

KEEP THE BEAM ON TRACK: STABILIZING REWARD TRAJECTORIES IN GUIDED DECODING

Anonymous authors

Paper under double-blind review

ABSTRACT

Decoding algorithms play a central role in enhancing the performance of large language models (LLMs) on complex reasoning tasks. A common approach incorporates Process Reward Models (PRMs), which estimate the quality of intermediate reasoning paths and guide the selection of possible continuations. In this setting, our analysis reveals two notable phenomena: reward estimates tend to decline as reasoning progresses, and the reasoning paths exhibit distinct *volatility patterns* across decoding steps depending on whether the paths lead to correct or incorrect final answers. In particular, correct reasoning tends to be associated with stable reward trajectories, while incorrect reasoning often shows high volatility. Motivated by this observation, we propose Volatility-Scaled Guided Decoding (VSGD), a decoding algorithm that prioritizes candidate paths with lower volatility by jointly considering the magnitude of PRM-estimated rewards and the volatility of these rewards across decoding steps. Experiments on datasets including GSM8K and MATH500 indicate that VSGD reduces the volatility of selected reward trajectories and improves the accuracy of the final answer. These findings suggest that considering the temporal dynamics of reward values, in addition to their magnitude, provides a potential direction for enhancing guided decoding in LLMs.

1 INTRODUCTION

Large language models (LLMs) achieve strong performance across a range of reasoning tasks, including mathematical problem solving (Vaswani et al., 2017; Brown et al., 2020; Wei et al., 2022; Kojima et al., 2022; Touvron et al., 2023; Achiam et al., 2023; Team et al., 2024; Grattafiori et al., 2024). Early studies suggest that carefully designed prompts, such as Chain-of-Thought, can elicit the reasoning capability of LLMs (Wei et al., 2022; Kojima et al., 2022). Beyond prompting, recent work highlights that the decoding algorithm is another crucial factor, as the decoding algorithm governs token selection and influences the reasoning path. (Shi et al., 2024). Among decoding strategies, tree-based methods such as beam search are prominent because the tree-based methods explore multiple candidate reasoning paths in parallel, thereby increasing the likelihood of uncovering promising reasoning paths before generating the final answer (Graves, 2012; Sutskever et al., 2014; Bahdanau et al., 2014; Wu et al., 2016; Kool et al., 2019; Leblond et al., 2021; Meister et al., 2021; Yang et al., 2024). Building on this idea, recent approaches propose guided decoding methods that incorporate additional signals to prioritize promising candidates. These signals range from rule-based constraints (Lu et al., 2021; Welleck et al., 2022) to model-based scoring functions, including reward models that assess the quality of generated sequences (He et al., 2017; Uesato et al., 2022; Krishna et al., 2022; Lightman et al., 2023; Liu et al., 2024; Snell et al., 2024; Wang et al., 2025).

A growing body of work incorporates external reward models into decoding algorithms to provide target-oriented guidance during text generation. Conventional reward models guide generation by evaluating only the final output, which restricts guidance to the end of the generation process. Process Reward Models (PRMs) address this limitation by assigning rewards to intermediate reasoning steps, thereby enabling step-level guidance of incomplete sequences during decoding (Chan et al., 2019; Uesato et al., 2022; Lee et al., 2023; Lightman et al., 2023; Wang et al., 2024; Setlur et al., 2024; Zeng et al., 2025). Guided decoding algorithms leverage these PRM-estimated scores to prioritize incomplete sequences that are more likely to yield high-quality final answers (Chen et al., 2024; Snell et al., 2024; Zhang et al., 2025b; Hu et al., 2025). However, existing approaches primarily emphasize the magnitude of individual reward values, overlooking how these values evolve across successive

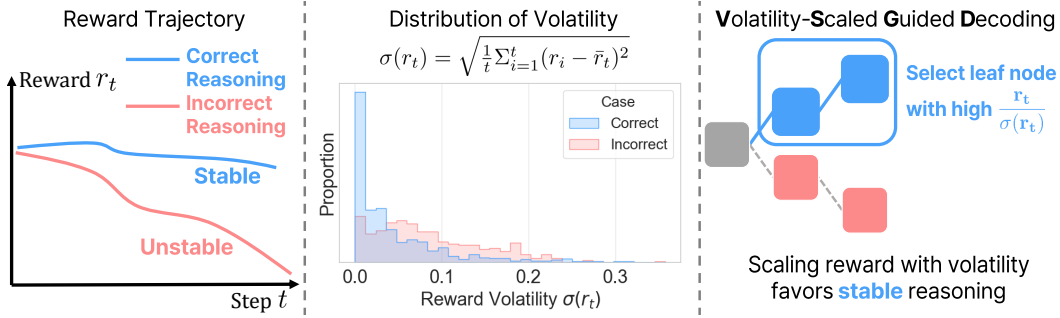


Figure 1: Our empirical observations on the reward trajectory, which motivated our proposed method VSGD. **(Left)** Reward trajectories r_t of process reward model (PRM)-based decoding methods show that correct reasoning paths typically maintain stable and low-volatility rewards, whereas incorrect paths exhibit unstable and high-volatility patterns. **(Middle)** We quantify volatility $\sigma(r_t)$ as a measure of how much rewards fluctuate across decoding steps. Histogram of volatility, computed over 500 samples from MATH500 dataset, shows that correct paths consistently exhibit lower volatility than incorrect paths. **(Right)** Motivated by this finding, we propose Volatility-Scaled Guided Decoding (VSGD), which scales rewards by volatility to favor candidate paths with stable reward trajectories.

reasoning steps. We refer to the sequence of reward scores assigned to each intermediate reasoning step as the *reward trajectory*, denoted by r_t . We use the term *volatility* $\sigma(r_t)$ to describe how stable the reward trajectory is across steps. Formally, we measure volatility as the root mean square deviation of the stepwise rewards from their mean. As illustrated in Fig. 1, our empirical analysis suggests an association between volatility and correctness: in observed cases, correct reasoning paths tend to maintain relatively stable reward trajectories, while incorrect reasoning paths show unstable reward trajectories. To obtain these trajectories, we use 500 samples from the MATH500 dataset, following the approach used in Lightman et al. (2023); Wang et al. (2024), which selects candidate continuations based on the minimum PRM-estimated reward across decoding steps. Motivated by this observation, we propose to adjust reward estimates using volatility, so that paths with both high rewards and stable trajectories are given higher priority during decoding.

In this paper, we propose the **Volatility-Scaled Guided Decoding** algorithm (VSGD), a decoding algorithm that integrates the volatility of reward trajectories into the decoding process. We define volatility as the root mean square deviation of reward values across decoding steps, capturing the stability of a reasoning path. Leveraging the volatility measure, VSGD scales reward signals to favor trajectories with more stable rewards, which empirically correlate with correct solutions. Our contributions are as follows:

- We empirically analyze reward trajectories and demonstrate that the volatility, which measures the variability of reward values, exhibits distinct patterns between correct and incorrect reasoning paths.
- Motivated by this observation, we propose Volatility-Scaled Guided Decoding (VSGD), a decoding algorithm that rescales reward values using the volatility of reward trajectory to prioritize stable reasoning paths.
- Experiments on datasets including GSM8K and MATH500, show that VSGD improves reasoning performance by favoring low-volatility reward trajectories. In addition, we show that VSGD maintains superior performance under constrained settings where the number of reasoning steps or the number of complete reasoning paths is limited.

2 RELATED WORK

Guided Decoding Algorithms. Decoding algorithms are strategies that decide which word or token a language model should generate next, thereby controlling the overall process of text generation. (Graves, 2012; Sutskever et al., 2014; Bahdanau et al., 2014; Wu et al., 2016; Kool et al., 2019; Josifoski et al., 2023). Control over text generation enables additional signals to guide the selection of candidate continuations. The guidance can take the form of rule-based constraints that restrict token selection to satisfy logical requirements (Lu et al., 2021; Welleck et al., 2022), or model-based scoring functions that assign values to incomplete sequences and thereby guide the

search toward preferred continuations (He et al., 2017; Uesato et al., 2022; Krishna et al., 2022; Lightman et al., 2023). Recent studies show that incorporating additional signals into decoding can enhance LLM reasoning by directly affecting the decoding process (Liu et al., 2024; Snell et al., 2024). Tree-based algorithms are particularly suited for guided decoding because tree-based algorithms maintain multiple candidate continuations, allowing additional signals to be applied when ranking or pruning alternatives before the final answer is generated (Yao et al., 2023; Feng et al., 2023; Chen et al., 2024; Yang et al., 2024). Among these, one variant is Monte Carlo Tree Search, which guides decoding by simulating possible continuations (Chaffin et al., 2022; Li et al., 2023a; Feng et al., 2023; Hao et al., 2023). However, the reliance on repeated simulations results in high computational cost (Chaffin et al., 2022; Hao et al., 2023; Liu et al., 2023). By contrast, beam search provides a more favorable trade-off between effectiveness and efficiency, as beam search retains a tractable number of candidate sequences, supporting straightforward integration of external scoring functions (Josifoski et al., 2023; Chen et al., 2024), making it a widely used algorithm for guided decoding.

Process Reward Models. While guided decoding typically relies on external signals applied to entire sequences, recent studies incorporate additional signals into decoding algorithms to provide guidance during text generation. Process Reward Models (PRMs) extend this paradigm by assigning rewards to intermediate reasoning steps rather than only to final outputs, thereby enabling step-level evaluation of incomplete sequences (Chan et al., 2019; Uesato et al., 2022; Lee et al., 2023; Lightman et al., 2023; Wang et al., 2024; Setlur et al., 2024; Zeng et al., 2025). Guided decoding algorithms exploit these PRM-estimated scores by aggregating the scores across steps to prioritize sequences that are more likely to generate correct outputs (Chen et al., 2024; Snell et al., 2024; Zhang et al., 2025b; Hu et al., 2025). Nonetheless, most existing approaches emphasize the magnitude of reward values while overlooking the temporal dynamics of reward trajectories.

Instability of Reward Trajectories. Instability in reward signals, often called volatility, refers to how much the reward values fluctuate along a reasoning path. In reinforcement learning, Bisi et al. (2019) formalizes volatility as the deviation of cumulative rewards from their average across the entire trajectory. Although this formulation offers a principled measure of instability, it requires access to complete trajectories and, to our knowledge, has not been considered in the context of guided decoding. More recently, Zhang et al. (2025c) introduce *reasoning volatility*, which evaluates the discrepancy between an intermediate reasoning step and the final answer. Concurrently, Zhang et al. (2025a) investigates reward distributions when training PRMs for mathematical reasoning, which reports that reasoning paths leading to correct final answers tend to maintain higher rewards compared to incorrect reasoning paths when tested on the training dataset. In contrast, our study analyzes reward trajectories during the decoding process and introduces *reward volatility* as a metric to quantify the instability of reward trajectories, highlighting the relation between output correctness and reward trajectories during decoding process.

3 VSGD: VOLATILITY-SCALED GUIDED DECODING ALGORITHM

In this section, we formally describe our proposed algorithm dubbed as VSGD. To begin with, Sec. 3.1 defines the notation and formalizes volatility as a measure of reward instability. Sec. 3.2 presents our observations that reveal the relationship between volatility and the correctness of reasoning paths. Motivated by these observations, in Sec. 3.3 we propose Volatility-Scaled Guided Decoding algorithm (VSGD), which prioritizes reasoning paths characterized by stable reward trajectories.

3.1 NOTATION

Let x_0 denote an input prompt, and let x_t represent the tokens generated at decoding step t . A reasoning path up to step t is denoted as $\mathbf{x}_t = [x_1, \dots, x_t]$. Let f denote a process reward model (PRM) that maps a reasoning path $\mathbf{x}_t$ to a scalar score $f(\mathbf{x}_t) \in [0, 1]$, where higher values indicate higher-quality reasoning. The step-level reward at step t is defined as $r_t := f(\mathbf{x}_t)$, and the reward trajectory up to step t is denoted as $\mathbf{r}_t = [r_1, \dots, r_t]$. We formalize the *volatility* of the reward trajectory $\mathbf{r}_t$ as

$$\sigma(\mathbf{r}_t) := \sqrt{\frac{1}{t} \sum_{i=1}^t (r_i - \bar{r}_t)^2}, \quad (1)$$

where $\bar{r}_t = \frac{1}{t} \sum_{i=1}^t r_i$ is the reward averaged over the reasoning step t . This formulation corresponds to the standard deviation of a time series, a measure widely used across domains including

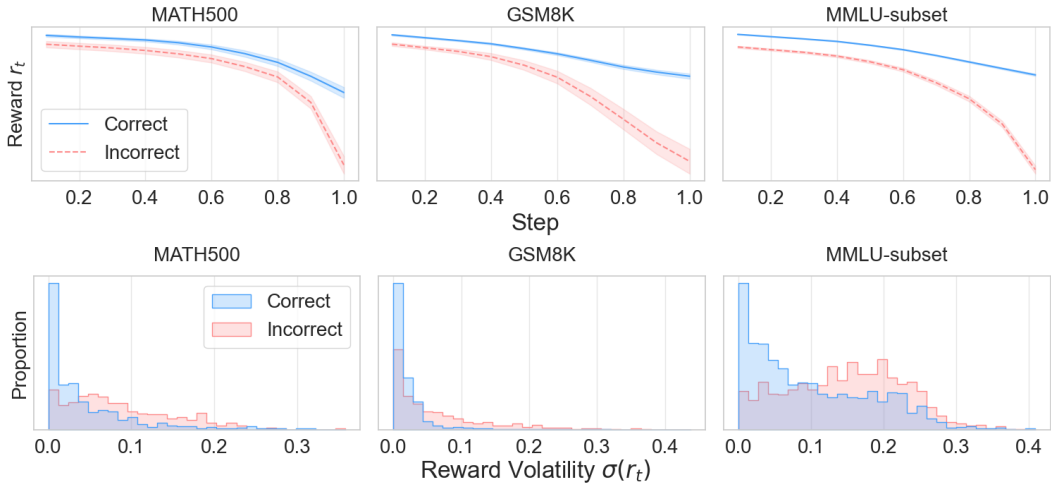


Figure 2: Dynamics of correct and incorrect reasoning paths under PRM-guided decoding as established in a prior work (Zeng et al., 2025) on MATH500, GSM8K and MMLU-subset. Correct paths are shown in blue and incorrect paths in red. **(Top)** Reward trajectories with standard error shading show that correct paths maintain higher rewards throughout decoding, while incorrect paths decline. **(Bottom)** Volatility histograms (Eq. 1) indicate that correct paths generally exhibit lower volatility than incorrect paths across benchmarks.

finance (Markowitz, 1952; Daly, 2008; 2011; Alan & Kyre, 2019), physiology (Brockmann & Hunt, 2023), and reinforcement learning (Bisi et al., 2019). In this work, we extend this notion to quantify the variability of PRM-estimated rewards across decoding steps. Notably, our formulation differs from that of Zhang et al. (2025c), who define *reasoning volatility* as the semantic discrepancy between an intermediate reasoning step and the final answer. In contrast, we define *reward volatility* as the variability of PRM-estimated rewards within the decoding process.

3.2 OBSERVATION ON REWARD TRAJECTORY

Here we present empirical results showing how conventional PRM guidance shapes decoding dynamics. Specifically, we compare the reward trajectories of two cases: (i) correct reasoning paths that lead to a correct answer, and (ii) incorrect paths that end with a wrong answer. We evaluate on three datasets: two mathematical reasoning benchmarks – GSM8K (Cobbe et al., 2021) and MATH500 (Lightman et al., 2023) – and a subset of MMLU dataset (Hendrycks et al., 2021). For the latter dataset, we focus on six categories (computer science, chemistry, econometrics, formal logic, philosophy, and virology) in MMLU, and refer to it as the “MMLU-subset” throughout the paper. To analyze reward trajectories, we apply a PRM-based guided decoding approach, following standard configurations from prior work (Zeng et al., 2025). Since the number of steps in reward trajectories varies across samples, we rescale each one to the unit interval $[0,1]$. We then use linear interpolation to map rewards onto this common scale, allowing us to compare trajectories point by point.

Fig. 2 shows the reward trajectory r_t and its volatility $\sigma(r_t)$ defined in Eq. 1, to quantify the instability of reward trajectories. As shown in the top row of Fig. 2, reasoning paths leading to correct final answers consistently maintain higher rewards, whereas reasoning paths ending in incorrect answers exhibit declines, indicating systematic differences in reward trajectories by correctness. The bottom row of Fig. 2 shows that correct reasoning paths generally have lower volatility than incorrect paths¹. For each dataset, we assess statistical significance using the Mann–Whitney U test (Mann & Whitney, 1947), which is a non-parametric method for testing whether two distributions differ. For GSM8K, MATH500 and all categories of MMLU-subset, the differences are significant ($p < 0.05$), confirming

¹Concurrently, Zhang et al. (2025a) observe that when training PRMs, correct reasoning paths in the training dataset exhibit higher reward values than incorrect reasoning paths. In contrast, we demonstrate that correct reasoning paths not only maintain higher reward values during decoding but also exhibit distinct *volatility* patterns, which we quantify explicitly.

Algorithm 1 RescaleWithVolatility**Input:** Reward trajectory $\mathbf{r}_t = [r_1, \dots, r_t]$, stability constant ϵ **Output:** Rescaled reward trajectory $\tilde{\mathbf{r}}_t = [\tilde{r}_1, \dots, \tilde{r}_t]$

```

1:  $\bar{r} \leftarrow \frac{1}{t} \sum_{i=1}^t r_i$ 
2:  $\sigma \leftarrow \sqrt{\frac{1}{t} \sum_{i=1}^t (r_i - \bar{r})^2}$ 
3:  $\tilde{\mathbf{r}}_t \leftarrow [r_i / (\sigma + \epsilon)]_{i=1}^t$ 
4: return  $\tilde{\mathbf{r}}_t$ 

```

Algorithm 2 Volatility-Scaled Guided Decoding (VSGD)**Input:** PRM f , prompt $\mathbf{x}_0$, expansion width M , beam size N , maximum step T , maximum number of complete reasoning paths L , stability constant ϵ , reward trajectory aggregation function Agg **Output:** Complete reasoning path list $\mathcal{X}_{\text{comp}}$ **Constraint:** N is divisible by M

```

1:  $\mathcal{X}_0 \leftarrow [(\mathbf{x}_0, []) \mid i = 1, \dots, N/M]$   $\triangleright$  List of (reasoning path, reward trajectory) pairs
2:  $\mathcal{X}_{\text{comp}} \leftarrow []$   $\triangleright$  List of complete reasoning paths
3: for  $t = 1$  to  $T$  do
4:    $\mathcal{C}_t, \mathcal{S}_t \leftarrow [], []$   $\triangleright$  Lists of incomplete reasoning paths and their aggregated scores
5:   for all  $(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}) \in \mathcal{X}_{t-1}$  do
6:     for all  $\mathbf{x}_t \in \text{Expand}(\mathbf{x}_{t-1}, M)$  do
7:       if  $\mathbf{x}_t$  ends with  $\langle \text{eos} \rangle$  then
8:          $\mathcal{X}_{\text{comp}} \leftarrow \mathcal{X}_{\text{comp}} \cdot \text{append}(\mathbf{x}_t)$   $\triangleright$  Store the complete reasoning path
9:         if  $|\mathcal{X}_{\text{comp}}| \geq L$  then
10:          return  $\mathcal{X}_{\text{comp}}$ 
11:        end if
12:      continue
13:    end if
14:     $\mathbf{r}_t \leftarrow \mathbf{r}_{t-1} \cdot \text{append}(f(\mathbf{x}_t))$ 
15:     $\tilde{\mathbf{r}}_t \leftarrow \text{RescaleWithVolatility}(\mathbf{r}_t, \epsilon)$ 
16:     $\mathcal{C}_t \leftarrow \mathcal{C}_t \cdot \text{append}((\mathbf{x}_t, \mathbf{r}_t))$ 
17:     $\mathcal{S}_t \leftarrow \mathcal{S}_t \cdot \text{append}(\text{Agg}(\tilde{\mathbf{r}}_t))$ 
18:  end for
19: end for
20:  $\text{best\_beam\_indices} \leftarrow \text{Argsort}(\mathcal{S}_t, \frac{N}{M})$ 
21:  $\mathcal{X}_t \leftarrow \mathcal{C}_t[\text{best\_beam\_indices}]$ 
22: end for
23: return  $\mathcal{X}_{\text{comp}}$ 

```

that volatility distributions for correct and incorrect reasoning paths differ significantly; the p -value for each dataset is given in Table 4 in Appendix.

3.3 DECODING ALGORITHM

We present the Volatility-Scaled Guided Decoding (VSGD) algorithm, which incorporates reward volatility into a step-level beam search framework to prioritize reasoning paths exhibiting stable reward trajectories. In this framework, beam search maintains a fixed number of candidate sequences at each step, and VSGD ranks candidate sequences using reward values scaled by their volatility. To be specific, Algorithm 1 describes the scaling procedure used in VSGD algorithm. Given a reward trajectory $\mathbf{r}_t$, we compute its volatility $\sigma(\mathbf{r}_t)$ as in Eq. 1. Each reward value is then divided by $(\sigma(\mathbf{r}_t) + \epsilon)$, where $\epsilon = 10^{-6}$ prevents division by zero. This rescaling penalizes trajectories with high volatility by attenuating their reward values, thereby highlighting trajectories with more stable reward patterns.

Algorithm 2 outlines the overall decoding process of VSGD. At each step t , for each incomplete sequence $\mathbf{x}_{t-1}$, the function $\text{Expand}(\mathbf{x}_{t-1}, M)$ generates M candidate continuations by using $\mathbf{x}_{t-1}$ as the prefix. The PRM then assigns a step-level reward $r_t = f(\mathbf{x}_t)$ for each sequence

Table 1: Comparison of VSGD with baseline decoding algorithms on GSM8K, MATH500, and an MMLU-subset. The MMLU-subset comprises six categories: computer science (CS), chemistry (Chem), econometrics (Econ), formal logic (Logic), philosophy, and virology. Final answers are selected with Best-of-N (BoN) and Weighted Majority Voting (WMV); accuracy is reported with standard deviation as subscripts. Best and second-best results per dataset are shown in bold and underlined, and Δ indicates the accuracy difference between VSGD and the strongest baseline for each dataset. On average, VSGD improves over the strongest baseline by 1.1 points under BoN and 1.4 points under WMV.

Selection Dataset	BoN					WMV				
	GD-Sum	GD-Last	GD-Min	VSGD	Δ	GD-Sum	GD-Last	GD-Min	VSGD	Δ
MATH500	48.9 _{0.3}	49.5 _{0.9}	<u>50.2_{0.8}</u>	50.8_{0.9}	+0.6	49.8 _{0.5}	49.5 _{0.7}	<u>50.1_{0.8}</u>	50.8_{0.8}	+0.7
GSM8K	86.3 _{0.4}	<u>87.9_{0.2}</u>	88.1_{0.2}	87.9 _{0.5}	-0.2	87.2 _{0.4}	<u>88.2_{0.2}</u>	87.7 _{0.2}	88.3_{0.5}	+0.1
CS	50.3 _{1.8}	50.7 _{0.3}	51.7 _{0.3}	52.3_{2.7}	+0.6	56.0_{0.6}	53.3 _{0.9}	53.7 _{0.3}	55.3 _{2.3}	-0.7
Chem	40.3 _{2.8}	<u>44.7_{0.9}</u>	43.0 _{1.2}	46.3_{1.5}	+1.6	<u>44.3_{1.5}</u>	43.0 _{0.6}	43.3 _{1.2}	45.3_{1.3}	+1.0
Econ	51.2 _{0.6}	52.9 _{1.2}	<u>53.2_{1.5}</u>	53.8_{2.5}	+0.6	50.9 _{1.3}	51.5 _{0.8}	<u>52.9_{0.6}</u>	54.4_{1.8}	+1.5
Logic	45.5 _{1.4}	<u>47.4_{0.3}</u>	46.6 _{1.1}	48.4_{0.9}	+1.0	46.3 _{2.5}	46.6 _{1.9}	<u>49.5_{0.7}</u>	49.7_{0.5}	+0.2
Philosophy	60.2 _{1.1}	<u>62.0_{1.1}</u>	62.0 _{0.6}	63.3_{1.9}	+1.3	61.8 _{1.1}	<u>62.3_{1.1}</u>	60.6 _{0.7}	63.6_{2.0}	+1.3
Virology	47.0_{0.6}	45.8 _{1.5}	46.4 _{0.9}	47.0_{0.3}	0.0	<u>46.8_{1.0}</u>	46.2 _{0.9}	46.4 _{1.0}	47.4_{0.4}	+0.6
Average	53.7 _{0.6}	55.1 _{0.3}	<u>55.1_{0.1}</u>	56.2_{0.7}	+1.1	55.4 _{0.5}	55.1 _{0.2}	<u>55.5_{0.3}</u>	56.9_{0.7}	+1.4

$\mathbf{x}_t$. We append r_t to $\mathbf{r}_{t-1}$, forming $\mathbf{r}_t$, and compute the rescaled reward trajectory $\tilde{\mathbf{r}}_t$ using `RescaleWithVolatility` defined in Algorithm 1. We will use this rescaled reward trajectory $\tilde{\mathbf{r}}_t$ for ranking the reasoning path candidates.

We define the *candidate list* C_t which stores all candidates of reasoning path $\mathbf{x}_t$ as well as its reward sequence $\mathbf{r}_t$. Let `Agg` be an aggregation function which return a scalar value for a given input vector, *e.g.*, the minimum value within the input vector. We define the *aggregated score list* S_t which stores the scalar score obtained by applying the aggregation function `Agg` to $\tilde{\mathbf{r}}_t$, where the choice of aggregation depends on the decoding algorithm. Given the aggregated score list S_t , we sort them to choose N/M good candidates. Let `Argsort`(S_t, k) be the operation that returns the indices of the top- k elements in S_t , sorted in descending order of their scores. Then, the beam set $\mathcal{X}_t$ is updated by applying `Argsort`($S_t, \frac{N}{M}$) to identify the indices of the $\frac{N}{M}$ highest scores in S_t , and subsequently selecting the corresponding tuples $(\mathbf{x}_t, \mathbf{r}_t)$ from C_t .

When a reasoning sequence $\mathbf{x}_t$ in the updated beam set $\mathcal{X}_t$ reaches the end-of-sequence token `<eos>`, we call such sequence is *complete*, and store the complete sequences $\mathbf{x}_t$ in the list of complete reasoning paths $\mathcal{X}_{\text{comp}}$. The decoding process terminates when either the maximum reasoning step T is reached, or all sequences end, or the maximum number of completed reasoning paths L is obtained, after which the algorithm returns the list of complete reasoning paths $\mathcal{X}_{\text{comp}}$.

4 EXPERIMENTAL RESULT

We evaluate our proposed VSGD and conventional PRM-based decoding algorithms on two mathematical reasoning benchmarks, GSM8K and MATH500, as well as on the MMLU-subset. Sec. 4.1 reports the main results, showing that VSGD generally achieves higher accuracy on most benchmark datasets we tested. In addition, VSGD makes efficient use of the computational budget, consistently outperforming baseline methods across different limits on decoding steps T and the number of complete reasoning paths L . Sec. 4.2 further analyzes the factors contributing to the performance of VSGD, with particular focus on the reduced volatility of reward trajectories selected by VSGD compared to an existing decoding algorithm. We also examine how VSGD improves reasoning quality, finding that VSGD tends to mitigate the issue of incomplete reasoning. Our implementation is publicly available at <https://anonymous.4open.science/r/Volatility-Scaled-Guided-Decoding-B26C/README.md>.

Datasets. We evaluate VSGD on three datasets: two mathematical reasoning benchmarks – GSM8K (Cobbe et al., 2021) and MATH500 (Lightman et al., 2023) – as well as MMLU-subset. Here, MMLU-subset is a subset of MMLU dataset (Hendrycks et al., 2021), consisting of six subject areas: computer science, chemistry, econometrics, formal logic, philosophy, and virology. For all

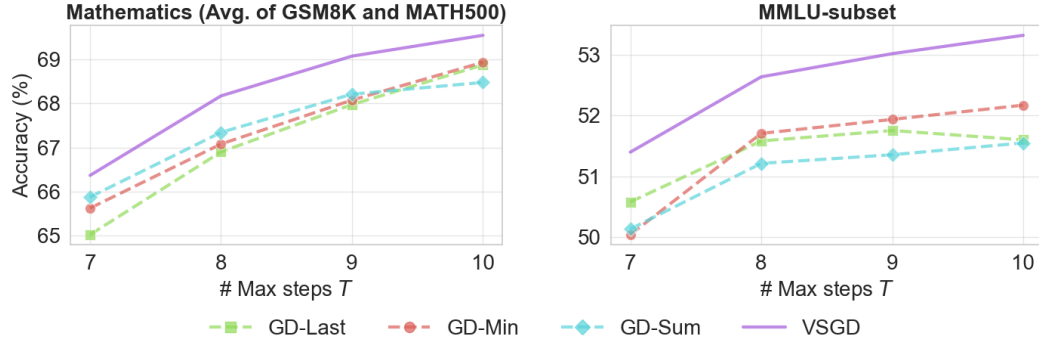


Figure 3: Accuracy of decoding algorithms under varying maximum number of reasoning step T in Algorithm 2. Results are reported on (Left) mathematical reasoning datasets (GSM8K and MATH500) and (Right) the MMLU-subset. The x -axis indicates the maximum number of reasoning steps T , and the y -axis denotes the accuracy of the final answer. For each T , only responses whose reasoning paths do not exceed T steps are retained when computing accuracy. Accuracy generally improves as T increases, and VSGD consistently outperforms the baselines, suggesting that VSGD can effectively utilize longer reasoning paths.

benchmark datasets, we report the average accuracy over three random seeds. The number of test queries for each dataset is listed in Table 3 in Appendix.

Models. We adopt LLAMA-3.1-8B-INSTRUCT (Llama-8B) (Grattafiori et al., 2024) as the base LLM and VersaPRM (Zeng et al., 2025) as the base PRM. It is notable that VersaPRM is trained on datasets covering subjects that include domains in the MMLU-subset and mathematical reasoning datasets. Unless otherwise specified, Llama-8B is used as the default LLM and VersaPRM as the default PRM throughout the experiments. All decoding methods are configured with the same LLM and PRM settings to ensure a fair comparison.

Baselines. We compare VSGD against baseline decoding algorithms that rank or expand incomplete sequences using step-level rewards estimated by a PRM. GD-Sum selects candidate continuations based on the total accumulated PRM rewards across decoding steps where we use PRMs as the state evaluator (Feng et al., 2023; Yao et al., 2023). GD-Min evaluates candidates by the minimum PRM-estimated reward across decoding steps (Lightman et al., 2023; Wang et al., 2024). GD-Last considers only the PRM-estimated reward at the final decoding step (Snell et al., 2024).

Our implementation builds on the decoding algorithm of Snell et al. (2024) to incorporate VSGD. We set the total beam size to $N = 8$ and the expansion width to $M = 2$. At each decoding step, the top $N/M = 4$ incomplete sequences with the highest aggregated scores, computed from rescaled reward trajectories $\tilde{r}_t$, are retained. Each retained sequence is then expanded into $M = 2$ candidates, yielding $N = 8$ candidate continuations for the next step. This procedure is repeated until a complete response is generated, with at most $T = 10$ reasoning steps and up to $L = 32$ complete reasoning paths, which together constrain the computational budget. For the reward trajectory aggregation function Agg , we follow prior work which use the minimum reward value of reward trajectories r_t , i.e., $\min_{i \in [t]} r_i$ (Sun et al., 2024; Wang et al., 2024; Zeng et al., 2025). For the final answer selection, we consider two strategies. The Best-of- N (BoN) strategy selects the candidate answer with the highest aggregated score, while the Weighted Majority Voting (WMV) strategy determines the final answer through voting, using the aggregated scores as weights (Li et al., 2023b).

4.1 DOES VSGD OUTPERFORM EXISTING GUIDED DECODING METHODS?

Yes, VSGD generally outperforms existing guided algorithms. Table 1 presents the performances of VSGD and baseline decoding methods, using Llama-3 as the base model and VersaPRM as the process reward model. Results are reported under two aggregation strategies (BoN and WMV), each averaged over three random seeds. Overall, VSGD achieves the highest average accuracy under both aggregation strategies, yielding relative improvements of 1.1 points over the strongest baseline in BoN and 1.4 points in WMV. At the dataset level, VSGD delivers overall gains: in Chemistry, it

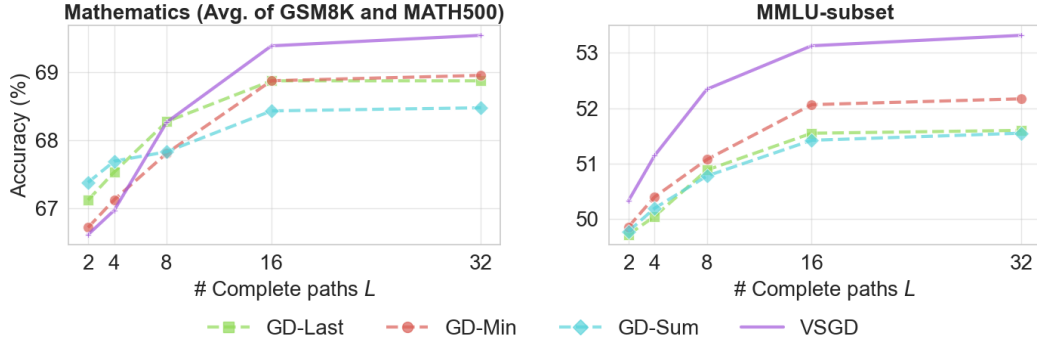


Figure 4: Accuracy of decoding algorithms with a limited number of reasoning paths L in Algorithm 2. We report the average accuracy on (Left) mathematical reasoning datasets, which consist of GSM8K and MATH500 and (Right) the MMLU-subset. The x -axis indicates the number of reasoning paths L , and the y -axis denotes accuracy of the final answer. For each L , we consider the first L reasoning paths that reach termination earliest during decoding. Across decoding algorithms, accuracy improves with larger L , reflecting the benefit of utilizing additional reasoning paths. This trend is pronounced for VSGD, which more effectively use additional reasoning paths compared to the baselines.

achieves a 1.6 point improvement under BoN; in Economics, a 1.5 point improvement under WMV; and in Philosophy, a 1.3 point improvement under both BoN and WMV. These gains are realized without substantial degradation in domains where baselines already achieve high accuracy, where performance of VSGD remains competitive despite minor drops. These results suggest that VSGD obtains performance improvements across both mathematical reasoning tasks and MMLU-subset.

Yes, VSGD effectively exploits reasoning steps. Fig. 3 compares the performances of VSGD and baseline decoding methods, by varying the maximum reasoning step T in Algorithm 2. Recall that all experiments are conducted with the default budget $T = 10$. To analyze performance under smaller budgets ($T < 10$), we exclude reasoning paths whose length exceeds T , and then compute accuracy on the remaining reasoning paths. Across all decoding algorithms, accuracy increases as T grows, highlighting the benefit of longer reasoning trajectories. Moreover, VSGD consistently surpasses the baselines, demonstrating stronger capability in leveraging reasoning steps to reach correct solutions.

Yes, VSGD achieves higher accuracy with additional reasoning paths. We consider a scenario where only the first L complete reasoning paths are used for evaluation. By default, we set $L = 32$, and here we investigate how performance varies as L increases. Fig. 4 reports the accuracy of decoding algorithms as L varies in $\{2, 4, 8, 16, 32\}$. Our experimental results show that, across decoding algorithms, accuracy generally increases with larger L , as more reasoning paths become available. Moreover, VSGD generally achieves higher accuracy than the baselines, indicating the effectiveness of VSGD in using additional reasoning paths compared to existing baseline methods.

4.2 WHAT EFFECTS DOES VSGD EXHIBIT IN DECODING?

VSGD prioritizes stable reward trajectories.

Table 2 reports a comparative analysis of volatility between VSGD and GD-Min. We use GD-Min as the baseline because it selects trajectories solely based on PRM reward scores without considering reward volatility, whereas VSGD incorporates reward volatility by weighting trajectories with more stable reward signals. The Δ column in Table 2 reports the relative reduction (%) of VSGD compared to GD-Min. Compared to GD-Min, VSGD shows lower volatility $\sigma(r_t)$ defined in Eq. 1, measured across the decoding steps. For example, volatility is reduced by -9.48% on the CS category and by -8.92% on Philosophy, with an average reduction

Table 2: Comparison between VSGD and GD-Min, in terms of volatility of the trajectory.

Dataset	GD-Min	VSGD	Δ (%)
MATH500	0.0587	0.0537	-8.49
GSM8K	0.0267	0.0246	-7.82
CS	0.1380	0.1249	-9.48
Chem	0.1041	0.1081	3.85
Econ	0.1543	0.1438	-6.75
Logic	0.1345	0.1303	-3.10
Philosophy	0.1100	0.1002	-8.92
Virology	0.1216	0.1180	-2.96
Average	0.1065	0.1010	-5.13

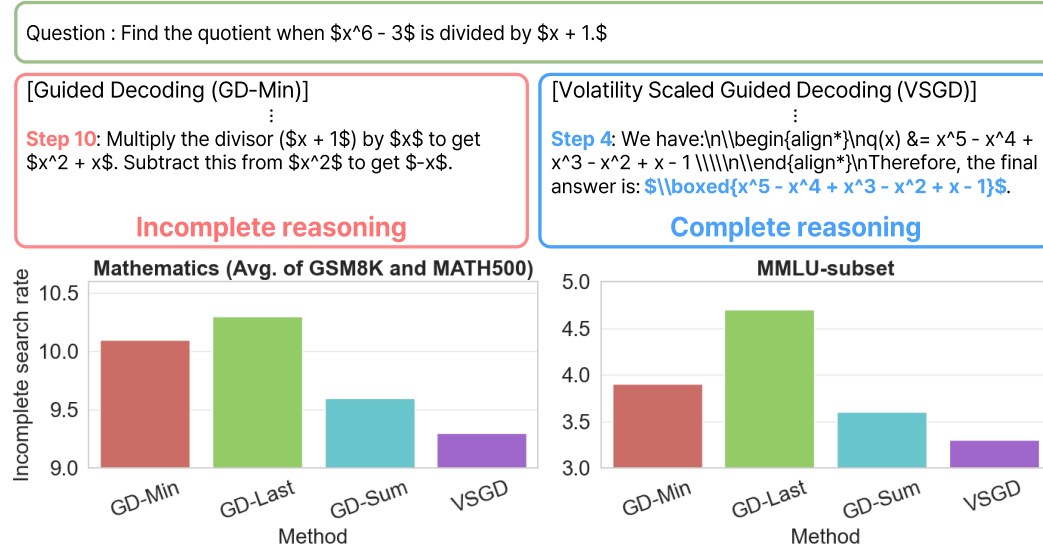


Figure 5: Qualitative and quantitative results showing that VSGD mitigates incomplete reasoning. **(Top)** Example reasoning paths from the MATH500 dataset: the baseline method GD-Min reaches the maximum step limit $T = 10$ without generating the final answer, while VSGD generates a complete reasoning path with the final answer. **(Bottom)** Empirical comparison of incomplete search rates on the MMLU-subset and mathematics benchmarks which is comprised of GSM8K and MATH500. For each dataset and decoding algorithm, we compute the proportion of cases that reach the maximum step limit T without generating the final answer, referred to as the incomplete search rate. We report averages across datasets for each decoding algorithm. Our experimental results show that VSGD consistently yields lower incomplete search rates than baseline methods.

of -5.13% across all datasets. The reduction across benchmarks is statistically significant under the Wilcoxon signed-rank test (Wilcoxon, 1945) yielding $p = 0.0098$, which is appropriate given that both methods are evaluated on the same datasets. These results demonstrate that VSGD effectively guides decoding toward reasoning paths whose reward trajectories are less volatile compared to GD-Min.

VSGD improves accuracy by mitigating failed searches and incomplete reasoning. One key advantage of VSGD over baseline methods is its ability to reduce cases where the reasoning process remains unfinished. Fig. 5 provides both qualitative and quantitative evidences. The top panel of Fig. 5 provides a qualitative example from the MATH500 dataset: the baseline method GD-Min reaches the maximum step limit $T = 10$ without producing a final answer, whereas VSGD completes the reasoning in only four steps and outputs the correct solution. The bottom panel of Fig. 5 reports quantitative results on the MMLU-subset and mathematics (GSM8K and MATH500) benchmarks. We measure the proportion of test cases where decoding terminates without producing the final answer, which we define as the *incomplete search rate*. Across all datasets and methods, VSGD consistently achieves substantially lower incomplete search rates than baselines, demonstrating its effectiveness in generating complete reasoning trajectories.

5 CONCLUSION

We introduced Volatility-Scaled Guided Decoding (VSGD), a decoding algorithm that leverages volatility in reward trajectories within a step-level beam search framework. Our analysis revealed that correct reasoning paths consistently exhibit lower volatility than incorrect ones, motivating the use of volatility as a stability criterion in decoding. Experiments on GSM8K, MATH500, and the MMLU-subset show that VSGD outperforms strong baselines in accuracy while using computational resources more efficiently. By reducing volatility, VSGD improves the rate of successful searches and thus explains its performance gains. More broadly, these results highlight that modeling the temporal dynamics of reward trajectories offers a principled foundation for guided decoding in LLMs.

REFERENCES

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- CHOW Alan and Lahtinen Kyre. Equity risk: Measuring return volatility using historical high-frequency data. *Studies in Business and Economics*, 14(3):60–71, 2019.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. *arXiv preprint arXiv:1912.03193*, 2019.
- Lars Brockmann and Kenneth J Hunt. Heart rate variability changes with respect to time and exercise intensity during heart-rate-controlled steady-state treadmill running. *Scientific Reports*, 13(1):8515, 2023.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Antoine Chaffin, Vincent Claveau, and Ewa Kijak. PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding. In Marine Carpuat, Marie-Catherine de Marnette, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2953–2967, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.215. URL <https://aclanthology.org/2022.naacl-main.215/>.
- Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. Neural keyphrase generation via reinforcement learning with adaptive rewards. *arXiv preprint arXiv:1906.04106*, 2019.
- Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: process supervision without process. *Advances in Neural Information Processing Systems*, 37:27689–27724, 2024.
- 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*, 2021.
- Kevin Daly. Financial volatility: Issues and measuring techniques. *Physica A: statistical mechanics and its applications*, 387(11):2377–2393, 2008.
- Kevin Daly. An overview of the determinants of financial volatility: An explanation of measuring techniques. *Modern Applied Science*, 5(5):46, 2011.
- Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.
- 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.
- Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173. Association for Computational Linguistics, December 2023. doi: 10.18653/v1/2023.emnlp-main.507. URL <https://aclanthology.org/2023.emnlp-main.507/>.

- Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. Decoding with value networks for neural machine translation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/2b24d495052a8ce66358eb576b8912c8-Paper.pdf.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
- Pengfei Hu, Zhenrong Zhang, Qikai Chang, Shuhang Liu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, Feng Ma, et al. Prm-bas: Enhancing multimodal reasoning through prm-guided beam annealing search. *arXiv preprint arXiv:2504.10222*, 2025.
- Martin Josifoski, Maxime Peyrard, Frano Rajič, Jiheng Wei, Debjit Paul, Valentin Hartmann, Barun Patra, Vishrav Chaudhary, Emre Kiciman, and Boi Faltings. Language model decoding as likelihood–utility alignment. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1455–1470, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.107. URL <https://aclanthology.org/2023.findings-eacl.107/>.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pp. 3499–3508. PMLR, 2019.
- Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. RankGen: Improving text generation with large ranking models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 199–232, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.15. URL <https://aclanthology.org/2022.emnlp-main.15/>.
- Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislari, Lespiau Jean-Baptiste, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. Machine translation decoding beyond beam search. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8410–8434, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.662. URL <https://aclanthology.org/2021.emnlp-main.662/>.
- Youngwon Lee, Jinu Lee, and Seung-won Hwang. Learning to rank generation with pairwise partial rewards. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6078–6092, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.371. URL <https://aclanthology.org/2023.emnlp-main.371/>.
- Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Language modeling with latent situations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12556–12571, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.795. URL <https://aclanthology.org/2023.findings-acl.795/>.
- Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5315–5333, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.291. URL <https://aclanthology.org/2023.acl-long.291/>.

- 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.
- Jiacheng Liu, Andrew Cohen, Ramakanth Pasunuru, Yejin Choi, Hannaneh Hajishirzi, and Asli Celikyilmaz. Don’t throw away your value model! generating more preferable text with value-guided monte-carlo tree search decoding. *arXiv preprint arXiv:2309.15028*, 2023.
- Xin Liu, Farima Fatahi Bayat, and Lu Wang. Enhancing language model factuality via activation-based confidence calibration and guided decoding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10436–10448, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.583. URL <https://aclanthology.org/2024.emnlp-main.583/>.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4288–4299, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.339. URL <https://aclanthology.org/2021.naacl-main.339/>.
- Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pp. 50–60, 1947.
- Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082, 15406261. URL <http://www.jstor.org/stable/2975974>.
- Clara Meister, Afra Amini, Tim Vieira, and Ryan Cotterell. Conditional Poisson stochastic beams. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 664–681, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.52. URL <https://aclanthology.org/2021.emnlp-main.52/>.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.
- Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. A thorough examination of decoding methods in the era of LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8601–8629, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.489. URL <https://aclanthology.org/2024.emnlp-main.489/>.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. *arXiv preprint arXiv:2403.09472*, 2024.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5a18e133cbf9f257297f410bb7eca942-Paper.pdf.

- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 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.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL <https://aclanthology.org/2024.acl-long.510/>.
- Zhihai Wang, Jie Wang, Jilai Pan, Xilin Xia, Huiling Zhen, Mingxuan Yuan, Jianye Hao, and Feng Wu. Accelerating large language model reasoning via speculative search. *arXiv preprint arXiv:2505.02865*, 2025.
- 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.
- Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. *Advances in Neural Information Processing Systems*, 35:4913–4927, 2022.
- Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- Yilin Yang, Stefan Lee, and Prasad Tadepalli. Language-informed beam search decoding for multilingual machine translation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15761–15772, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.932. URL <https://aclanthology.org/2024.findings-acl.932/>.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.
- Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, et al. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*, 2025.
- Dan Zhang, Min Cai, Jonathan Li, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Tdrm: Smooth reward models with temporal difference for llm rl and inference. *arXiv preprint arXiv:2509.15110*, 2025a.
- Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. OpenPRM: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=fGIQgFmgkW>.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Yingjie Wang, Baisheng Lai, Jieping Ye, Mingli Song, and Dacheng Tao. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.08745*, 2025c.

Table 3: Number of evaluation queries used per dataset.

Dataset	# Test Queries
<i>Task category: Mathematical reasoning</i>	
GSM8K	1319
MATH500	500
<i>Task category: MMLU-subset knowledge</i>	
Computer Science	100
Chemistry	100
Econometrics	114
Formal Logic	126
Philosophy	311
Virology	166

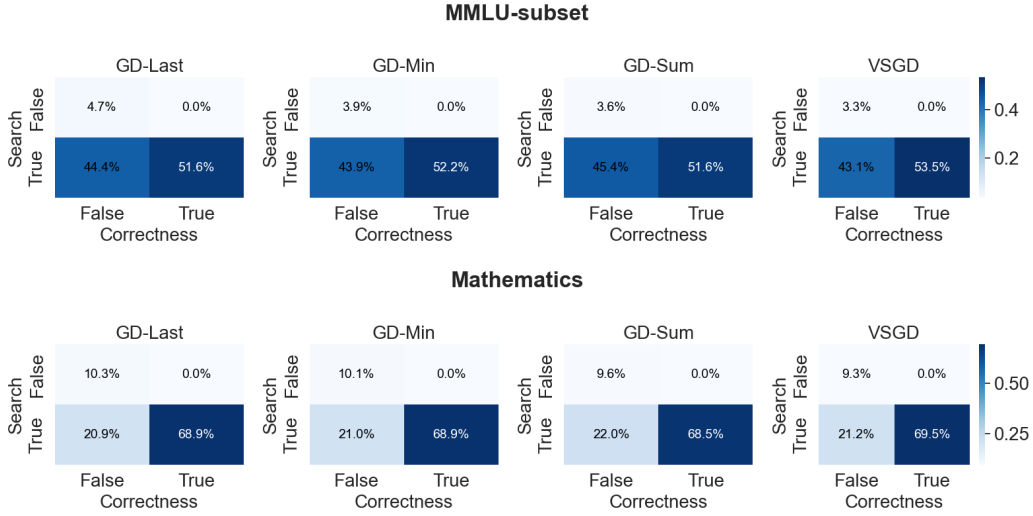


Figure 6: Distribution of outcomes across guidance methods, categorized by search completion (y-axis) and answer correctness (x-axis). Each heatmap illustrates the proportion of examples in four categories defined by whether the decoding algorithm completes a search and whether the final answer is correct. For instance, the *Search=True, Correct=True* cell corresponds to accurate responses, while *Search=True, Correct=False* denotes cases where the search completes but yields an incorrect answer. Across both the **(Top)** MMLU-subset and **(Bottom)** mathematics datasets, VSGD generally results in fewer search failures and fewer incorrect answers upon completion compared to baseline methods, producing a higher proportion of accurate responses. These results demonstrate that VSGD improves reasoning performance by enhancing both search completion and answer correctness.

A EXPERIMENTS DETAILS

We evaluate our models on a subset of the MMLU benchmark covering six domains. Among these, Chemistry and Computer Science are selected from the college-level categories of MMLU, while the remaining subjects represent diverse knowledge domains. The number of test samples for each dataset is summarized in Table 3.

To ensure efficient inference, we apply 4-bit quantization to all models and enable FlashAttention in every experiment. This configuration reduces memory consumption and accelerates attention computation without compromising accuracy. All experiments are conducted on NVIDIA RTX 4090 GPUs.

B COMPREHENSIVE RESULTS

Fig. 6 presents the distribution of outcomes by jointly considering search completion and answer correctness. We evaluate decoding algorithms by partitioning examples into four categories: (i) complete search with a correct answer, (ii) complete search with an incorrect answer, (iii) failed search with a correct answer, which is essentially absent, and (iv) failed search with an incorrect answer. This decomposition highlights how decoding algorithms influence the reasoning performance of LLMs. Across both the MMLU-subset and mathematical reasoning datasets including GSM8K and MATH500, VSGD consistently shows lower rates of search failure compared to baseline methods. Moreover, within the subset of successful searches, VSGD generally produces a smaller fraction of incorrect answers, suggesting that the guidance of VSGD is associated with both higher completion rates and improved output quality. As a result, VSGD yields a higher proportion of accurate responses and a lower proportion of failed or misleading trajectories relative to the baselines. These findings suggest that the performance gains of VSGD can be attributed to two complementary factors: reducing the frequency of failed searches and improving correctness within successful searches.

Table 4: Mann–Whitney U test p -values showing significant differences in volatility distributions between correct and incorrect reasoning paths across domains.

Dataset	p-value
GSM8K	1.1×10^{-36}
MATH500	7.4×10^{-49}
Computer Science	2.2×10^{-3}
Chemistry	1.4×10^{-5}
Econometrics	2.6×10^{-7}
Formal Logic	1.2×10^{-5}
Philosophy	4.8×10^{-52}
Virology	1.9×10^{-8}

Table 4 reports the p -values from the Mann–Whitney U test across mathematical reasoning benchmarks (GSM8K and MATH500) as well as the MMLU subset, which includes Chemistry, Computer Science, Philosophy, Econometrics, Formal Logic, and Virology. In all cases, the p -values are well below the 0.05 threshold, indicating that the volatility distributions of correct and incorrect reasoning paths differ significantly across domains, thereby reinforcing the patterns observed in Fig. 2.