

# Improving Value-based Process Verifier via Low-Cost Variance Reduction

Zetian Sun, Dongfang Li\*, Baotian Hu, Min Zhang

Harbin Institute of Technology, Shenzhen  
zetiansun.cs@gmail.com, {lidongfang,hubaotian,zhangmin2021}@hit.edu.cn

## Abstract

Large language models (LLMs) have achieved remarkable success in a wide range of tasks. However, their reasoning capabilities, particularly in complex domains like mathematics, remain a significant challenge. Value-based process verifiers, which estimate the probability of a partial reasoning chain leading to a correct solution, are a promising approach for improving reasoning. Nevertheless, their effectiveness is often hindered by estimation error in their training annotations, a consequence of the limited number of Monte Carlo (MC) samples feasible due to the high cost of LLM inference. In this paper, we identify that the estimation error primarily arises from high variance rather than bias, and the MC estimator is a Minimum Variance Unbiased Estimator (MVUE). To address the problem, we propose the Compound Monte Carlo Sampling (ComMCS) method, which constructs an unbiased estimator by linearly combining the MC estimators from the current and subsequent steps. Theoretically, we show that our method leads to a predictable reduction in variance, while maintaining an unbiased estimation without additional LLM inference cost. We also perform empirical experiments on the MATH-500 and GSM8K benchmarks to demonstrate the effectiveness of our method. Notably, ComMCS outperforms regression-based optimization method by 2.8 points, the non-variance-reduced baseline by 2.2 points on MATH-500 on Best-of-32 sampling experiment.

## 1 Introduction

In recent years, large language models (LLMs) have demonstrated remarkable capabilities across various reasoning tasks that require complex multi-step reasoning, such as mathematics and programming (Hurst et al. 2024; Yang et al. 2024a,b; Dubey et al. 2024), yet they still make mistakes when solving challenging problems. To address this issue, verification-based method has recently emerged to improve LLM reasoning (Uesato et al. 2022; Lightman et al. 2024). In general, verification models are trained to evaluate and potentially correct the reasoning trajectories during the reasoning process, which can ensure higher accuracy and consistency in LLM outputs by re-ranking candidate responses. They can also provide valuable feedback for further improvement of LLMs (Li et al. 2023; Wu et al. 2023).

\*Corresponding author

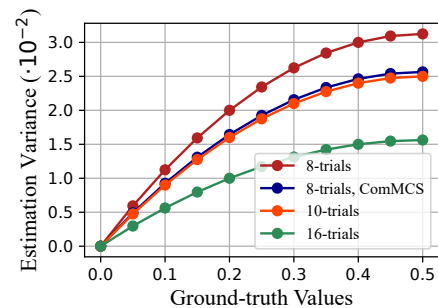


Figure 1: Illustration of the estimation variance across different ground-truth values. We compare variances with number of trials ranging from  $\{8, 10, 16\}$ , and variance applying ComMCS. The variance after using our method (8-trials, ComMCS) is approximately the variance obtained by using 25% more sampling samples (10-trials).

Verification-based methods can be categorized by annotation granularity and strategy, with the value-based process verifier being one approach in this classification. Specifically, value-based process verifier is a kind of **process-supervised verifier**, as the training annotations are based on evaluations of each reasoning step. Value-based process verifier is also a kind of **value-based verifier**, as the verifier aims at predicting the success rate of each reasoning step, which is the empirical probability of this step leading to the correct final answer in mathematical reasoning scenario, i.e., the value of the current step. The training data of value-based process verifier mainly relies on Monte Carlo (MC) estimation. Typically, multiple reasoning trajectories are independently generated by LLMs, and the success rate is computed as the average outcome correctnesses across all trajectories. Outcome correctness is verified through reliable rule-based matching. However, the value estimation is inaccurate due to the limitation of sampling size, which is restricted due to the high cost of LLM inference (Zhang et al. 2025b). This motivates us to study the following research question: *is it possible to reduce the estimation error when estimating values without introducing additional LLM inference overhead?*

In this paper, we present an analysis about the MC estimation error in mathematical reasoning scenario from the perspective of Markov Decision Process (MDP) in §4.1. We

build a bridge between the MC estimation and the Binomial distribution, and conclude that the MC estimation is the Minimal Variance Unbiased Estimator (MVUE). The conclusion implies that estimation error is due to estimation variance instead of biased estimation, and the variance cannot be reduced without introducing additional information.

Based on the analysis, we propose ComMCS in §4.2, an unbiased estimation method that utilizes compound MC sampling results to reduce variance without additional LLM inference overhead. This core idea is conceptually analogous to variance reduction techniques in reinforcement learning, such as Temporal Difference (TD) learning, where future value estimates are used to update current ones. We adapt this principle to the context of LLM verifiers. Specifically, we quantify the form of the variance of MC estimation that is based on current step, and the compound variance of MC estimation that is based on the subsequent few steps. Through theoretical analysis, we show that (1) *the linear combination of MC estimations of subsequent few steps is also an unbiased estimation of the value of current step*, and (2) *under specific conditions, the estimation variance can be reduced*. Typically, the condition is related to the distribution of future values and the coefficients of linear combinations.

We provide the practical implementation of ComMCS in §4.3, where we narrow down the linear combination into two items, the current step and its next step. We make several approximations. First of all, we use a categorical distribution to estimate the one-step value distribution, which models the shape of distribution of next step value and is used to estimate the variances. The categorical distribution is modeled by the output of our value-based process verifier, and the verifier is then optimized by cross entropy loss. Secondly, we assume that the one-step value distribution follows the Gaussian distribution class. Finally, we use the MC estimations of values as the proxy of true values to estimate the variances. These practical approximations allow us to calculate and compare variances, after which we find the suitable coefficients for items in the linear combination heuristically and optimize verifier iteratively. The simulated effect of our variance reduction method is shown in Figure 1, where we assume value distribution follows Gaussian distribution.

We then perform extensive experiments on two mathematical reasoning tasks: MATH-500 (Lightman et al. 2024) and GSM8K (Cobbe et al. 2021) to demonstrate the effectiveness of our method across different model series and search strategies. Compared with verifiers that preform return distribution modeling or regression, the value distribution modeling method can achieve comparable performances. After performing our method, we observe a consistent improvement across different settings. For example, our method trained on Deepseek-math-7b-instruct outperforms baselines by 2-3 points on Math-500 on Best-of-N sampling. In beam search experiments, our method trained on Deepseek-math-7b-instruct outperforms other baselines by 1-2 points on MATH-500.

Our contributions are summarized as follows:

- We are the first to systematically identify and address the high variance of MC estimators as key bottleneck limiting the performance of value-based process verifiers.

- We propose **ComMCS**, a theoretically-grounded method that reduces variance by compounding estimators from subsequent steps without extra LLM inference cost.
- Our extensive experiments show that ComMCS achieves improved performance over baseline verifiers and demonstrate the potential for diversification in modeling the objective of value-based process verifiers.

## 2 Related Work

### 2.1 Test-time Scaling for Math Reasoning

Math reasoning task remains a significant challenge for LLMs (Lightman et al. 2024; Zheng et al. 2024). The test-time scaling technique requires models to generate long Chain-of-Thought (CoT) explicitly or implicitly as its reasoning steps, which can effectively improve the reasoning capabilities of LLMs (Snell et al. 2024; Wang et al. 2024a; Liu et al. 2025). Prior efforts have explored different methodologies to improve the math reasoning capability of language models, such as training-based methods including pre-training (Azerbayev et al. 2024) and fine-tuning (Luo et al. 2025; Yu et al. 2024; Wang et al. 2023), and prompt-based methods like few-shot prompting (Wei et al. 2022) and in-context learning (Zhou et al. 2022). However, these approaches can be resource-intensive (Lu et al. 2024) or require human-crafted demonstrations (Qin et al. 2024). Unlike methods that directly modify parameters or prompts, our method is based on verification-based method, which focuses on training an additional verification model to select the desired output from the candidate model outputs.

### 2.2 Process-Supervised Verifier (PSV)

Researchers have found that PSVs that are trained on fine-grained annotations are effective for LLM reasoning, compared with Outcome-Supervised Verifiers (OSV, Lightman et al. (2024)). Depending on the definition of fine-grained annotations, PSVs can be classified as several variants: (1) *Reward-based Process Verifiers*. The annotations are based on the correctness of current reasoning step (Lightman et al. 2024) or rule-based strategies like calculation errors and formatting errors (Xi et al. 2024; Zheng et al. 2024). (2) *Value-based Process Verifiers*. The annotations are based on the expected return of current reasoning step, where the supervision signals are collected via repeated rollouts and MC sampling (Wang et al. 2024b; Luo et al. 2024). (3) *Generative Verifiers*. The annotations are based on the CoT capabilities of LLMs (Zhang et al. 2025a; Zhao et al. 2025; Khalifa et al. 2025). Among the many variants of PSVs, our work focuses on optimizing value-based process verifier, which is able to balance short-term mistakes and long-term gains, i.e., achieve credit assignments (Setlur et al. 2024).

## 3 Preliminaries

### 3.1 Modeling Math Reasoning as an MDP

Given a question  $q$ , the answer generation process of LLM  $\pi$  can be broken down into several non-overlapped reasoning steps. The atomic reasoning step can be a single line that is split by delimiters like “\n” (Lightman et al. 2024), sentence

separated by supporting clauses (OpenAI 2025), or implicitly reflect (Shinn et al. 2023) on their past reasoning steps and do inner-monologue (Huang et al. 2022) with sentences that start with “wait”.

The reasoning process can thus be conceptualized as an MDP  $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ , where the state  $s_t \in \mathcal{S}$  denotes the concatenation of the question and the partial reasoning trace already generated by timestep  $t - 1$  and  $s_1 = q$  as the initial state. The action  $a_t = \pi(\cdot|s_t) \in \mathcal{A}$  represents that for any time step  $t$ , the reasoning step  $a_t$  is generated by  $\pi$  given current state  $s_t$ . The state transition  $\mathcal{T}$  is deterministic in math reasoning scenario, as the state transition from state  $s_t$  to  $s_{t+1}$  is accomplished through a simple operation of concatenation. The reward  $\mathcal{R}$  is often outcome-based, where for each intermediate reasoning step, the reward is set to 0, as a common practice in previous studies (Wang et al. 2024b; Lightman et al. 2024). Let  $T$  be the terminal state, the reward can be represented as follows:

$$\mathcal{R}(s_t, a_t) = \begin{cases} 1 & t = T \wedge [s_t; a_t] \text{ is correct} \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

and the discount factor  $\gamma$  is set to 1.

The state value  $V^\pi(s) = \mathbb{E}[G_t|S_t = s]$  is the expectation of the return  $G_t$  given state  $s$ , where return  $G_t = \sum_{i=t+1}^T \gamma^{i-t-1} R_i$ . The state-action value  $Q^\pi(s, a) = \mathbb{E}[G_t|S_t = s, A_t = a]$  is the expectation of the return  $G_t$  when taking action  $a$  at state  $s$ . The state value  $V^\pi(s)$  can be expressed as the expected value of  $Q^\pi(s, a)$  under the policy  $\pi$ :

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \cdot Q^\pi(s, a). \quad (2)$$

Following the MDP settings in math reasoning scenarios, the return  $G_t$  depends on the outcome reward only:

$$G_t = \sum_{i=t+1}^T \gamma^{i-t-1} R_i = R_T, \quad (3)$$

which allows for an estimation of state value  $V^\pi(s)$  via MC estimation that samples different returns under the state  $s$ :

$$\begin{aligned} V^\pi(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_T|S_t = s] \\ &\approx \frac{1}{N} \sum_{\tau^{(i)} \sim \pi(\cdot|s)} [R_T^{(i)}|s, \tau^{(i)}], \end{aligned} \quad (4)$$

where  $\tau^{(i)}$  is the  $i$ -th trajectory sampled under  $\pi$  at state  $s$ .

### 3.2 Value-based Process Verifier

A value-based process verifier  $f_\theta : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$  is trained to estimate  $Q^\pi(s, a)$  for each intermediate or final state, acting as a surrogate for expensive MC sampling during the reasoning process, which is commonly used in previous studies (Snell et al. 2025; Luo et al. 2024; Wang et al. 2024b).

Given state  $s$  and action  $a$ , we use policy  $\pi$  to acquire the estimated state-action value  $\hat{Q}^\pi(s, a)$  by MC estimation.

Then an offline dataset  $\mathcal{D} = \{s, a, \hat{Q}^\pi(s, a)\}$  is collected to train the value-based process verifier.

The value-based process verifier can be trained to model the return distribution of  $Q^\pi$  based on the binary support set  $\{0, 1\}$ . The verifier is then optimized by minimizing a soft Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{bce}(f_\theta) = \mathbb{E}_{(s,a,\hat{Q}^\pi(s,a)) \sim \mathcal{D}} [-\hat{Q}^\pi(s, a) \log f_\theta(s, a)], \quad (5)$$

where  $f_\theta(s, a) = 0 \cdot \mathbb{P}_\theta(y = 0|s, a) + 1 \cdot \mathbb{P}_\theta(y = 1|s, a)$ . As an alternative to the soft BCE loss, the value-based process verifier can be trained to predict value directly by minimizing a Mean-Squared Error (MSE) loss:

$$\mathcal{L}_{mse}(f_\theta) = \mathbb{E}_{(s,a,\hat{Q}^\pi(s,a)) \sim \mathcal{D}} -[f_\theta(s, a) - \hat{Q}^\pi(s, a)]^2, \quad (6)$$

where  $f_\theta(s, a)$  is a regression model that predicts  $Q^\pi(s, a)$  directly. We provide more details in Appendix.

## 4 Methodology

In this section, we start with an introduction on the statistics properties of MC estimation, which is used to estimate  $Q^\pi(s, a)$  in every intermediate reasoning steps. Then, we formally introduce ComMCS, our variance reduction method that utilizes **Compound Monte Carlo Sampling** results, which aims at reducing the estimation variance while maintaining the unbiasedness property. Furthermore, we introduce the practical implementation of ComMCS by modeling value distribution of next state-action value distribution. All proofs are provided in Appendix.

### 4.1 Analysis of MC Estimation

First of all, we build a bridge between the current math reasoning problem and classical Binomial distribution. The following theorem holds under current MDP condition:

**Theorem 4.1.** (Equivalence of MC Value Estimation for Binary Returns and Binomial Distribution) *Suppose the minimal support set of the return distribution is  $\{0, 1\}$ . Let  $V^\pi(s)$  be the true state value for given policy  $\pi$ , i.e., the expected return starting from state  $s$  and policy  $\pi$ .*

*Suppose we estimate  $V^\pi(s)$  via MC estimation, i.e.,*

- *Simulating  $N$  independent episodes starting from state  $s$  and following policy  $\pi$ .*
- *For each episode  $i \in \{1, \dots, N\}$ , observing the realized return  $G^{(i)} \in \{0, 1\}$ .*
- *Estimating the value  $V^\pi(s)$  as the empirical average of the observed returns:  $\hat{V}^\pi(s) = \frac{1}{N} \sum_{i=1}^N G^{(i)}$ .*

*Then, the MC estimation process is probabilistically equivalent to sampling from a binomial distribution, where the probability of success is  $p = V^\pi(s)$  and the number of trials is  $N$ . Specifically, the sum of observed returns,  $\sum_{i=1}^N G^{(i)}$ , follows a binomial distribution  $B(N, V^\pi(s))$ .*

The equivalence allows us to analyze the properties of MC estimation under a predefined distribution class, which is the binomial distribution class. Given the number of trials  $N$  and the true state value  $V^\pi(s)$ , define  $X$  as the sum of  $N$  Bernoulli trials, the expectation of MC estimation is

$$\mathbb{E}[\hat{V}^\pi(s)] = \mathbb{E}\left[\frac{X}{N}\right] = \frac{1}{N} \mathbb{E}[X] = V^\pi(s), \quad (7)$$

and the variance of MC estimation is

$$\mathbb{V}[\hat{V}^\pi(s)] = \mathbb{V}\left[\frac{X}{N}\right] = \frac{V^\pi(s) \cdot (1 - V^\pi(s))}{N}. \quad (8)$$

As the expectation of MC estimation is exactly  $V^\pi(s)$ , the estimation is unbiased. The variance of MC estimation is affected by the number of trials  $N$  and state value  $V^\pi(s)$ . For math reasoning scenarios, the number of trials are restricted due to the high cost of LLM inference, which results in the unneglectable estimation error and inferior performance of models trained on the estimated values (Zhang et al. 2025b; Chen et al. 2025). However, it is difficult to reduce the variance while maintaining the unbiasedness, which is due to the MVUE property of MC estimation.

**Theorem 4.2.** (Optimality of the MC Estimator) *Let  $X_1, X_2, \dots, X_n$  be independent and identically distributed (i.i.d) random variables following a Bernoulli distribution with parameter  $p$ , i.e.,*

$$X_i \sim \text{Bernoulli}(p), i = 1, 2, \dots, n.$$

*Define the MC estimator as the sample mean:  $\hat{p}_n = \frac{1}{n} \sum_{i=1}^n X_i$ . Then, the MC estimator is the Minimum Variance Unbiased Estimator (MVUE).*

The theorem implies that MC estimation achieves the minimum variance among all unbiased estimators, given the limited information (i.e., number of trials) defined by Fisher Information in Appendix. In the following section, we introduce our approach towards reducing the variance of value estimation while maintaining the unbiasedness of MC estimation by incorporating additional unbiased information, which is inspired by Temporal Difference (TD).

## 4.2 Variance Reduction via Compound Sampling

We now formally derive ComMCS, our variance reduction method that reduces the estimation variance while maintain the unbiasedness property. To put it simply, we try to compound the MC results of multiple steps as additional information to reduce the variance of the MC estimation of current step. We start with the Bellman Equation between state-action value  $Q^\pi(s, a)$  and next state value  $V^\pi(s')$  under deterministic transition scenario:

$$Q^\pi(s, a) = R(s, a) + \gamma V^\pi(s'), \quad (9)$$

where  $s'$  the next state transitioned from current state  $s$  and action  $a$ . Under the MDP condition defined in §3.1, the above expression can be further simplified assuming  $V^\pi(s') = R(s')$  when  $s'$  being the terminal state:

$$Q^\pi(s, a) = V^\pi(s'). \quad (10)$$

Combining Eq. (2) and Eq. (10), we have

$$\begin{aligned} V^\pi(s_n) &= \sum_{a_n \in \mathcal{A}} \pi(a_n | s_n) \cdot V^\pi(s_{n+1}) \\ &= \sum_{[a_n; \dots; a_m] \in \mathcal{A}^{m-n+1}} \prod_{i=0}^{m-n} \pi(a_{n+i} | s_{n+i}) V^\pi(s_m) \\ &= \mathbb{E}_\pi[V^\pi(s_m | s_n)], \end{aligned} \quad (11)$$

where  $s_m$  is the  $m$ -th state subsequent to the current state  $s_n$ . The above equation implies that we are able to estimate the value of current step (i.e.,  $V^\pi(s_n)$ ) through MC sampling from any future step (e.g.,  $V^\pi(s_m | s_n)$ ). We can also estimate  $V^\pi(s_n)$  through the linear combination of MC estimations based on several future steps. Let  $\{c_i\}$  be the coefficients of different expectations with  $\sum_i c_i = 1$ , we have

$$V^\pi(s_n) = \sum_{i=1}^{|\{c_i\}|} c_i \cdot \mathbb{E}_\pi[V^\pi(s_{n+i} | s_n)]. \quad (12)$$

The MC estimation via Eq. (12) is an unbiased estimation due to the linear additivity of expectations.

Let  $\sigma_n^2$  be the variance of a Bernoulli distribution with parameter  $p = V^\pi(s_n)$ . We can rewrite the variance of MC estimation at state  $s$  and policy  $\pi$ , given Eq. (8) and the total number of trials  $N$ :

$$\mathbb{V}[\hat{V}^\pi(s_n)] = \frac{1}{N} \sigma_n^2, \quad (13)$$

and rewrite the compound variance when estimating  $V^\pi(s_n)$  via the linear combination of MC estimations of several future steps into the expression introduced in Theorem 4.3.

**Theorem 4.3.** (Compound variance of MC estimation) *Following  $c_i, \sigma_n^2$  defined in Eq. (12) and (13). Let  $V_n, \hat{V}_n$  be the true value and the MC estimated value under state  $s_n$ ,  $\mathbb{V}[\hat{V}_{n \rightarrow m} | s_n]$  be the compound variance when estimating  $V_n$  by the linear combination of the MC estimations of states from  $s_n$  to  $s_m$  conditioned on state  $s_n$ . The variance  $\mathbb{V}[\hat{V}_{n \rightarrow m} | s_n]$  satisfies*

$$\begin{aligned} \mathbb{V}[\hat{V}_{n \rightarrow m} | s_n] &= \sum_{i=n}^m c_i^2 \left( \frac{1}{N} \mathbb{E}[\sigma_i^2 | s_n] + \mathbb{V}[V_i | s_n] \right) \\ &\quad + \sum_{n < i < j \leq m} 2c_i c_j \text{Cov}[\hat{V}_i, \hat{V}_j | s_n]. \end{aligned} \quad (14)$$

To achieve that the compound variance is lower than the variance of MC estimation of current step, i.e.,

$$\mathbb{V}[\hat{V}_n | s_n] > \mathbb{V}[\hat{V}_{n \rightarrow m} | s_n], \quad (15)$$

we need to model the value distribution of future states, and adjust the parameter  $c_i$  heuristically, which is theoretically intractable but can be practically approximated.

## 4.3 One-step Value Distribution Modeling

In this section, we provide a practically tractable approximation method to model the compound variance  $\mathbb{V}[\hat{V}_{n \rightarrow m} | s_n]$  and optimizing value-based process verifier.

Theorem 4.3 implies that for any subsequent states, the covariance should be accounted for due to shared dependencies from previous reasoning steps. Let  $m = n + 1$ ,  $\mathbb{V}[\hat{V}_{n \rightarrow m} | s_n]$  can be simplified as

$$\begin{aligned} \mathbb{V}[\hat{V}_{n \rightarrow m} | s_n] &= \sum_{i=n}^m c_i^2 \left( \frac{1}{N} \mathbb{E}[\sigma_i^2 | s_n] + \mathbb{V}[V_i | s_n] \right) \\ &= c_n^2 \cdot \frac{1}{N} \sigma_n^2 + c_m^2 \cdot \left( \frac{1}{N} \mathbb{E}[\sigma_m^2 | s_n] + \mathbb{V}[V_m | s_n] \right), \end{aligned} \quad (16)$$

and the covariance term is 0 as proved in Appendix. We focus on the case where  $m = n + 1$  in the following part.

**One-step Value Distribution.** First of all, we introduce the definition of one-step value distribution.

**Definition 4.4.** (One-step Value Distribution) Given state  $s_n$  and policy  $\pi$ , the one-step value distribution  $DV_1(s_n)$  is a continuous distribution within the interval  $[0,1]$ . The cumulative distribution function of  $DV_1(s_n)$  satisfies

$$F_X(b) - F_X(a) = \sum_{a_n \in \mathcal{A}'} \pi(a_n | s_n), \quad (17)$$

where  $\mathcal{A}'$  satisfies  $\mathcal{A}' \subseteq \mathcal{A}$  and

$$\begin{aligned} \forall a_n \in \mathcal{A}', Q^\pi(s_n, a_n) &\in (a, b], \\ \forall a_n \in \mathcal{A} - \mathcal{A}', Q^\pi(s_n, a_n) &\notin (a, b]. \end{aligned} \quad (18)$$

$DV_1(s_n)$  measures the distribution of next state value given current state  $s_n$ , whose expectation is exact  $V^\pi(s_n)$ . We can represent the expectation of variance  $\mathbb{E}[\sigma_m^2 | s_n]$  and the variance of expectation  $\mathbb{V}[V_m | s_n]$  in Eq. (16) as follows:

$$\begin{aligned} \mathbb{E}[\sigma_m^2 | s_n] &= \int_0^1 f_X^{(n)}(x) \cdot x(1-x) dx \\ \mathbb{V}[V_m | s_n] &= \int_0^1 f_X^{(n)}(x) \cdot (x - V_n)^2 dx, \end{aligned} \quad (19)$$

where  $X \sim DV_1(s_n)$ ,  $f_X^{(n)}(x)$  is the probability density function of  $DV_1(s_n)$ . However, it is difficult to model the continuous distribution parametrically.

**Categorical Distribution Approximation.** We introduce the categorical distribution  $Z$  as a parametric approximation of the true value distribution  $DV_1(s_n)$  under mild assumptions. We follow the definition of categorical distribution in Farebrother et al. (2024), and project  $DV_1(s_n)$  onto a histogram with bins of width  $\xi = 1/|Z|$ . These bins are centered at  $z_i$ , and the probabilities  $p_i$  for each bin are obtained by integrating over the interval  $[z_i - \xi/2, z_i + \xi/2]$ :

$$p_i = \int_{z_i - \xi/2}^{z_i + \xi/2} f_X^{(n)}(x) dx. \quad (20)$$

The locations  $\{z_i\}_{i=1}^{|Z|}$  are evenly distributed within the interval  $[0,1]$ . The Dirac delta function  $\delta_{z_i}$  is defined as  $\delta_{z_i} = z_i$ .

The categorical distribution is equivalent to  $DV_1(s_n)$  when the number of intervals satisfying  $m \rightarrow \infty$ . When  $m$  is limited, we make the assumption that for any action  $a_i$  whose state-action value  $Q^\pi(s_n, a_i)$  lies within the interval  $[z_i - \xi/2, z_i + \xi/2]$ ,  $Q^\pi(s_t, a_i)$  is approximately equal to the bin center  $z_i$ . This approximation allows us to convert statistical expectations over the continuous value distribution into discrete summations over the categorical bins. Specifically, Eq. (19) can be expressed as:

$$\begin{aligned} \mathbb{E}[\sigma_m^2 | s_n] &= \sum_{i=1}^{|Z|} p_i \cdot z_i(1 - z_i) \\ \mathbb{V}[V_m | s_n] &= \sum_{i=1}^{|Z|} p_i \cdot (z_i - V_n)^2, \end{aligned} \quad (21)$$

and then the variance  $\mathbb{V}[\hat{V}_{n \rightarrow m} | s_n]$  can be estimated. Our goal is to model one-step categorical distribution, as an approximation to the one-step value distribution, which is the optimization objective of the value-based process verifier  $f_\theta$ .

**Practical Verifier Optimization.** We estimate the categorical distribution by assuming it belongs to a specific distribution class, specifically the Gaussian distribution. We provide an empirical analysis about the reasonableness of the Gaussian approximation in Appendix. This assumption allows us to model the distribution effectively with limited sampled data. More precisely, we estimate the distribution's first moment (mean) to be the linear combination of  $\hat{V}^\pi(s_n)$  and  $\hat{V}^\pi(s_{n+1})$ , as introduced in previous section. Subsequently, the distribution's second moment (variance) is estimated by using the difference between  $\hat{V}^\pi(s_{n+1})$  and the calculated first moment for the standard deviation.

To determine the appropriate coefficients that satisfy Eq. (15), we employ a heuristic search over a predefined set of candidate coefficients, where we need to estimate  $\mathbb{V}[\hat{V}_n | s_n]$  and  $\mathbb{V}[\hat{V}_{n \rightarrow n+1} | s_n]$  respectively. Firstly,  $\mathbb{V}[\hat{V}_n | s_n]$  is obtained by treating the MC estimation of  $V^\pi(s_n)$  as an approximation of the true  $V^\pi(s_n)$ , and then applying the operation defined in Eq. (8). Secondly,  $\mathbb{V}[\hat{V}_{n \rightarrow n+1} | s_n]$  is estimated by first deriving  $DV_1(s_n)$  from the output of the value-based process verifier  $f_\theta$ , and then performing the operations defined in Eq. (16) and Eq. (21). After all, the heuristic search will find the coefficients that satisfies Eq. (15). If no coefficient from the predefined set satisfies the criterion, the estimated state value remains to be  $\hat{V}_n$ , aiming at preventing the potential variance increase.

We then proceed to obtain a Gaussian distribution using the updated state value  $\hat{V}^\pi(s_n)$  and the estimated state-action value  $\hat{Q}^\pi(s_n, a_n)$ . This Gaussian distribution is then mapped to the categorical distribution  $Z$  following the methodology as introduced in Eq. (20). The value-based verifier is optimized using a cross-entropy loss function:

$$\mathcal{L}_{ce} = -\mathbb{E}_{(s_n, a_n) \sim \mathcal{D}} \left[ \sum_{i=1}^{|Z|} f_\theta(z_i | s_n, a_n) \log p(z_i | s_n, a_n) \right],$$

where  $p(z_i | s, a)$  is the probability of the categorical distribution  $Z$  at a specific location  $z_i$  and  $f_\theta(z_i | s_n, a_n)$  denotes the output probability of the value-based process verifier at location  $z_i$ . To provide a clear image, we illustrate variance reduction method and practical implementation in Appendix.

## 5 Experiments

### 5.1 Experimental Settings

**Tasks.** We conduct experiments using the test split of two widely used math reasoning datasets: GSM8K (Cobbe et al. 2021) and MATH-500 (Lightman et al. 2024). Besides, we test our method on different base models across different model families: Qwen2.5-Math-7B-Instruct (Yang et al. 2025) and Deepseek-math-7b-instruct (Shao et al. 2024). Following (Wang et al. 2024b), the generator in our experiments is LLemma-7b (Azerbayev et al. 2024).

**Baselines.** For variants trained on different objective functions, we include the return distribution modeling method that trained on BCE loss ( $V_{BCE}$ , (Wang et al. 2024b)), and regression method that trained on MSE loss ( $V_{MSE}$ , (Lu

Method	MATH-500					GSM8K				
	8	16	32	64	128	8	16	32	64	128
Oracle	29.8	35.8	40.4	45.7	49.6	89.4	92.6	94.3	95.9	96.8
<b>Qwen2.5-Math-7B-Instruct</b>										
$V_{MSE}$	24.2	26.8	28.6	29.8	30.6	85.1	87.3	88.5	89.8	90.4
$V_{BCE}$	24.2	26.6	28.4	30.4	31.4	85.6	87.4	88.2	89.5	90.0
$V_{CE}$ w/o ComMCS (ours)	<b>24.6</b>	27.0	<b>29.0</b>	29.4	31.6	85.4	87.6	88.2	89.5	90.3
$V_{CE}$ w/ ComMCS (ours)	24.4	<b>27.2</b>	<b>29.0</b>	<b>30.6</b>	<b>31.8</b>	<b>85.7</b>	<b>88.0</b>	<b>88.6</b>	<b>90.3</b>	<b>91.1</b>
<b>Deepseek-math-7b-instruct</b>										
$V_{MSE}$	21.0	23.2	23.0	23.6	24.4	82.1	82.8	83.5	83.6	83.5
$V_{BCE}$	<b>21.8</b>	23.6	23.6	24.2	25.8	82.8	84.6	84.9	85.4	85.6
$V_{CE}$ w/o ComMCS (ours)	21.0	23.0	23.6	24.8	25.0	83.2	84.5	84.7	85.1	85.4
$V_{CE}$ w/ ComMCS (ours)	21.4	<b>24.4</b>	<b>25.8</b>	<b>26.2</b>	<b>26.6</b>	<b>83.5</b>	<b>85.1</b>	<b>85.1</b>	<b>86.6</b>	<b>86.6</b>

Table 1: Performance of Best-of-N sampling on MATH-500 and GSM8K with different base models. The results are reported as the average accuracy across three random seeds. 8, 16, 32, 64, 128 denote the accuracy with Best-of-8/16/32/64/128, respectively. The oracle results are calculated as whether there is correct answer in the sampled  $N$  answers.  $V_{MSE}$  denotes verifiers trained with mean-squared error loss.  $V_{BCE}$  denotes verifiers trained with soft binary cross entropy loss.  $V_{CE}$  denotes verifiers trained with cross entropy loss, with (w/) or without (w/o) our method (ComMCS). The best results are marked in **bold**. All results are passed with significance test ( $p < .05$ ).

et al. 2024)). For variants trained to modeling value distribution ( $V_{CE}$ ) and with or without the variance reduction technique, we include the results of verifiers trained on cross-entropy loss only. We present more details in Appendix.

**Implementation Details.** We train our generator first, then the generator is used to generate reasoning trajectories to construct the training dataset of the verifiers. Specifically, we construct a dataset of 180,000 samples, each sample is a reasoning trajectory consists of several reasoning steps, and each reasoning step has a value annotation that is estimated via 8 rollouts and their corresponding correctness labels. In the training phase, we use 180,000 sampled solutions to train different verifiers for one epoch with a learning rate set to  $2 \times 10^{-6}$ . More details are provided in Appendix.

**Evaluation Metrics.** Following Lightman et al. (2024); Wang et al. (2024b); Lu et al. (2024), we conduct Best-of-N (BoN) sampling and beam search experiments as an evaluation of our method. We provide more details about evaluation metrics in Appendix.

## 5.2 Results

We present the comparable performances of models trained with different methods for BoN sampling and beam search experiments in Table 1 and Table 2, respectively. Our observations are as follows: **(1) Modeling value distribution is a meaningful replacement to methods that model return distribution, or regression.** The verifier trained on CE loss shows competitive performance in most cases, compared with that trained on BCE and MSE. For instance,  $V_{CE}$  based on Qwen2.5-Math-7B-Instruct outperforms  $V_{MSE}$  and  $V_{BCE}$  on MATH-500 dataset in some cases. We also observe that  $V_{MSE}$  and  $V_{BCE}$  outperforms  $V_{CE}$  based on Deepseek-math-7b-instruct on MATH-500 dataset, showing that the value distribution modeling is not a complete upper-level replacement to modeling return distribution or performing regression. We also observe that  $V_{CE}$

Method	MATH-500	GSM8K
<b>Qwen2.5-Math-7B-Instruct</b>		
$V_{MSE}$	56.0	88.2
$V_{BCE}$	55.2	88.6
$V_{CE}$ w/o ComMCS (ours)	56.4	88.1
$V_{CE}$ w/ ComMCS (ours)	<b>57.8</b>	<b>88.9</b>
<b>Deepseek-math-7b-instruct</b>		
$V_{MSE}$	42.6	81.0
$V_{BCE}$	46.8	84.7
$V_{CE}$ w/o ComMCS (ours)	46.4	84.7
$V_{CE}$ w/ ComMCS (ours)	<b>47.2</b>	<b>84.8</b>

Table 2: Performance of Beam Search sampling on MATH-500 and GSM8K with different base models. The results are reported as the average accuracy across three random seeds. The beam size of our experiments is set as 8.

outperforms  $V_{BCE}$  and  $V_{MSE}$  after performing ComMCS in most cases, showing that the value distribution modeling method can be further improved. **(2) Our variance reduction method improves value distribution modeling without additional LLM inference overhead.** We observe a consistent improvement after applying our variance reduction method when comparing the varieties of  $V_{CE}$  with or without using ComMCS in different tasks and different settings. The results show that the practical approximations we introduced in §4.3 is tolerable. We note that a similar performance for  $V_{CE}$  varieties based on Qwen2.5-Math-7B-Instruct on MATH-500 dataset. We contribute the phenomenon as the limited accuracy of our distribution approximation and heuristic search method regarding to the coefficients, and thus there is still room for further optimization in terms of value distribution approximation. Similar results can also be observed in the beam search experiments.

CoefficientSetting	MATH-500				
	8	16	32	64	128
Oracle	29.8	35.8	40.4	45.7	49.6
<b>Deepseek-math-7b-instruct</b>					
Static (0.9)	21.0	23.2	<b>25.8</b>	25.8	25.8
Static (0.99)	<b>21.4</b>	23.8	25.2	25.8	25.6
Static (1.0)	21.0	23.0	23.6	24.8	25.0
Dynamic	<b>21.4</b>	<b>24.4</b>	<b>25.8</b>	<b>26.2</b>	<b>26.6</b>

Table 3: Performance of Best-of-N sampling on MATH-500 with different coefficient settings. The results are reported as the average accuracy across three random seeds. ‘‘Static’’ denotes using a fixed coefficient during the training loop, while ‘‘Dynamic’’ denotes using different coefficient that is decided by the variance comparison during the training loop.

Method	$\sigma$	MATH-500				
		8	16	32	64	128
<b>Deepseek-math-7b-instruct</b>						
$V_{CE}$	1	21.0	23.0	23.6	24.8	25.0
$V_{CE}$ w/ ComMCS	1	<b>21.4</b>	<b>24.4</b>	<b>25.8</b>	<b>26.2</b>	<b>26.6</b>
$V_{CE}$	2	21.6	<b>24.2</b>	24.6	24.8	24.6
$V_{CE}$ w/ ComMCS	2	<b>22.0</b>	23.0	<b>25.2</b>	<b>25.4</b>	<b>25.2</b>
$V_{CE}$	3	21.6	24.2	24.6	25.4	24.8
$V_{CE}$ w/ ComMCS	3	<b>21.8</b>	<b>24.8</b>	<b>25.6</b>	<b>25.8</b>	<b>25.3</b>

Table 4: Performance of Best-of-N sampling on MATH-500 with different scales. The results are reported as the average accuracy across three random seeds.  $\Delta$  denotes the difference between the MC estimation of  $V^\pi(s_n)$  (for static methods) or the updated estimation of  $V^\pi(s_n)$  (for dynamic methods) and the MC estimation of  $Q^\pi(s_n, a_n)$ .

## 6 Analysis and Discussions

### 6.1 Comparison between Static and Dynamic Coefficients

We conduct a further analysis of the coefficients of linear combination in our method. Specifically, we compare different coefficient selection strategies, including static coefficients ranging from  $\{0.9, 0.99, 1.0\}$  and dynamic coefficients that are derived from heuristic search as introduced in Appendix. We report the BoN results in Table 3. We observe that the verifier trained with dynamic coefficient shows a consistent improvement compared with those trained with static methods. In contrast, the performance between different static coefficients varies. We conclude that: **(1) Different estimation variance can result in different performances.** Although all the labels of the training dataset were obtained through unbiased estimation, the estimation variances in different experiments were not the same. The varying variances directly lead to the differences in experimental results. **(2) The optimal coefficient is not static.** Our experimental results show that the performance of models trained with different static coefficients can produce mutually competitive results under different experimental conditions. As implied by Theorem 4.3, the variance of linear combination of sev-

eral reasoning steps is influenced by the value distributions of each states and the coefficients. Given the coefficient to be static, it cannot take into account the differences between value distributions and thus cannot result in a minimal estimation variance for different value estimations. **(3) The dynamic coefficient works.** Through variance modeling and variance comparison, our distribution modeling and variance estimation method can help to achieve a better estimation while maintaining the condition of unbiased estimation.

### 6.2 Comparison between Different Value Distribution Hypothesis

In the main experiments, we estimate Gaussian distribution by setting the variance to be the difference between  $\hat{V}^\pi(s_{n+1})$  and calculated first moment. In this section, we conduct a further analysis of our distribution hypothesis and the generalization ability of our method under different distributions. Specifically, we regard the difference between  $\hat{V}^\pi(s_n)$  and  $\hat{Q}^\pi(s_n, a_n)$  to be one, two or three standard deviations, which means that 68%, 95% or 99.7% of all sampled values are within the range of  $[\hat{V}^\pi(s_n) - \hat{Q}^\pi(s_n, a_n), \hat{V}^\pi(s_n) + \hat{Q}^\pi(s_n, a_n)]$ . When more standard deviations are used for estimation, the sampled  $\hat{Q}^\pi(s_n, a_n)$  will be treated as a value that is more distant from distribution center, which makes the estimated distribution acute.

We report the BoN results in Table 4. We find that after variance reduction, the verifiers can achieve better performances compared with their baselines that trained without variance reduction in most cases. The result shows that under different distribution hypothesis, our method can provide a stable improvement without additional LLM inference cost, which reveals the generalization ability and solidness of our method.

## 7 Conclusion and Discussion

In this work, We introduced ComMCS, a theoretically-grounded method that reduces the estimation variance when performing MC estimations for the training annotations of value-base process verifiers without additional LLM inference cost. Utilizing the linear combination of MC estimations of current step and its subsequent step, we refine the state value of current step heuristically while maintaining the unbiased property of MC estimation. Our experiments demonstrate the effectiveness of our method across various math reasoning tasks, outperforming existing value-based verifier optimization methods like MSE and BCE. Through detailed analysis, we highlight the effect of variance reduction and variance modeling. We also note that there are some potential limitations of our method. Firstly, our method relies on the Gaussian distribution hypothesis. While effective in practice, our method may not hold for all tasks or distributions. Secondly, applying ComMCS to other reasoning domains, such as code generation, presents an exciting avenue for research. We hope that our approach contributes valuable insights to the field of MC estimation optimization and value-based process verifier optimization.

## Acknowledgements

We thank reviewers and ACs for their comments. This work is supported by the National Natural Science Foundation of China (Grant No. 62406088) and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2025A1515011376) and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024B0101050003) and the Shenzhen Science and Technology Program (Grant No. ZDSYS20230626091203008).

## References

- Azerbayev, Z.; Schoelkopf, H.; Paster, K.; Santos, M. D.; McAleer, S. M.; Jiang, A. Q.; Deng, J.; Biderman, S.; and Welleck, S. 2024. Llemma: An Open Language Model for Mathematics. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Chen, W.; He, W.; Xi, Z.; Guo, H.; Hong, B.; Zhang, J.; Li, N.; Gui, T.; Li, Y.; Zhang, Q.; and Huang, X. 2025. Better Process Supervision with Bi-directional Rewarding Signals. In *ACL (Findings)*, 14471–14485. Association for Computational Linguistics.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Srvankumar, A.; Korenev, A.; Hinsvark, A.; Rao, A.; Zhang, A.; Rodriguez, A.; Gregerson, A.; Spataru, A.; Rozière, B.; Biron, B.; Tang, B.; Chern, B.; Caucheteux, C.; Nayak, C.; Bi, C.; Marra, C.; McConnell, C.; Keller, C.; Touret, C.; Wu, C.; Wong, C.; Ferrer, C. C.; Nikolaidis, C.; Allonsius, D.; Song, D.; Pintz, D.; Livshits, D.; Esiobu, D.; Choudhary, D.; Mahajan, D.; Garcia-Olano, D.; Perino, D.; Hupkes, D.; Lakomkin, E.; AlBadawy, E.; Lobanova, E.; Dinan, E.; Smith, E. M.; Radenovic, F.; Zhang, F.; Synnaeve, G.; Lee, G.; Anderson, G. L.; Nail, G.; Mialon, G.; Pang, G.; Cucurell, G.; Nguyen, H.; Korevaar, H.; Xu, H.; Touvron, H.; Zarov, I.; Ibarra, I. A.; Kloumann, I. M.; Misra, I.; Evtimov, I.; Copet, J.; Lee, J.; Geffert, J.; Vranes, J.; Park, J.; Mahadeokar, J.; Shah, J.; van der Linde, J.; Billock, J.; Hong, J.; Lee, J.; Fu, J.; Chi, J.; Huang, J.; Liu, J.; Wang, J.; Yu, J.; Bitton, J.; Spisak, J.; Park, J.; Rocca, J.; Johnstun, J.; Saxe, J.; Jia, J.; Alwala, K. V.; Upasani, K.; Plawiak, K.; Li, K.; Heafield, K.; Stone, K.; and et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.
- Farebrother, J.; Orbay, J.; Vuong, Q.; Taïga, A. A.; Chebotar, Y.; Xiao, T.; Irpan, A.; Levine, S.; Castro, P. S.; Faust, A.; Kumar, A.; and Agarwal, R. 2024. Stop Regressing: Training Value Functions via Classification for Scalable Deep RL. In *ICML*. OpenReview.net.
- Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.
- Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; Madry, A.; Baker-Whitcomb, A.; Beutel, A.; Borzunov, A.; Carney, A.; Chow, A.; Kirillov, A.; Nichol, A.; Paino, A.; Renzin, A.; Passos, A. T.; Kirillov, A.; Christakis, A.; Conneau, A.; Kamali, A.; Jabri, A.; Moyer, A.; Tam, A.; Crookes, A.; Tootoonchian, A.; Kumar, A.; Vallone, A.; Karpathy, A.; Braunstein, A.; Cann, A.; Codispoti, A.; Galu, A.; Kondrich, A.; Tulloch, A.; Mishchenko, A.; Baek, A.; Jiang, A.; Pelisse, A.; Woodford, A.; Gosalia, A.; Dhar, A.; Pantuliano, A.; Nayak, A.; Oliver, A.; Zoph, B.; Ghorbani, B.; Leimberger, B.; Rossen, B.; Sokolowsky, B.; Wang, B.; Zweig, B.; Hoover, B.; Samic, B.; McGrew, B.; Spero, B.; Giertler, B.; Cheng, B.; Lightcap, B.; Walkin, B.; Quinn, B.; Guarraci, B.; Hsu, B.; Kellogg, B.; Eastman, B.; Lugaresi, C.; Wainwright, C. L.; Bassin, C.; Hudson, C.; Chu, C.; Nelson, C.; Li, C.; Shern, C. J.; Conger, C.; Barette, C.; Voss, C.; Ding, C.; Lu, C.; Zhang, C.; Beaumont, C.; Hallacy, C.; Koch, C.; Gibson, C.; Kim, C.; Choi, C.; McLeavey, C.; Hesse, C.; Fischer, C.; Winter, C.; Czarnecki, C.; Jarvis, C.; Wei, C.; Koumouzelis, C.; and Sherburn, D. 2024. GPT-4o System Card. *CoRR*, abs/2410.21276.
- Khalifa, M.; Agarwal, R.; Logeswaran, L.; Kim, J.; Peng, H.; Lee, M.; Lee, H.; and Wang, L. 2025. Process Reward Models That Think. *CoRR*, abs/2504.16828.
- Li, Y.; Lin, Z.; Zhang, S.; Fu, Q.; Chen, B.; Lou, J.-G.; and Chen, W. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5315–5333.
- Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2024. Let’s Verify Step by Step. In *ICLR*. OpenReview.net.
- Liu, R.; Gao, J.; Zhao, J.; Zhang, K.; Li, X.; Qi, B.; Ouyang, W.; and Zhou, B. 2025. Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. *CoRR*, abs/2502.06703.
- Lu, J.; Dou, Z.; Wang, H.; Cao, Z.; Dai, J.; Feng, Y.; and Guo, Z. 2024. AutoPSV: Automated Process-Supervised Verifier. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; Tang, Y.; and Zhang, D. 2025. Wizard-Math: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Luo, L.; Liu, Y.; Liu, R.; Phatale, S.; Lara, H.; Li, Y.; Shu, L.; Zhu, Y.; Meng, L.; Sun, J.; and Rastogi, A. 2024. Improve Mathematical Reasoning in Language Models by Automated Process Supervision. *CoRR*, abs/2406.06592.
- OpenAI. 2025. o3-mini System Card. Accessed: 2025-01-31.

- Qin, C.; Zhang, A.; Chen, C.; Dagar, A.; and Ye, W. 2024. In-Context Learning with Iterative Demonstration Selection. In *EMNLP (Findings)*, 7441–7455. Association for Computational Linguistics.
- Setlur, A.; Garg, S.; Geng, X.; Garg, N.; Smith, V.; and Kumar, A. 2024. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37: 43000–43031.
- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *CoRR*, abs/2402.03300.
- Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 8634–8652.
- Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *CoRR*, abs/2408.03314.
- Snell, C. V.; Lee, J.; Xu, K.; and Kumar, A. 2025. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *ICLR*. OpenReview.net.
- Uesato, J.; Kushman, N.; Kumar, R.; Song, H. F.; Siegel, N. Y.; Wang, L.; Creswell, A.; Irving, G.; and Higgins, I. 2022. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275.
- Wang, J.; Fang, M.; Wan, Z.; Wen, M.; Zhu, J.; Liu, A.; Gong, Z.; Song, Y.; Chen, L.; Ni, L. M.; Yang, L.; Wen, Y.; and Zhang, W. 2024a. OpenR: An Open Source Framework for Advanced Reasoning with Large Language Models. *CoRR*, abs/2410.09671.
- Wang, P.; Li, L.; Chen, L.; Song, F.; Lin, B.; Cao, Y.; Liu, T.; and Sui, Z. 2023. Making Large Language Models Better Reasoners with Alignment. *CoRR*, abs/2309.02144.
- Wang, P.; Li, L.; Shao, Z.; Xu, R.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2024b. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In *ACL (1)*, 9426–9439. Association for Computational Linguistics.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Wu, Z.; Hu, Y.; Shi, W.; Dziri, N.; Suhr, A.; Ammanabrolu, P.; Smith, N. A.; Ostendorf, M.; and Hajishirzi, H. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36: 59008–59033.
- Xi, Z.; Chen, W.; Hong, B.; Jin, S.; Zheng, R.; He, W.; Ding, Y.; Liu, S.; Guo, X.; Wang, J.; Guo, H.; Shen, W.; Fan, X.; Zhou, Y.; Dou, S.; Wang, X.; Zhang, X.; Sun, P.; Gui, T.; Zhang, Q.; and Huang, X. 2024. Training Large Language Models for Reasoning through Reverse Curriculum Reinforcement Learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024a. Qwen2 Technical Report. *CoRR*, abs/2407.10671.
- Yang, A.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Huang, H.; Jiang, J.; Tu, J.; Zhang, J.; Zhou, J.; Lin, J.; Dang, K.; Yang, K.; Yu, L.; Li, M.; Sun, M.; Zhu, Q.; Men, R.; He, T.; Xu, W.; Yin, W.; Yu, W.; Qiu, X.; Ren, X.; Yang, X.; Li, Y.; Xu, Z.; and Zhang, Z. 2025. Qwen2.5-1M Technical Report. *CoRR*, abs/2501.15383.
- Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; Lu, K.; Xue, M.; Lin, R.; Liu, T.; Ren, X.; and Zhang, Z. 2024b. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *CoRR*, abs/2409.12122.
- Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *ICLR*. OpenReview.net.
- Zhang, L.; Hosseini, A.; Bansal, H.; Kazemi, M.; Kumar, A.; and Agarwal, R. 2025a. Generative Verifiers: Reward Modeling as Next-Token Prediction. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025b. The Lessons of Developing Process Reward Models in Mathematical Reasoning. In *ACL (Findings)*, 10495–10516. Association for Computational Linguistics.
- Zhao, J.; Liu, R.; Zhang, K.; Zhou, Z.; Gao, J.; Li, D.; Lyu, J.; Qian, Z.; Qi, B.; Li, X.; and Zhou, B. 2025. GenPRM: Scaling Test-Time Compute of Process Reward Models via Generative Reasoning. *CoRR*, abs/2504.00891.
- Zheng, C.; Zhang, Z.; Zhang, B.; Lin, R.; Lu, K.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2024. ProcessBench: Identifying Process Errors in Mathematical Reasoning. *CoRR*, abs/2412.06559.
- Zhou, H.; Nova, A.; Larochelle, H.; Courville, A. C.; Neyshabur, B.; and Sedghi, H. 2022. Teaching Algorithmic Reasoning via In-context Learning. *CoRR*, abs/2211.09066.