to sampling m outputs from the original pool $\{Y_1, \ldots, Y_n\}$ with replacement. Then, similar to BoN+SC, we can run BoN on each dataset, and then pick the most common outcome. Formally,

$$\hat{Z}_m^{\text{Best},(i)} = f\left(\operatorname*{argmax}_{y \in D_i} r(y)\right) \qquad\qquad (i = 1, \ldots, B), \tag{4}$$

$$Z_{m,N}^{\text{MoB}} = \operatorname*{argmax}_{z} \sum_{i=1}^{B} \mathbb{I}\left[\hat{Z}_m^{\text{Best},(i)} = z\right]. \tag{5}$$

This procedure is our MoB algorithm for a given m. We define $\hat{\pi}_{m,N}$ to be the (random) distribution of $\hat{Z}_m^{\text{Best},(1)}$ given $\{Y_i\}$ at hand. With sufficiently large B (usually $B = 10,000$ is sufficient), the empirical distribution of $\{\hat{Z}_m^{\text{Best},(i)}\}$ will accurately estimate $\hat{\pi}_{m,N}$. With this approximation, we can write

$$Z_{m,N}^{\text{MoB}} \approx \operatorname*{argmax}_{z} \hat{\pi}_{m,N}(z) \tag{6}$$

Note that this is a light computation that can be carried out on the CPU. Therefore, we can freely choose a large B. In the supplementary material, we provide an even more efficient way of estimating $\hat{\pi}_{m,N}$ with $\mathcal{O}(N \log N)$ complexity that finds $Z_{m,N}^{\text{MoB}}$ directly and without creating B datasets. It is worth mentioning that our use of bootstrap samples resembles bagging (Breiman, 1996) and subagging (Scornet et al., 2015), where a family of models is trained on the subsampled datasets and then aggregated.

In Figure 4, we compare MoB with BoN+SC in the same setup as Figure 3. In the left plot, we fix $m = 8$ and compare the algorithms' error on estimating π_m for a range of values for N. We measure the distance between the two distributions according to the ℓ_1-norm. As we can see, bootstrapping is consistently the superior approach for this approximation task and offers a more accurate estimation of π_m. In the right plot, we set $m = \lfloor\sqrt{N}\rfloor$ and compare the final accuracy of the algorithms. The choice of $m = \lfloor\sqrt{N}\rfloor$ ensures that $k \approx \sqrt{N}$ and will also increase as N increases. We observe that the superior accuracy of bootstrapping in the estimation of π_m translates to a better final accuracy of the algorithm, especially when the budget N is more limited.

One might wonder if it is possible to choose m to be much larger than what was possible in BoN+SC, potentially even $m = N$. There is no obvious limitation on the size of resampled datasets D_i, and nonetheless, most commonly in bootstrapping, the size of resampled datasets is equal to the original

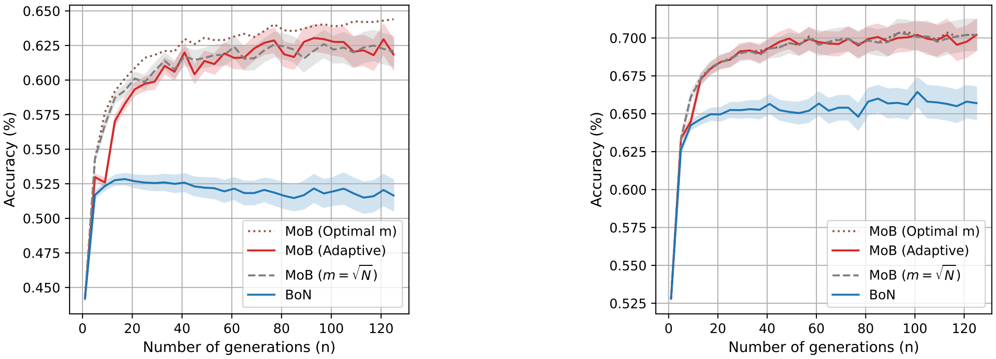

Figure 5: Comparing m selection methods using ArmoRM reward model with MMLU-Pro-Math and Qwen2.5-3B *(Left)* and MATH500 and Llama3.1-8B *(Right)*. Shaded area indicates the standard error.

dataset. However, estimating the distribution of values related to the extremes of random samples is a classic example of failure for the conventional bootstrapping, see for example Athreya and Fukuchi (1994) and Efron and Tibshirani (1994, Section 7.4). Since BoN selects the output with the highest reward, it is affected by the same failure. To see this, note that the output with the highest reward appears in each dataset with the probability of $1 - (\frac{N-1}{N})^m$, and it will be chosen in any dataset in which it appears. Therefore, if $m = N$,

$$\mathbb{P}\Big(\hat{Z}_m^{\text{Best},(i)} = Z_N^{\text{Best}}\Big) \geq 1 - \big(\frac{N-1}{N}\big)^N \approx 1 - e^{-1} \approx 0.632.$$

This means that $\hat{\pi}_{N,N}$ will always incorrectly assign a probability of at least 0.632 to the conventional BoN's answer.

Fortunately, using smaller resampled datasets, as we do in MoB, is one of the remedies for such failures of bootstrapping and is well-studied in the literature(Athreya and Fukuchi, 1994; Bickel et al., 2011). This approach is referred to as m-out-of-n bootstrapping. We show that under the usual conditions of m-out-of-n bootstrapping and mild assumptions on the tail of reward distributions, our use of bootstrapping to estimate π_m is a valid one. Similar to the typical guarantees for bootstrap estimations, we show that our bootstrap estimation is indeed consistent.

Theorem 1. *Under mild assumptions on the distribution of rewards, if there are finite possible values for Z and as $N \to \infty$, we have $m \to \infty$ and $m/N \to 0$, then the estimated $\hat{\pi}_{m,N}$ will converge to the true distribution π_m. That is, for any $\epsilon > 0$,*

$$\lim_{N\to\infty} \mathbb{P}\big(\|\hat{\pi}_{m,N} - \pi_m\|_1 \geq \epsilon\big) = 0.$$

We defer the exact technical statement and proof to the supplementary material. Theorem 1 shows that the estimated distribution $\hat{\pi}_{m,N}$ will match the true BoN output distribution π_m. It means that MoB with bootstrapped distribution in (6) will reach the same accuracy as its oracle version in (1), but with a larger required budget due to $m < N$. To achieve this, it suffices to pick m such that the condition of Theorem 1 holds, which is possible by simply using a fixed schedule of the form $m(n) = n^\alpha$ for some $0 < \alpha < 1$. In the next section, we will discuss the choice of m in more detail and provide a procedure to choose m automatically.

4.3 Adaptive Subsample Size m

The choice of m imposes a trade-off. A larger value of m means that we are running BoN with a larger number of samples. Since we expect the success probability of BoN to increase with more samples, this means that the mode of π_m will be more likely to be correct. On the other hand, as m becomes larger and closer to n, our estimate $\hat{\pi}_{m,N}$ of π_m becomes more inaccurate. As we saw in Section 4.2, bootstrapping might fail to provide a consistent estimate if $m = N$.

Ideally, we would like to find an m such that our final answer $Z_{m,N}^{\text{MoB}}$ based on the estimated distribution as in (6) becomes closest to the Oracle MoB (1) of Section 4.1. The natural approach for this goal is to find the value of m that minimizes the distance between $\hat{\pi}_{m,N}$ and π_N, that is

$$M_N^* = \underset{m}{\operatorname{argmin}} \|\hat{\pi}_{m,N} - \pi_N\|_1. \tag{7}$$

This minimization problem automatically captures both aspects of the trade-off. Large values of m make π_m, which is approximated by $\hat{\pi}_{m,N}$ closer to π_N, but at the same time if m is too large, the error of this approximation becomes too large and increases the objective $\|\hat{\pi}_{m,N} - \pi_N\|_1$.

Unfortunately, the distribution π_N in the objective of (7) is unknown, and therefore cannot be used in practice. The theoretical results by Götze and Račkauskas (2001) show that if Z only takes two possible values and under some other technical conditions, the distance $\|\hat{\pi}_{m,N} - \hat{\pi}_{m/2,N}\|_1$ is proportional to the one in (7)[3]:

$$\|\hat{\pi}_{m,N} - \hat{\pi}_{m/2,N}\|_1 \propto \|\hat{\pi}_{m,N} - \pi_N\|_1.$$

Inspired by this result, Bickel and Sakov (2008) provides some optimality results for choosing m by minimizing the more general loss $\|\hat{\pi}_{m,N} - \hat{\pi}_{qm,N}\|_1$ for some $0 < q < 1$ instead of just $q = 0.5$ considered by Götze and Račkauskas (2001).

Based on the findings of Bickel and Sakov (2008), we propose using the following approach to pick m. We first consider the candidates of the form $\lfloor q^j N \rfloor$ and pick the value among them that minimizes $\|\hat{\pi}_{m,N} - \hat{\pi}_{qm,N}\|_1$.

$$\begin{aligned} m_j &= \lfloor q^j N \rfloor & (j = 0, 1, 2, \ldots), \\ \hat{M}_N^* &= \underset{m=m_j}{\operatorname{argmin}} \|\hat{\pi}_{m_j,N} - \hat{\pi}_{m_{j-1},N}\|_1. \end{aligned}$$

Note that this involves calculating $\hat{\pi}_{m,N}$ for all values of m_j. These will be just $\mathcal{O}(\log N)$ distributions and computationally cheap. Finally, output selected by MoB with adaptive m is $Z_N^{\text{MoB}} = Z_{\hat{M}_N^*,N}^{\text{MoB}}$.

The choice of q has been observed not to be critical in most applications. Bickel and Sakov (2008) observes no significant difference among $q = 0.75, 0.65, 0.6, 0.5$. In our experiments, we fix $q = 0.75$. In Figure 5, we evaluate the efficiency of this procedure to select m. For each N, we measure the highest accuracy achieved by MoB when choosing m from $\{N^\alpha\}$ for $\alpha \in [0.1, 0.9]$. We plot the accuracy of our adaptive m as well as $m = \sqrt{N}$ approach against this optimal performance for two different settings. These figures show both that adaptive m and the simple $m = \sqrt{N}$ achieve performance close to the optimal m variant. This indicates MoB's performance is not sensitive to the choice of m and both the adaptive and simple square root choices achieve a near-optimal performance without hyperparameter tuning. We repeat this comparison in more settings in Appendix E.1.

5 Intuitions and Conditions for Improvement

We now investigate why and when MoB outperforms BoN. To this end, we decompose the algorithmic changes from BoN to MoB into two steps, allowing us to study their individual impact more easily. MoB is the result of applying these two changes to BoN:

1. **Change from Best-of-N to Best-of-m.** To be able to approximate BoN's output distribution, MoB is forced to work with π_m, the output distribution of Best-of-m, for some $m < N$.
2. **Change from Best-of-m to MoB.** The output of Best-of-m is a sample from its output distribution π_m. On the other hand, MoB estimates π_m and selects its mode.

Together, the effects of these two steps determine whether MoB outperforms BoN. The impact of each step varies by question and depends on the base model's generation distribution and the reward model's reward distribution for that question. These distributions—especially the reward distribution—can be complex. To enable an intuitive analysis, we measure two metrics for each question: the base model's success probability and the reward model's accuracy, defined as the fraction of incorrect–correct output pairs in which the correct output receives a higher reward. We

[3] This is a rough interpretation of the results by Götze and Račkauskas (2001), where the ratio of the two losses is studied. We refer the reader to the original paper for more details.

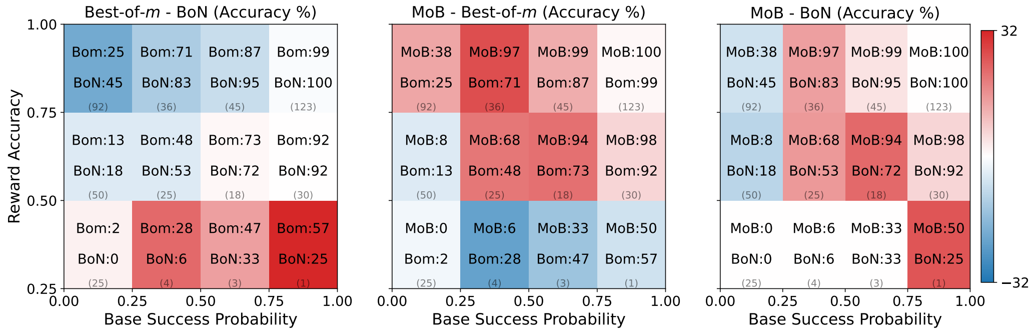

Figure 6: Comparison of Best-of-m (m chosen adaptively) vs. Best-of-N *(Left)*, MoB vs. Best-of-m *(Middle)*, and MoB vs. BoN *(Right)*. Numbers in parentheses are the size of each group.

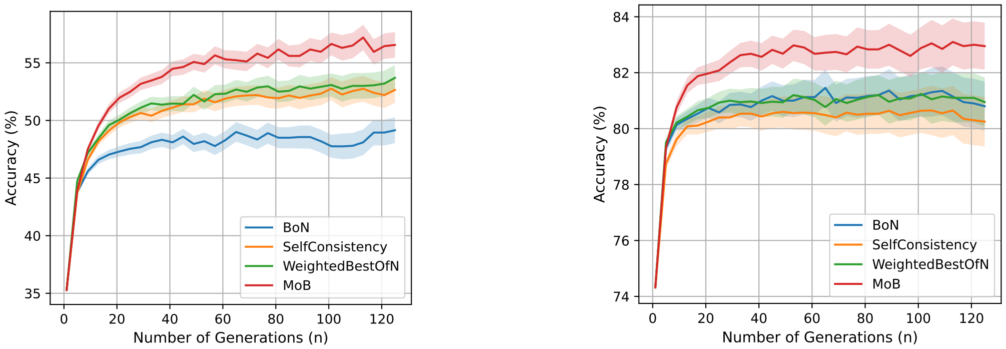

Figure 7: Accuracy comparison on different datasets using base model Qwen2.5-3B and GRM reward model. Standard deviation is shown as the shaded area. *(Left)*: MMLU-Pro-Chem *(Right)*: GSM8k

then categorize questions into 12 groups according to the value of these two metrics and analyze the effects of our changes in each group.

In Figure 6, we compare the accuracy of Best-of-N, Best-of-m, and MoB across the groups for MMLU-Pro-Math benchmark with Gemma2-9B base model, ArmoRM reward model, and $N = 128$. The left plot compares Best-of-N with Best-of-m to show the impact of the first step. In questions with accurate rewards but weak base model performance, BoN benefits the most with more outputs. Therefore, we observe that using m outputs instead of N has the most negative effect on the performance. The performance of Best-of-m is compared to MoB in the middle plot of Figure 6 to measure the impact of the second step. In questions where the mode is correct, MoB will outperform Best-of-m by picking the correct answer with high probability, even if its chance of selection by Best-of-m is low. On the other hand, if the mode is incorrect, MoB will be wrong with high probability, but Best-of-m can still solve the problem by chance. The effect of choosing the mode instead of sampling on the performance depends on the relative number of these two kinds of questions. We expect the mode to be correct more often in groups with accurate enough rewards and base model generations. This is verified by our observation where we see the highest improvement by MoB over Best-of-m for these questions.

Lastly, the right plot in Figure 6 shows the combined effect of the two changes and compares MoB with BoN in each group. MoB improves upon BoN the most in questions where the reward and base models are good enough to make the mode correct, but are not good enough to achieve near perfect accuracy. In Appendix C, we study the success probability of MoB and BoN in a synthetic setup with $N = \infty$ and make similar observations.

6 Experiments

We conducted a series of experiments to compare the performance of our proposed method against other well-known sample-and-marginalize approaches across a range of datasets, generative models, and reward models. The datasets include MATH500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021b), MMLU-Pro (Wang et al., 2024b) questions in math (MMLU-Pro-Math) and chemistry (MMLU-Pro-Chem), and CommonSenseQA (Talmor et al., 2018). We have experimented with three different generative models from different families and different sizes: Qwen2.5-3B-Instruct (Qwen Team, 2024), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Gemma-2-9B-it (Team et al., 2024). For reward models, we used two widely adopted ORMs: ArmoRM (Wang et al., 2024a) and GRM (Yang et al., 2024), with 8B and 3B parameters, respectively. These choices result in thirty diverse experimental setups that rigorously evaluate our method's performance.

Figure 7 presents the accuracy of different methods on GSM8K and MMLU-Pro-Chem across varying values of N. Our method consistently outperforms the baselines, showing clear improvements even at smaller N values. Table 1 presents the accuracy and its standard error for MoB with adaptive m and $m = \sqrt{N}$ alongside SC, BoN, and WBoN for $N = 128$ across all benchmarks, using Qwen and GRM models. Algorithms with statistically insignificant difference to the best algorithm according to a paired one-sided t-test (p-value > 0.05) are also shown in bold. In Table 2, we report the accuracy on MATH500 for all base and reward model combinations. This table also includes a row showing the performance improvement of our method over BoN. As shown in both tables, MoB consistently outperforms BoN in every setting. These results show the potential of MoB as a strong replacement of BoN in tasks with discrete final answers. Complete results for all thirty experiment configurations are provided in Appendix E.2.

Table 1: Results for Qwen2.5-3B and GRM as base and reward models ($N = 128$).

	MATH500	MMLU-Pro-Math	MMLU-Pro-Chem	GSM8K	CSQA
BoN	63.95±1.07	66.10±1.06	49.00±1.12	80.95±0.88	**77.70±0.93**
SC	66.40±1.06	65.60±1.06	52.50±1.12	80.40±0.89	76.20±0.95
WBoN	67.45±1.05	64.35±1.07	53.10±1.12	81.25±0.87	54.90±1.11
MoB-Adaptive (Ours)	**69.95±1.03**	69.30±1.03	**56.45±1.11**	**82.85±0.84**	**77.40±0.94**
MoB-Poly (Ours)	**69.45±1.03**	**70.15±1.02**	**56.30±1.11**	**83.10±0.84**	**77.45±0.93**
↑MoB over BoN	6.00±0.78	3.20±0.79	7.45±0.94	1.90±0.51	-0.30±0.52

Table 2: Results on MATH500 across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	51.55±1.12	52.20±1.12	60.60±1.09	56.65±1.11	54.95±1.11	63.95±1.07
SC	60.65±1.09	52.90±1.12	66.40±1.06	60.65±1.09	52.90±1.12	66.40±1.06
WBoN	**62.90±1.08**	53.85±1.11	67.10±1.05	**63.55±1.08**	56.15±1.11	67.45±1.05
MoB-Adaptive (Ours)	**62.90±1.08**	56.15±1.11	**68.50±1.04**	**64.30±1.07**	57.45±1.11	**69.95±1.03**
MoB-Poly (Ours)	**62.40±1.08**	**57.05±1.11**	67.85±1.04	**64.00±1.07**	**58.10±1.10**	**69.45±1.03**
↑MoB over BoN	11.35±0.86	3.95±0.68	7.90±0.78	7.65±0.80	2.50±0.64	6.00±0.78

7 Conclusion and Future Work

In this paper we highlighted that with imperfect rewards, BoN's chosen answer can be highly stochastic and fail to pick the correct answer reliably. To address this, we introduced Majority-of-the-Bests (MoB), which estimates BoN's output distribution via bootstrapping and chooses the most probable outcome. MoB achieves superior performance compared to other selection algorithms such as, BoN, Self-consistency, and Weighted BoN outperforming them in most of our 30 experimental setups. Our method is scalable, requires no hyperparameter tuning, and adds only negligible CPU computational overhead. MoB can serve as a strong alternative to BoN and SC in problems with discrete final answers. Looking forward, we believe MoB's selection signal could enable early stopping in parallel LLM generation, or be applied more broadly in any framework that relies on sampling from an LLM. However, MoB is limited to settings where the task requires producing a final answer, and like all sampling-based methods, it incurs higher inference costs compared to zero-shot approaches.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers who provided valuable feedback that led to significant improvements to the paper. AMF acknowledges the funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant program (2021-03701).

References

K. B. Athreya and J. Fukuchi. Bootstrapping extremes of I.I.D. random variables. In *Proceedings of the Conference on Extreme Value Theory and Applications, Volume 3*, pages 23–30. National Institute of Standards and Technology (NIST), 1994. 7

Peter J. Bickel and Anat Sakov. On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statistica Sinica*, pages 967–985, 2008. 8

Peter J. Bickel, Friedrich Götze, and Willem R. van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. In *Selected works of Willem van Zwet*, pages 267–297. Springer, 2011. 7, 16

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 6

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. 1

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021a. 18

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9, 2021b. 10

DeepSeek-AI. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. 1

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer, 1992. 3

Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994. 3, 7

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. 19

Friedrich Götze and Alfredas Račkauskas. Adaptive choice of bootstrap sample sizes. *Lecture Notes-Monograph Series*, pages 286–309, 2001. 8

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. 10, 19

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023. 1

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. 5, 18, 19

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, and Eiji Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*, 2025. 3

Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. Regularized best-of-n sampling to mitigate reward hacking for language model alignment. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024. 3

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. 18

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. 1, 3

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. 1, 2, 3, 5, 10, 18

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025. 21

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023. 1

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. 1

OpenAI. Introducing openai o1. https://openai.com/o1/, 2024. OpenAI Blog. 1

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/. 4, 10

Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015. 6

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018. 10

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 18

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 10

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. 3

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. 19

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Conference on Empirical Methods in Natural Language Processing*, 2024a. 4, 10

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 1, 2

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. 4, 5, 10, 18

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural information processing systems*, 35:24824–24837, 2022. 1

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. 1

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*, 2024. 10

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

List of Appendices

We provide a brief description of the material in the appendix of the paper.

- Appendix A provides theoretical results on the asymptotic behavior of BoN's output distribution and the proof for Theorem 1.
- Appendix B provides a closed-form calculation of bootstrapped BoN's output distribution for more efficient calculations.
- Appendix C investigates the effect of reward noise and base model on different algorithms in a synthetic setup.
- Appendix D provides extra details for the experiments and implementations.
- Appendix E provides additional experimental results.

A Theoretical Results

In this section, we provide the formal theoretical results and the proof of Theorem 1. To do so, we first need to show the convergence of BoN's output distribution, which is done in Section A.1 and Theorem 2. We prove Theorem 1 in Section A.2.

A.1 Asymptotic Behavior of BoN's Output Distribution

Theorem 2. *For final answer z such that $\pi_{\textit{ref}}(z) \in (0,1)$, let F_0 and F_1 represent cumulative distribution functions (CDFs) of the conditional distributions $\mathbb{P}(r(Y)|f(Y)=z)$ and $\mathbb{P}(r(Y)|f(Y)\neq z)$, respectively. Define x_0 and x_1 to be their right endpoints,*

$$x_0 \triangleq \sup\{x \in \mathbb{R} : F_0(x) < 1\}, \quad x_1 \triangleq \sup\{x \in \mathbb{R} : F_1(x) < 1\}.$$

As $N \to \infty$,

(i) if $x_0 < x_1$, we have $\pi_N(z) \to 0$.

(ii) if $x_0 > x_1$, we have $\pi_N(z) \to 1$.

(iii) if $x_0 = x_1 = x^*$, F_0 and F_1 are continuous and strictly increasing, and for some $c \in [0,\infty]$,

$$\lim_{x \uparrow x^*} \frac{1-F_0(x)}{1-F_1(x)} = c, \tag{1}$$

then we have,

$$\pi_N(z) \to \frac{c \cdot \pi_{\textit{ref}}(z)}{1+(c-1)\cdot \pi_{\textit{ref}}(z)}.$$

Proof. We first define some random variables to better express $\pi_N(z)$. Assume we use F_0 and F_1 to generate i.i.d. samples $R_1^0, R_2^0, \ldots \overset{\text{i.i.d.}}{\sim} F_0$ and $R_1^1, R_2^1, \ldots \overset{\text{i.i.d.}}{\sim} F_1$. For $n \geq 1$, let S_n^0 and S_n^1 be the maximum of the first n samples from F_0 and F_1, that is,

$$S_n^0 \triangleq \max_{i=1,\ldots,n} R_i^0 \quad , \quad S_n^1 \triangleq \max_{i=1,\ldots,n} R_i^1.$$

Also, for outputs $Y_1, \ldots, Y_N$, let $Z_i = f(Y_i)$, N^0 be the number of outputs that reach the final answer z, and $N^1 = N - N^0$ be the number of outputs that do not reach the final answer z.

We can express $\pi_N(z)$ as

$$\begin{aligned}\pi_N(z) &= \sum_{z_{1:N}} \mathbb{P}\big(Z_N^{\text{Best}} = z | Z_{1:N} = z_{1:N}\big) \cdot \mathbb{P}(Z_{1:N} = z_{1:N}) \\ &= \sum_{z_{1:N}} \mathbb{P}\left(\max_{z_i = z} r(Y_i) > \max_{z_i \neq z} r(Y_i) | Z_{1:N} = z_{1:N}\right) \cdot \mathbb{P}(Z_{1:N} = z_{1:N}). \end{aligned} \tag{2}$$

Now, note that due $Y_1, \dots, Y_N$ being i.i.d., we have

$$\mathbb{P}(r(Y_1), \dots, r(Y_N) | Z_{1:N} = z_{1:N}) = \prod_i \mathbb{P}(r(Y_i) | Z_i = z_i).$$

By definition of R_i^0 and R_i^1, we can therefore write (2) as

$$\pi_N(z) = \sum_{z_{1:N}} \mathbb{P}\big(S_{N^0}^0 > S_{N^1}^1 | Z_{1:N} = z_{1:N}\big) \cdot \mathbb{P}(Z_{1:N} = z_{1:N}) = \mathbb{P}\big(S_{N^0}^0 > S_{N^1}^1\big).$$

For simplicity, we define $S^1 \triangleq S_{N^1}^1$ and $S^0 \triangleq S_{N^0}^0$. Now, we can express $\pi_N(z)$ as

$$\pi_N(z) = \mathbb{P}\big(S^0 > S^1\big).$$

Note that $S^0 \xrightarrow{d} x_0$ and $S^1 \xrightarrow{d} x_1$, which leads to the statement for cases *(i)* and *(ii)*. We focus on case *(iii)*. Let $\bar{F}_0(x) \triangleq 1 - F_0(x)$ and $\bar{F}_1(x) \triangleq 1 - F_1(x)$ be the complementary CDFs of F_0 and F_1, respectively. To quantify $\mathbb{P}\big(S^0 > S^1\big)$, we note that $\bar{F}_1$ is strictly decreasing in a neighborhood of S^1. Thus,

$$\lim_{N \to \infty} \pi_N(z) = \lim_{N \to \infty} \mathbb{P}\big(S^0 > S^1\big) = \lim_{N \to \infty} \mathbb{P}\big(N\bar{F}_1(S^0) < N\bar{F}_1(S^1)\big). \tag{3}$$

Therefore, we turn to study the joint distribution of $(N\bar{F}_1(S^0), N\bar{F}_1(S^1))$ as $N \to \infty$. This will be achieved by quantifying the distribution of $(n_0\bar{F}_1(S_{n_0}^0), n_1\bar{F}_1(S_{n_1}^1))$ as $n_0, n_1 \to \infty$ and relating it to the distribution of $(N\bar{F}_1(S^0), N\bar{F}_1(S^1))$.

Since F_1 is continuous, $F_1(R_i^1) \sim U[0,1]$ is uniformly distributed for any i. Define $U_i = \bar{F}_1(R_i^1) \sim U[0,1]$. It is well known that

$$n_1 \min_{i=1,\dots,n_1} U_i \xrightarrow{d} \text{Exp}(1) \qquad (n_1 \to \infty),$$

which due to $\min_i \bar{F}_1(R_i^1) = \bar{F}_1(S_{n_1}^1)$, translates to

$$n_1\bar{F}_1(S_{n_1}^1) \xrightarrow{d} \text{Exp}(1) \qquad (n_1 \to \infty). \tag{4}$$

Similarly, we can show that $n_0\bar{F}_0(S_{n_0}^0) \xrightarrow{d} \text{Exp}(1)$ as $n_0 \to \infty$. However, our goal is to analyze the distribution of $n_0\bar{F}_1(S_{n_0}^0)$. To do so, we use the tail-equivalence condition (1). We note that $S_{n_0}^0 \xrightarrow{d} x^*$, therefore, $\bar{F}_0(S_{n_0}^0)/\bar{F}_1(S_{n_0}^0) \xrightarrow{d} c$ as $n_0 \to \infty$. Together, we get

$$n_0\bar{F}_1(S_{n_0}^0) = \frac{n_0\bar{F}_0(S_{n_0}^0)}{\bar{F}_0(S_{n_0}^0)/\bar{F}_1(S_{n_0}^0)} \xrightarrow{d} \frac{\text{Exp}(1)}{c} \qquad (n_0 \to \infty). \tag{5}$$

Due to the independence of $S_{n_1}^1$ and $S_{n_0}^0$, we can combine (4) and (5) to get

$$\big(n_0\bar{F}_1(S_{n_0}^0), n_1\bar{F}_1(S_{n_1}^1)\big) \xrightarrow{d} (E/c, F) \qquad (n_0, n_1 \to \infty),$$

where $E, F \overset{\text{i.i.d.}}{\sim} \text{Exp}(1)$. As $N \to \infty$, we have $N^0, N^1 \xrightarrow{p} \infty$, therefore,

$$\big(N^0\bar{F}_1(S_{N^0}^0), N^1\bar{F}_1(S_{N^1}^1)\big) \xrightarrow{d} (E/c, F) \qquad (N \to \infty).$$

Finally, we use the fact that $N^0/N \xrightarrow{d} \pi_{\text{ref}}(z)$ and $N^1/N \xrightarrow{d} 1 - \pi_{\text{ref}}(z)$ to get

$$\big(N\bar{F}_1(S^0), N\bar{F}_1(S^1)\big) = \left(\frac{N^0\bar{F}_1(S_{N^0}^0)}{N^0/N}, \frac{N^1\bar{F}_1(S_{N^1}^1)}{N^1/N}\right) \xrightarrow{d} \left(\frac{E}{c \cdot \pi_{\text{ref}}(z)}, \frac{F}{1 - \pi_{\text{ref}}(z)}\right). \tag{6}$$

Combined with (3), we conclude that

$$\lim_{N \to \infty} \pi_N(z) = \mathbb{P}\left(\frac{E}{c \cdot \pi_{\text{ref}}(z)} < \frac{F}{1 - \pi_{\text{ref}}(z)}\right) = \frac{c\pi_{\text{ref}}(z)}{1 - \pi_{\text{ref}}(z) + c\pi_{\text{ref}}(z)}.$$

□

A.2 Proof of Theorem 1

We restate Theorem 1 with the assumptions not included in the main text.

Theorem 3. *Assume that there are finite possible values for Z and for every possible final answer z, the conditions of Theorem 2 for one the cases hold. If as $N \to \infty$, we have $m \to \infty$ and $m/N \to 0$, then for any $\epsilon > 0$, the estimated $\hat{\pi}_{m,N}$ will converge to the true distribution π_m. That is,*

$$\lim_{n\to\infty} \mathbb{P}\big(\|\hat{\pi}_{m,N} - \pi_m\|_1 \geq \epsilon\big) = 0.$$

Proof. Since there are finite possible values for Z, it suffices to show the convergence in estimated probability of each possible final answer z. We show that for any z, and $\epsilon > 0$, we have

$$\lim_{N\to\infty} \mathbb{P}(|\hat{\pi}_{m,N}(z) - \pi_m(z)| \geq \epsilon) = 0. \tag{7}$$

We use the result by Bickel et al. (2011, Equation 3.14) to show this claim. To do so, we first frame our problem in their notation. For $1 \leq i \leq N$, let $Z_i \triangleq f(Y_i)$ be (the one-hot encoding of) the final answer reached by Y_i, and $R_i \triangleq r(Y_i)$ be the numerical reward of Y_i. We define

$$X_i \triangleq (Z_i, R_i).$$

We define the bootstrap statistic of $X_1, \ldots, X_m$ as

$$T_m = \mathbb{I}\big[Z_m^{\text{Best}} = z\big] + \frac{D}{4} \sim L_m,$$

where $\mathbb{I}[\cdot]$ is the indicator function, $D \sim \text{Bernoulli}(0.5)$ is an independent Bernoulli random variable, and L_m is defined to be the distribution of T_m. Basically, T_m is the indicator of z being selected by BoN, plus a small random noise to ensure the non-degeneracy condition as $m \to \infty$. We define the function $h(t) = \mathbb{I}[t > 0.5]$, so that the parameter of interest θ_m becomes

$$\theta_m \triangleq \mathbb{E}h(T_m) = \pi_m(z),$$

as intended. Lastly, one can verify that since T_m is invariant of repetitions and permutations of its inputs $X_1, \ldots, X_m$, in our case, we have for any $0 < x < 1$,

$$\delta_m(x) \triangleq \big|\pi_{\lfloor mx \rfloor}(z) - \pi_m(z)\big|.$$

We now show the conditions of Bickel et al. (2011, Theorem 2). First, we need to show that L_m, the distribution of T_m, is convergent. According to Theorem 2, we have

$$\lim_{m\to\infty} \pi_m(z) \triangleq \pi_\infty(z)$$

for some $\pi_\infty(z) \in [0, 1]$. Therefore, as $m \to \infty$, we have

$$L_m \xrightarrow{d} \text{Bernoulli}(\pi_\infty(z)) + \frac{\text{Bernoulli}(0.5)}{4}.$$

For condition Bickel et al. (2011, Equation 3.11) we need to show that for any $M < \infty$, we have

$$\delta_m(1 - xm^{-1/2}) \to 0$$

uniformly for all $0 < x < M$. By definition, it suffices to show that for any $0 < x < M$, we have

$$\big|\pi_{\lfloor m - x\sqrt{m} \rfloor}(z) - \pi_m(z)\big| \to 0.$$

This follows from the fact that $\pi_m(z)$ is convergent to $\pi_\infty(z)$. For any $\varepsilon > 0$, pick M_0 such that for any $m_0 \geq M_0$, we have

$$|\pi_{m_0}(z) - \pi_\infty(z)| < \frac{\varepsilon}{2},$$

and M_1 such that for any $M_1 - M\sqrt{M_1} \geq M_0$. Then for any $m \geq M_1$, we have

$$\big|\pi_{\lfloor m - x\sqrt{m} \rfloor}(z) - \pi_\infty(z)\big| < \varepsilon/2 \quad \text{and} \quad |\pi_m(z) - \pi_\infty(z)| < \varepsilon/2.$$

Together, we have

$$\big|\pi_{\lfloor m - x\sqrt{m} \rfloor}(z) - \pi_m(z)\big| < \varepsilon$$

and achieve the uniform convergence condition.

Finally, note that our statistic T_m is not dependent on the sampling distribution p_{ref} and Bickel et al. (2011, Equation 3.13) is satisfied. □

B Closed-Form Calculation of Bootstrapped BoN's Output Distribution

In Section 4.2, we proposed approximating $\hat{\pi}_{m,N}$ by running BoN on a large number B of subsets of size m sampled with replacement from the N generated outputs. In practice, $B = 10,000$ is commonly considered sufficient. This calculation is negligible compared to the generation of outputs from the LLM and can be carried out on a CPU. Nonetheless, we here show that it can also be done in $\mathcal{O}(N \log N)$.

Define $R_i = r(Y_i)$ for $1 \leq i \leq N$, and let $i_1, i_2, \ldots, i_N$ be such that

$$R_{i_1} < R_{i_2} < \ldots < R_{i_N}.$$

For simplicity, we assume no ties occur among the rewards. The key insight is that for any $1 \leq k \leq N$, the probability of Y_{i_k} being selected in a randomly sampled subset of m outputs can be calculated in closed-form. We note that Y_{i_k} is selected if the subset only includes outputs among $Y_{i_1}, \ldots, Y_{i_k}$, but is not limited to $Y_{i_1}, \ldots, Y_{i_{k-1}}$ (and therefore contains Y_{i_k}). We get

$$\mathbb{P}(Y_{i_k} \text{ is the output of BoN on a resampled subset}) = \left(\frac{k}{N}\right)^m - \left(\frac{k-1}{N}\right)^m.$$

Thus, for any final answer z, the probability of it being selected in a subset is

$$\hat{\pi}_{m,N}(z) = \sum_{k: Z_{i_k} = z} \left(\frac{k}{N}\right)^m - \left(\frac{k-1}{N}\right)^m.$$

This procedure only requires sorting the outputs according to their rewards and therefore has complexity of $\mathcal{O}(N \log N)$.

C Effect of Reward Noise and Base Model's Success Probability

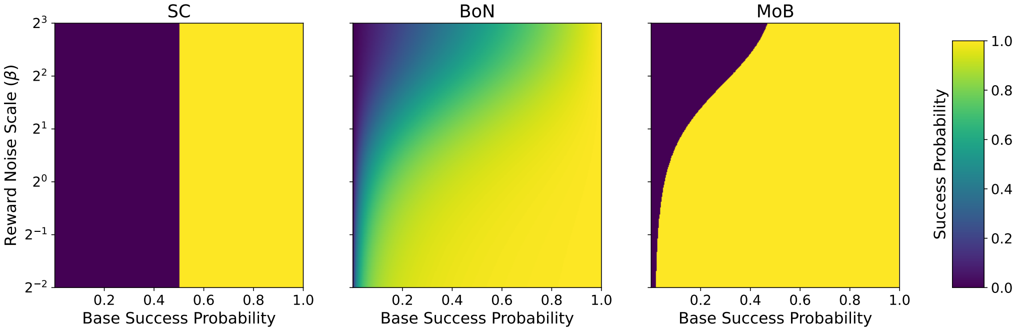

Figure 8: Success probability of SC, BoN, and MoB with infinite budget ($N = \infty$) for different values of the base model's success probability and reward noise.

In this section, we investigate the effect of the base model and reward noise on the success probability of SC, BoN, and MoB. We consider a synthetic setup for a TRUE/FALSE question, where the correct answer is TRUE. Let p be the success probability of the base model, which is the probability that the base model generates a solution reaching the correct final answer.

Assume r^{oracle} is an oracle reward model that always assigns the reward of 1 to solutions that reach the correct answer, and 0 otherwise:

$$r^{\text{oracle}}(Y) = \begin{cases} 1, & \text{if } f(Y) = \text{TRUE}, \\ 0, & \text{if } f(Y) = \text{FALSE}. \end{cases}$$

To investigate the effect of an imperfect reward model, we consider a noisy reward model r^{noisy} that is equal to the oracle reward plus an exponentially distributed noise:

$$r^{\text{noisy}}(Y) = r^{\text{oracle}}(Y) + \text{Exp}(1/\beta).$$

The parameter β controls the noise level, where a larger β indicates a noisier reward model. To see this, note that the expected value and the standard deviation of the noise are equal to β. If β is large, the noise will dominate the signal from the oracle reward, and the noisy reward model will be less informative.

We visualize the success probability of SC, BoN, and MoB with infinite budget $N = \infty$ in Figure 8. SC's success probability, as shown in the left plot of Figure 8, is independent of the reward noise. It is either equal to 1 when $p > 0.5$ (the correct answer is the most probable answer), or equal to 0 otherwise. For BoN, consider two extreme cases for the reward noise. When the reward model is perfect (β small), BoN's success probability is 1 regardless of the base model's success probability. This is shown in the bottom edge of the middle plot in Figure 8. In this case, BoN is preferable over SC. On the other hand, when the reward model is completely uninformative (β large), BoN's success probability is equal to the base model's success probability. This is shown in the top edge of the middle plot in Figure 8. MoB's success probability is equal to 1 if BoN's success probability is at least 0.5, as shown in the right plot of Figure 8. We see that MoB shows a similar behavior to SC when the reward model is uninformative, and when the reward model is perfect, MoB behaves like BoN.

In this setup, we can study the success probability of BoN and MoB with an infinite budget $N = \infty$ theoretically. BoN's success probability depends on the reward's noise level. It can be calculated from Theorem 2 as

$$\text{BoN success probability with infinite budget} = \frac{e^{1/\beta} p}{1 - p + e^{1/\beta} p}.$$

Note that if the reward model is perfect ($\beta = 0$), both the numerator and denominator go to infinity, and we reach the success probability of 1. With $\beta = \infty$, the noise becomes dominant, and BoN's success probability remains equal to the base model p even with infinite budget. Due to Theorem 1, MoB solves the problem if the correct answer is BoN's most probable outcome. Therefore,

$$\text{MoB success probability with infinite budget} = \begin{cases} 1, & \text{if } \frac{e^{1/\beta} p}{1-p+e^{1/\beta} p} > 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

This is favorable over BoN in scenarios where BoN still prefers the correct answer, as it can find the correct answer reliably without randomness.

D Implementation and Experiment Details

In this section, we provide more details on how the experiments in the paper are conducted.

D.1 Evaluation Experiments

Benchmarks. We run our experiments on five popular benchmarks. MATH500, first introduced by Lightman et al. (2023), is a randomly sampled subset of 500 math questions with short final answers from the MATH dataset (Hendrycks et al., 2021). We use the math and chemistry questions from the MMLU-Pro benchmark (Wang et al., 2024b), which includes multiple-choice questions on a variety of topics. We also run our experiments on GSM8K (Cobbe et al., 2021a) that contains grade school math questions in short final answer format. Lastly, we use the CommonsenseQA benchmark (Talmor et al., 2019) that tests the model's commonsense reasoning through multiple-choice questions. For all benchmarks, we randomly select 500 questions for our experiments.

Base and Reward Models. We have used the models `Qwen/Qwen2.5-3B-Instruct`, `google/gemma-2-9b-it`, and `meta-llama/Llama-3.1-8B-Instruct` from Huggingface as base generative models. Our reward models are `Ray2333/GRM-Llama3.2-3B-rewardmodel-ft` and `RLHFlow/ArmoRM-Llama3-8B-v0.1`, which are the best performing 3B and 8B reward models according to Rewardbench (Lambert et al., 2024) in reasoning tasks.

Implementation Details. In the implementation of MoB, we always use the closed-form calculation of $\hat{\pi}_{m,N}$ discussed in Appendix B to efficiently perform the bootstrap estimate. Therefore, in the actual implementation, there is no parameter B and we effectively operate as if $B = \infty$ was chosen.

We use Huggingface's Python library for all the output generations. The generation was carried on H100 GPUs. The compute cost was not tracked, but we estimate it to be on the order of a few thousand GPU-hours. We always use temperature 1 for inference and no extra modification of the next-token sampling procedure. The final answer extraction and evaluation are calculated using the Language Model Evaluation Harness (Gao et al., 2024). For each question, we generate 512 outputs and for each budget size N, we run each algorithm $\lfloor 512/N \rfloor$ times. Reported standard errors for the accuracies are calculated with the assumption of normal errors and from the standard deviation of $500 \times \lfloor 512/N \rfloor$ independent runs of the algorithm (500 is the dataset size across all benchmarks). The numbers reported as the improvement of MoB over BoN are based on the adaptive MoB, and its standard error is calculated from the standard deviation of paired differences of the algorithms score in $500 \times \lfloor 512/N \rfloor$ runs. We use the Scipy library (Virtanen et al., 2020) in python to conduct one-sided paired t-test to decide statistical significance of the difference between the best performing algorithm with another algorithm in our tables. Algorithms with insignificant (p-value > 0.05) difference are also shown in bold.

For GSM8K, we use a 5-shot prompt. For MATH and MMLU-Pro questions, we use the zero-shot chain-of-thought prompting used in the official Llama3.1 models evaluation (Grattafiori et al., 2024) on MATH (Hendrycks et al., 2021). This prompt and the prompt used for CommonsenseQA are given in the following.

Prompt for MATH and MMLU-Pro

```
Solve the following <topic> 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.
Problem: <problem from dataset>
```

Prompt for CommonsenseQA

```
Use commonsense to solve the following multiple choice question. First
explain your solution and then give the final answer. Always finish
your answer with "the answer is (X)" where X is the correct letter
choice. Question:: <problem from dataset>
```

D.2 Details of Other Experiments

In Figure 2, we discussed the success probability of BoN, which requires an estimate of BoN's output distribution. We use the same technique as in MoB to estimate this output distribution. To minimize the error of this approximation, we specifically generate 1,400 outputs for the math problems in MMLU-Pro with Qwen2.5-3B . Then, we use $\hat{\pi}_{N,1400}$, as defined in Section 4.2 as an estimate for π_N. Same technique is used in Figure 3 where the mode of $\hat{\pi}_{N,1400}$ is chosen as the output of oracle MoB, and Figure 4 to where the distribution estimation error is calculated with respect to $\hat{\pi}_{m,1400}$ instead of the true π_m.

In Figure 5, we consider seven fixed schedules for m, specifically $m = \lfloor N^\alpha \rfloor$ for $\alpha = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$. At any budget N, we compared the accuracy of MoB with adaptive m against the highest accuracy among the seven instantiations of fixed schedule MoB.

In Figure 6, for each question, we measure base model's success probability and reward model's accuracy using 512 outputs. We ignore questions with all-correct or all-incorrect outputs, since the reward accuracy is not defined for them, as well as questions with reward accuracy bellow 0.25 due to all algorithms having zero success probability on them. Also, m is calculated adaptively for each question.

E Additional Experimental Results

In this section, we provide additional experimental results for all 30 setups.

E.1 Adaptive Subset Size Selection

In Section 4, we compared MoB with adaptive choice of m and $m = \sqrt{N}$ with the optimal choice of m. We provide this comparison in MATH500 (Figure 9), MMLU-Pro-Math (Figure 10), MMLU-Pro-Chem (Figure 11), GSM8K (Figure 12), and CommonsenseQA (Figure 13). In Table 3, we compare the performance of MoB with adaptive q for various values of q on all benchmarks Llama3.1-8B base model and ArmoRM reward model. As also observed in the literature, we observe that the choice of q is not a sensitive one.

Table 3: Performance of MoB with adaptive m across different choices of q for Llama3.1-8B base model and ArmoRM reward model.

q	0.40	0.50	0.60	0.70	0.80	0.90
MATH500	61.85%	63.00%	63.15%	62.20%	62.45%	60.70%
MMLU-Pro-Math	66.60%	66.85%	66.75%	66.60%	67.10%	66.35%
MMLU-Pro-Chem	56.75%	57.40%	57.85%	57.15%	56.95%	56.35%
GSM8k	91.60%	91.55%	91.55%	91.85%	91.80%	91.85%
CSQA	77.45%	77.40%	77.30%	77.25%	77.35%	77.30%

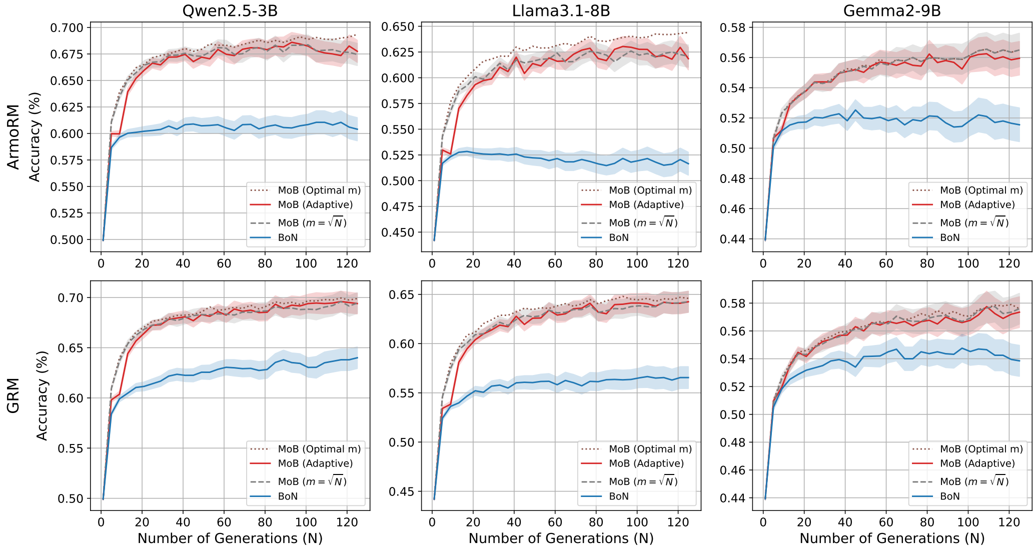

Figure 9: Comparison of MoB with adaptive m and $m = \sqrt{N}$ against MoB with optimal m on the MATH500 dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

E.2 Evaluation Experiments

We compare MoB with adaptive m and $m = \sqrt{N}$ with baselines in MATH500 (Figure 14, Table 4), MMLU-Pro-Math (Figure 15, Table 5), MMLU-Pro-Chem (Figure 16, Table 6), GSM8K (Figure 17, Table 7), and CommonsenseQA (Figure 18, Table 8).

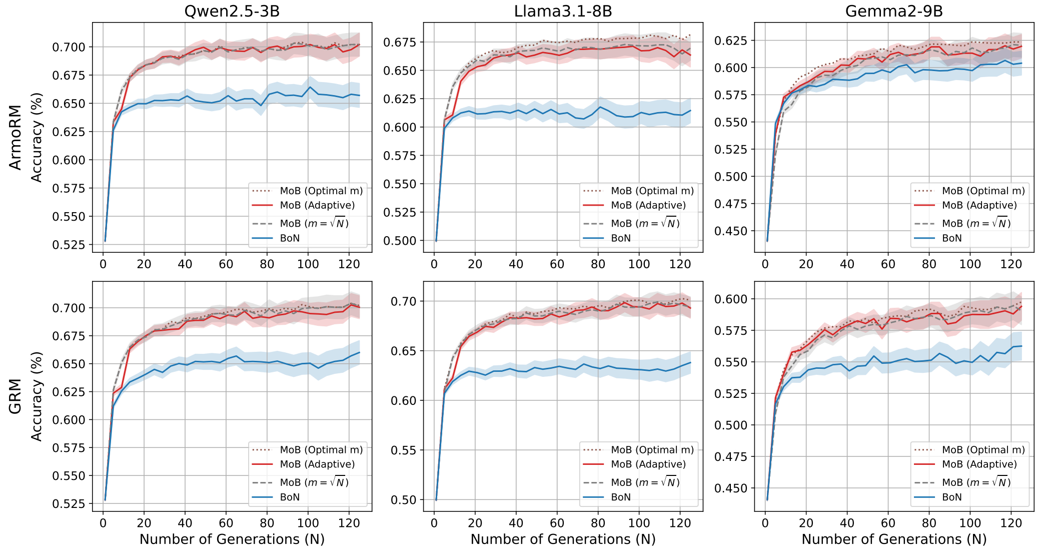

Figure 10: Comparison of MoB with adaptive m and $m = \sqrt{N}$ against MoB with optimal m on the MMLU-Pro-Math dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

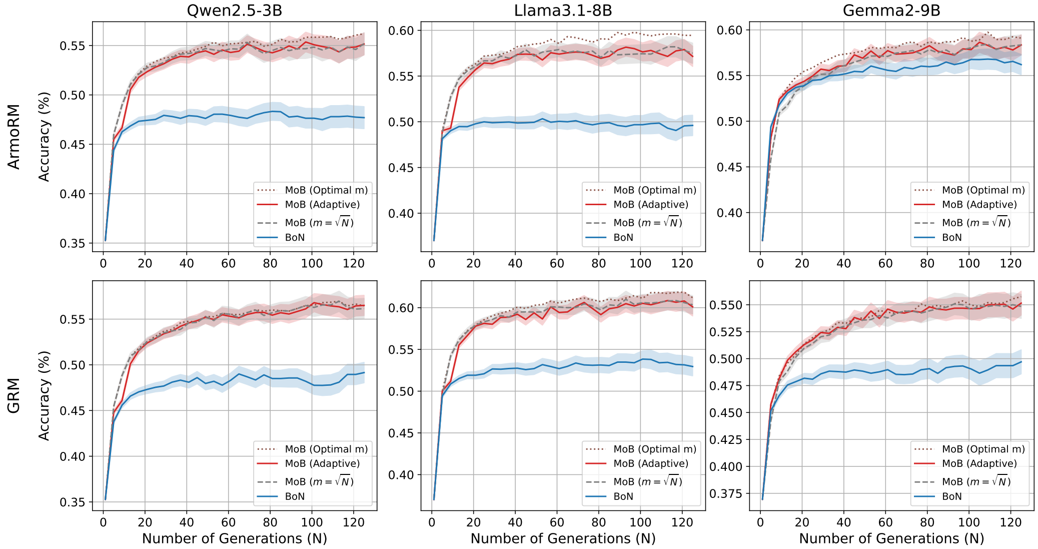

Figure 11: Comparison of MoB with adaptive m and $m = \sqrt{N}$ against MoB with optimal m on the MMLU-Pro-Chem dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

E.3 Results on Skywork-v2 Reward Model

Additionally, we report the results for `Skywork/Skywork-Reward-V2-Llama-3.1-8B` (Liu et al., 2025), a more recent reward model in Tables 9-13.

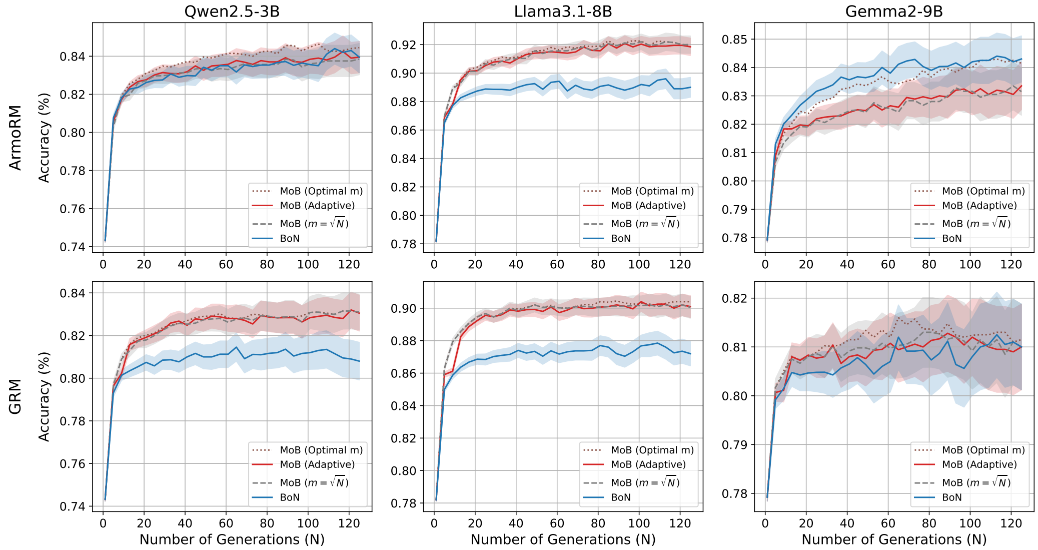

Figure 12: Comparison of MoB with adaptive m and $m = \sqrt{N}$ against MoB with optimal m on the GSM8K dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

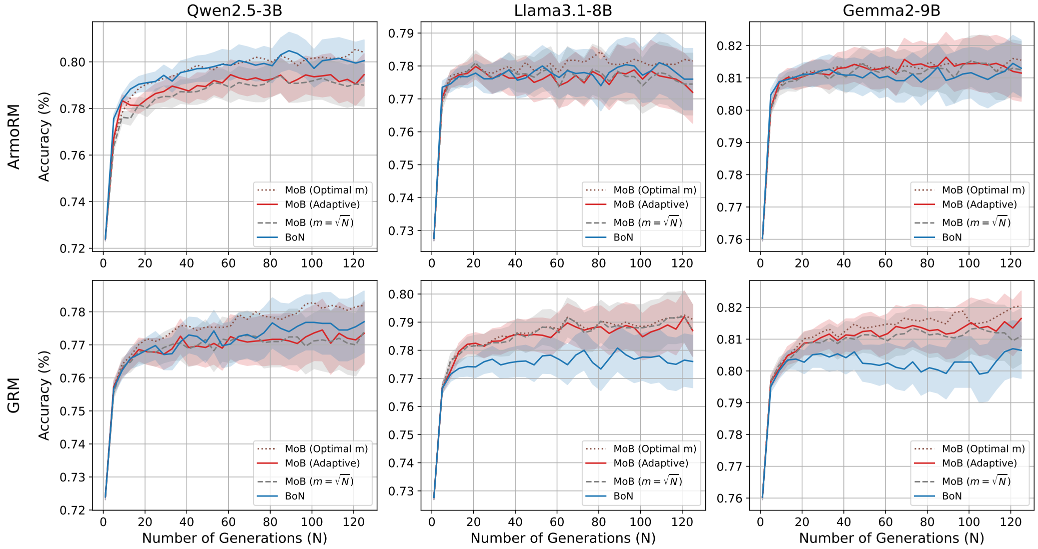

Figure 13: Comparison of MoB with adaptive m and $m = \sqrt{N}$ against MoB with optimal m on the CommonsenseQA dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

Table 4: Results on MATH500 across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	51.55±1.12	52.20±1.12	60.60±1.09	56.65±1.11	54.95±1.11	63.95±1.07
SC	60.65±1.09	52.90±1.12	66.40±1.06	60.65±1.09	52.90±1.12	66.40±1.06
WBoN	**62.90±1.08**	53.85±1.11	67.10±1.05	**63.55±1.08**	56.15±1.11	67.45±1.05
MoB-Adaptive (Ours)	**62.90±1.08**	56.15±1.11	**68.50±1.04**	**64.30±1.07**	57.45±1.11	**69.95±1.03**
MoB-Poly (Ours)	**62.40±1.08**	**57.05±1.11**	67.85±1.04	**64.00±1.07**	**58.10±1.10**	**69.45±1.03**
↑MoB over BoN	11.35±0.86	3.95±0.68	7.90±0.78	7.65±0.80	2.50±0.64	6.00±0.78

Table 5: Results on MMLU-Pro-Math across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	61.40$_{\pm1.09}$	60.45$_{\pm1.09}$	65.95$_{\pm1.06}$	64.10$_{\pm1.07}$	56.15$_{\pm1.11}$	66.10$_{\pm1.06}$
SC	62.95$_{\pm1.08}$	49.95$_{\pm1.12}$	65.60$_{\pm1.06}$	62.95$_{\pm1.08}$	49.95$_{\pm1.12}$	65.60$_{\pm1.06}$
WBoN	**66.45$_{\pm1.06}$**	52.25$_{\pm1.12}$	66.70$_{\pm1.05}$	60.05$_{\pm1.10}$	56.45$_{\pm1.11}$	64.35$_{\pm1.07}$
MoB-Adaptive (Ours)	**66.70$_{\pm1.05}$**	**61.55$_{\pm1.09}$**	**69.80$_{\pm1.03}$**	**69.05$_{\pm1.03}$**	**59.35$_{\pm1.10}$**	69.30$_{\pm1.03}$
MoB-Poly (Ours)	**67.20$_{\pm1.05}$**	**62.05$_{\pm1.09}$**	**70.05$_{\pm1.02}$**	**69.30$_{\pm1.03}$**	**59.45$_{\pm1.10}$**	**70.15$_{\pm1.02}$**
↑MoB over BoN	5.30$_{\pm0.81}$	1.10$_{\pm0.71}$	3.85$_{\pm0.80}$	4.95$_{\pm0.82}$	3.20$_{\pm0.83}$	3.20$_{\pm0.79}$

Table 6: Results on MMLU-Pro-Chem across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	49.70$_{\pm1.12}$	56.60$_{\pm1.11}$	48.05$_{\pm1.12}$	53.05$_{\pm1.12}$	49.25$_{\pm1.12}$	49.00$_{\pm1.12}$
SC	50.25$_{\pm1.12}$	43.40$_{\pm1.11}$	52.50$_{\pm1.12}$	50.25$_{\pm1.12}$	43.40$_{\pm1.11}$	52.50$_{\pm1.12}$
WBoN	**57.65$_{\pm1.10}$**	45.45$_{\pm1.11}$	53.30$_{\pm1.12}$	49.75$_{\pm1.12}$	**57.25$_{\pm1.11}$**	53.10$_{\pm1.12}$
MoB-Adaptive (Ours)	**57.40$_{\pm1.11}$**	58.05$_{\pm1.10}$	**54.75$_{\pm1.11}$**	**60.75$_{\pm1.09}$**	54.60$_{\pm1.11}$	**56.45$_{\pm1.11}$**
MoB-Poly (Ours)	**57.80$_{\pm1.10}$**	**58.80$_{\pm1.10}$**	**54.90$_{\pm1.11}$**	60.00$_{\pm1.10}$	55.00$_{\pm1.11}$	**56.30$_{\pm1.11}$**
↑MoB over BoN	7.70$_{\pm0.92}$	1.45$_{\pm0.80}$	6.70$_{\pm0.92}$	7.70$_{\pm0.93}$	5.35$_{\pm0.92}$	7.45$_{\pm0.94}$

Table 7: Results on GSM8K across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	89.00$_{\pm0.70}$	**84.20$_{\pm0.82}$**	**83.85$_{\pm0.82}$**	87.15$_{\pm0.75}$	**81.20$_{\pm0.87}$**	80.95$_{\pm0.88}$
SC	88.15$_{\pm0.72}$	80.55$_{\pm0.89}$	80.40$_{\pm0.89}$	88.15$_{\pm0.72}$	**80.55$_{\pm0.89}$**	80.40$_{\pm0.89}$
WBoN	88.70$_{\pm0.71}$	80.75$_{\pm0.88}$	81.10$_{\pm0.88}$	77.75$_{\pm0.93}$	79.45$_{\pm0.90}$	81.25$_{\pm0.87}$
MoB-Adaptive (Ours)	**91.75$_{\pm0.62}$**	83.30$_{\pm0.83}$	**83.85$_{\pm0.82}$**	**90.50$_{\pm0.66}$**	**81.15$_{\pm0.87}$**	**82.85$_{\pm0.84}$**
MoB-Poly (Ours)	**91.80$_{\pm0.61}$**	83.15$_{\pm0.84}$	**83.80$_{\pm0.82}$**	90.05$_{\pm0.67}$	**80.85$_{\pm0.88}$**	**83.10$_{\pm0.84}$**
↑MoB over BoN	2.75$_{\pm0.56}$	-0.90$_{\pm0.52}$	0.00$_{\pm0.50}$	3.35$_{\pm0.56}$	-0.05$_{\pm0.47}$	1.90$_{\pm0.51}$

Table 8: Results on CSQA across all base and reward models ($N = 128$).

	ArmoRM			**GRM**		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	**77.80$_{\pm0.93}$**	**81.20$_{\pm0.87}$**	**80.15$_{\pm0.89}$**	**78.05$_{\pm0.93}$**	80.55$_{\pm0.89}$	**77.70$_{\pm0.93}$**
SC	75.75$_{\pm0.96}$	79.25$_{\pm0.91}$	76.20$_{\pm0.95}$	75.75$_{\pm0.96}$	79.25$_{\pm0.91}$	76.20$_{\pm0.95}$
WBoN	76.75$_{\pm0.94}$	80.05$_{\pm0.89}$	76.60$_{\pm0.95}$	36.35$_{\pm1.08}$	49.80$_{\pm1.12}$	54.90$_{\pm1.11}$
MoB-Adaptive (Ours)	**77.40$_{\pm0.94}$**	**81.20$_{\pm0.87}$**	**79.40$_{\pm0.90}$**	**78.45$_{\pm0.92}$**	**81.45$_{\pm0.87}$**	**77.40$_{\pm0.94}$**
MoB-Poly (Ours)	**77.30$_{\pm0.94}$**	**81.45$_{\pm0.87}$**	79.15$_{\pm0.91}$	**78.65$_{\pm0.92}$**	**81.15$_{\pm0.87}$**	**77.45$_{\pm0.93}$**
↑MoB over BoN	-0.40$_{\pm0.47}$	0.00$_{\pm0.43}$	-0.75$_{\pm0.48}$	0.40$_{\pm0.54}$	0.90$_{\pm0.48}$	-0.30$_{\pm0.52}$

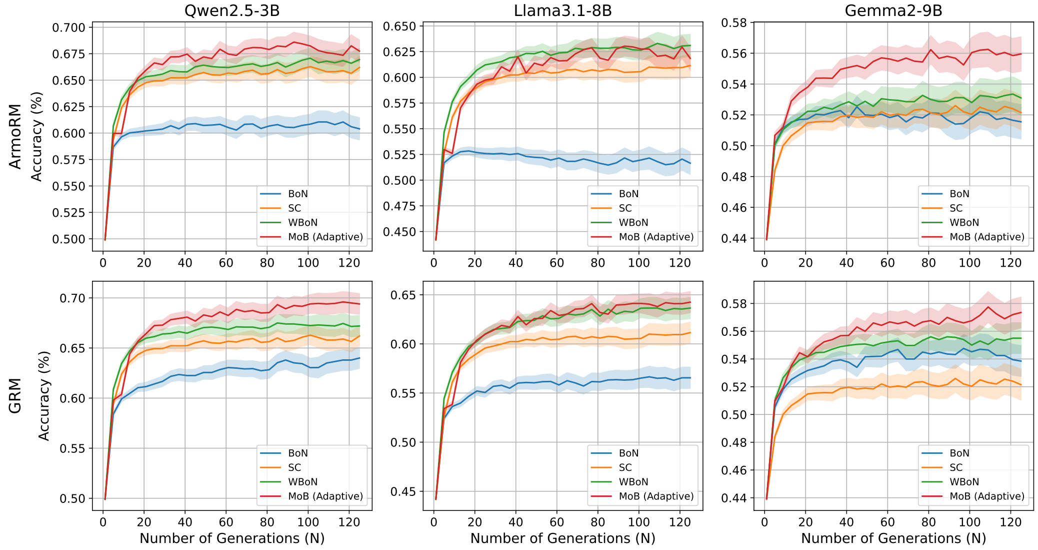

Figure 14: Comparison of MoB with the baselines on the MATH500 dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

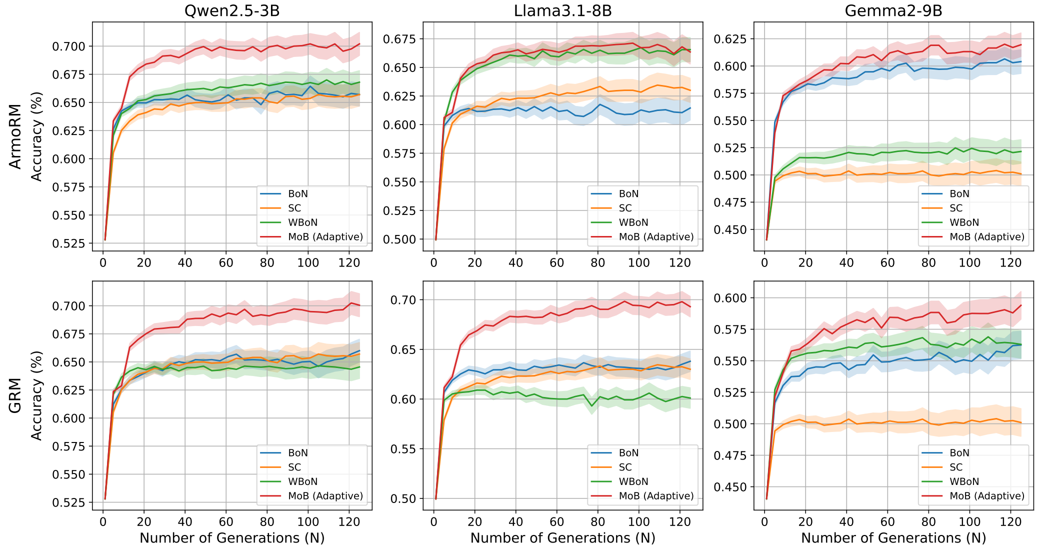

Figure 15: Comparison of MoB with the baselines on the MMLU-Pro-Math dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

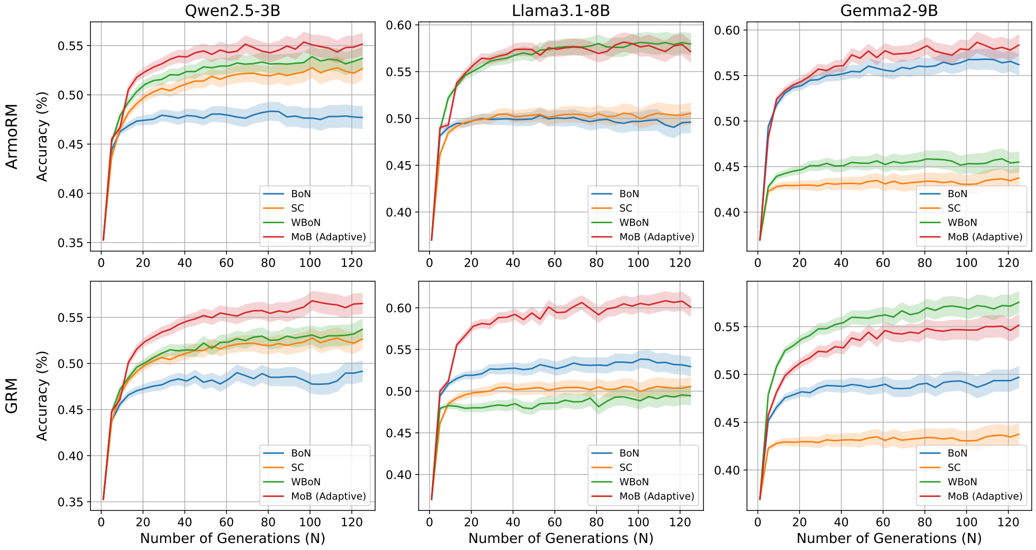

Figure 16: Comparison of MoB with the baselines on the MMLU-Pro-Chem dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

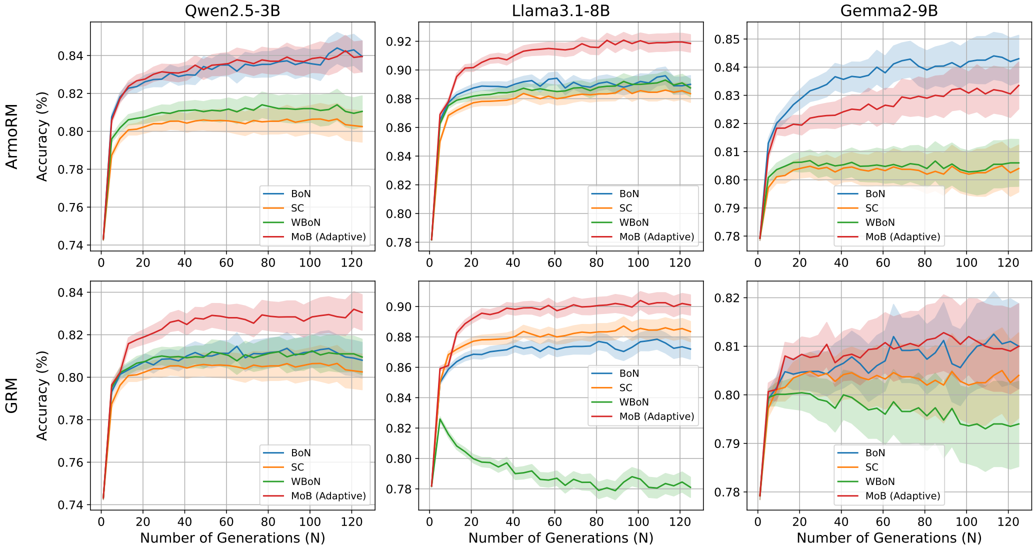

Figure 17: Comparison of MoB with the baselines on the GSM8K dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

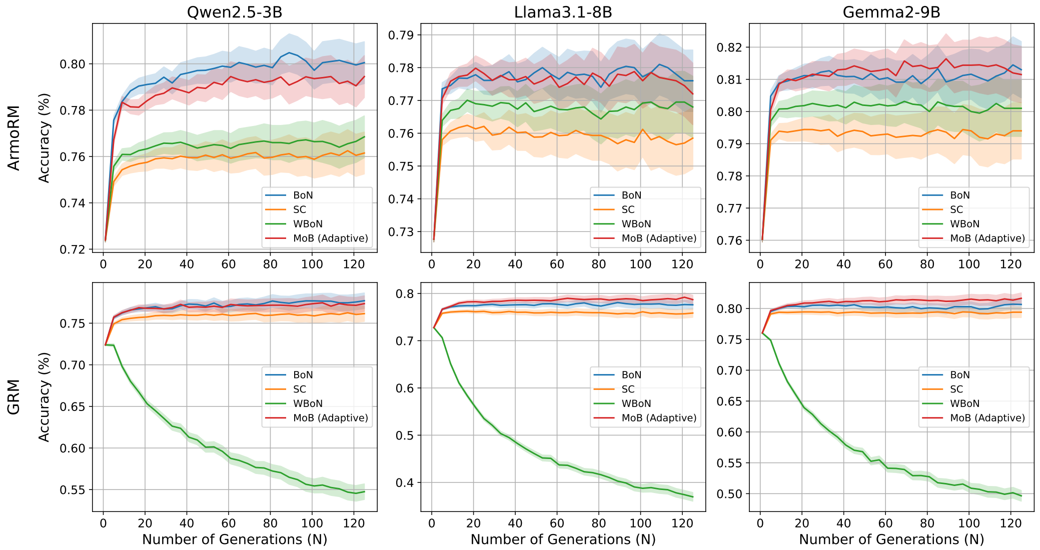

Figure 18: Comparison of MoB with the baselines on the CommonsenseQA dataset with ArmoRM *(Up)* and GRM *(Down)* reward models, and Qwen2.5-3B *(Left)*, Llama3.1-8B *(Middle)*, and Gemma2-9B *(Right)* base models. Shaded areas show standard error.

Table 9: Results on MATH500 with Skywork reward model across all base models ($N = 128$).

	Skywork		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	54.50±0.79	55.85±0.79	63.35±0.76
SC	60.65±1.09	52.90±1.12	66.40±1.06
WBoN	**65.05±1.07**	57.95±1.10	68.80±1.04
MoB-Adaptive (Ours)	63.95±1.07	**59.45±1.10**	69.70±1.03
MoB-Poly (Ours)	63.65±1.08	**59.15±1.10**	**70.35±1.02**
↑MoB over BoN	9.45±0.81	3.60±0.72	6.35±0.78

Table 10: Results on MMLU-Pro-Math with Skywork reward model across all base models ($N = 128$).

	Skywork		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	60.10±0.77	54.20±0.79	67.80±0.74
SC	62.95±1.08	49.95±1.12	65.60±1.06
WBoN	**69.45±1.03**	**60.00±1.10**	69.80±1.03
MoB-Adaptive (Ours)	66.00±1.06	**59.10±1.10**	**72.55±1.00**
MoB-Poly (Ours)	66.70±1.05	**59.10±1.10**	**72.85±0.99**
↑MoB over BoN	5.90±0.75	4.90±0.88	4.75±0.75

Table 11: Results on MMLU-Pro-Chem with Skywork reward model across all base models ($N = 128$).

	Skywork		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	57.23±0.82	53.20±0.79	57.70±0.78
SC	50.60±1.17	43.40±1.11	52.50±1.12
WBoN	**62.83±1.13**	**58.60±1.10**	58.65±1.10
MoB-Adaptive (Ours)	60.87±1.14	**57.75±1.10**	**61.50±1.09**
MoB-Poly (Ours)	60.92±1.14	**57.40±1.11**	**61.55±1.09**
↑MoB over BoN	3.64±0.79	4.55±0.98	3.80±0.82

Table 12: Results on GSM8K with Skywork reward model across all base models ($N = 128$).

	Skywork		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	85.15±0.56	**80.98±0.62**	82.37±1.05
SC	88.15±0.72	80.53±0.89	80.85±1.53
WBoN	88.25±0.72	80.13±0.89	82.07±1.50
MoB-Adaptive (Ours)	**89.55±0.68**	**81.23±0.87**	**84.04±1.43**
MoB-Poly (Ours)	**89.85±0.68**	**81.23±0.87**	**83.89±1.43**
↑MoB over BoN	4.40±0.61	0.25±0.46	1.67±0.82

Table 13: Results on CSQA with Skywork reward model across all base models ($N = 128$).

	Skywork		
	Llama3.1-8B	Gemma2-9B	Qwen2.5-3B
BoN	77.00±0.67	80.15±0.63	**78.85±0.65**
SC	75.75±0.96	79.25±0.91	76.20±0.95
WBoN	76.45±0.95	**80.20±0.89**	76.15±0.95
MoB-Adaptive (Ours)	**77.80±0.93**	**81.00±0.88**	77.00±0.94
MoB-Poly (Ours)	**78.00±0.93**	**81.00±0.88**	77.25±0.94
↑MoB over BoN	0.80±0.43	0.85±0.41	-1.85±0.49

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 results supporting the claims are reported in the Method and Experiments section.

   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: Mentioned the limitations in the Conclusions and Future Work section.

   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: To the best of our knowledge, all theorems are correct, and every mathematical citation is properly used.

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: All the implementation details and the choice of model/dataset/benchmark is provided in the main text and the appendix.

   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: All of the data that we have used is already public. We will publish the code upon acceptance.

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: Our model has only one hyperparameter which is automatically tuned by our method. The generative models and datasets are all well-known and widely used in the community.

   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 provided confidence intervals in our tables and figures.

   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: Yes, we will provide as many quantifiable and trackable information as possible to address these questions.

   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: Yes we have followed the guidelines mentioned on the website during the course of the project.

   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: Our method is a general technique for improving inference in generative models. The societal impacts depend heavily on the specific use cases and deployment contexts of these models. Therefore, a detailed discussion of societal impacts falls outside the scope of this work.

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: [Yes]

    Justification: We are not releasing any new dataset or scraped datasets. Every model and dataset used in this paper is already open-source and public.

    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: [No]

    Justification: We have provided proper citation for every model/dataset that we have used. The reader can find the licensing of those assets in those references, but we have not mentioned those licenses in our manuscript.

    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: The code of our method will be public. We are not releasing any new asset (e.g. model/dataset) except our method's code.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in our research does not involve LLMs as any important, original, or non-standard components.

    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.