

# SPARE: Single-Pass Annotation with Reference-Guided Evaluation for Automatic Process Supervision and Reward Modelling

Md Imbesat Hassan Rizvi<sup>1</sup>, Xiaodan Zhu<sup>2</sup>, Iryna Gurevych<sup>1</sup>

<sup>1</sup>Ubiquitous Knowledge Processing Lab (UKP Lab), Department of Computer Science and Hessian Center for AI (hessian.AI), Technical University of Darmstadt, Germany

<sup>2</sup>Department of Electrical and Computer Engineering & Ingenuity Labs Research Institute, Queen’s University, Canada

www.ukp.tu-darmstadt.de, xiaodan.zhu@queensu.ca

## Abstract

Process or step-wise supervision has played a crucial role in advancing complex multi-step reasoning capabilities of Large Language Models (LLMs). However, efficient, high-quality automated process annotation remains a significant challenge. To address this, we introduce **Single-Pass Annotation with Reference-Guided Evaluation (SPARE)**, a novel structured framework that enables efficient per-step annotation by jointly aligning solution steps to reference solutions and determine its accuracy with explicit reasoning in single generation. We demonstrate SPARE’s effectiveness across four diverse datasets spanning mathematical reasoning (GSM8K, MATH), multi-hop question answering (MuSiQue-Ans), and spatial reasoning (SPaRP), showing consistent improvements in two applications: (1) training Process Reward Models (PRMs) for ranking and aggregating multiple generations, and (2) fine-tuning models via offline reinforcement learning for greedy decoding. On PROCESSBENCH, SPARE demonstrates data-efficient out-of-distribution generalization, using only  $\sim 16\%$  of training samples compared to human-labeled and other synthetically trained baselines. Additionally, it achieves competitive performance with MCTS-based methods while offering  $2.3\times$  speedup in terms of total token count. Manual analysis reveals complementary precision-recall characteristics with MCTS approaches, suggesting potential for ensemble methods. These results establish SPARE as a practical and scalable solution for automatic process supervision in LLM reasoning.

**Process Reward Models (SPARE-PRM)** —

<https://huggingface.co/collections/UKPLab/spare-prm>

**Code** — <https://github.com/UKPLab/aaai2026-spare-prm>

**Extended version with Appendices** —

<https://www.arxiv.org/abs/2506.15498>

## 1 Introduction

While large language models (LLMs) have demonstrated strong performance across a broad range of tasks (Brown et al. 2020; Wei et al. 2022a,b; Chowdhery et al. 2023; Touvron et al. 2023; BigBench-Team 2023), complex multi-step reasoning still remains a challenge for LLMs even when they are trained and finetuned with ground-truth chains of

thoughts (Azerbayev et al. 2024; Yu et al. 2024). Self-consistency can improve performance by voting over multiple generations, only if the answers are correct in majority of them. To address this, reward models trained to assess output correctness have gained popularity. Outcome Reward Models (ORMs) (Cobbe et al. 2021; Yu, Gao, and Wang 2024) are trained using outcome supervision relying on the correctness of the final answer, while Process Reward Models (PRMs) (Uesato et al. 2022; Lightman et al. 2024) use process supervision that relies on the correctness of individual reasoning steps.

PRMs achieve better performance due to the targeted step-level feedback but suffer from expensive and complex annotation requirements. Human-supervision (Uesato et al. 2022; Lightman et al. 2024) is very demanding in terms of highly skilled human evaluators, motivating efforts toward automatic process annotation largely driven by Monte Carlo Tree Search (MCTS)-based methods (Wang et al. 2024a,b; Luo et al. 2024; Zhang et al. 2024). In MCTS-based approaches, models are initially trained on ground-truth reasoning traces and answers through supervised fine-tuning. However, during step evaluation, these methods overlook the valuable step-by-step information *already present* in the reference ground-truth rationales. Instead, they rely exclusively on final answer matching across multiple model roll-outs, resulting in both computational inefficiency and *underutilization* of the data already available at hand.

Parallel efforts aim to leverage valuable signals from reference reasoning traces, that are either *existing* ground-truth or *synthetically* generated rationales. For instance, Li et al. (2023); Khalifa et al. (2023) generated step-level annotations by decomposing candidate and reference solutions into individual steps, performing alignment using dataset-specific discriminative models, and annotating steps in a restricted context where a candidate step is matched to a single reference step. AutoPRM (Chen et al. 2024) decomposes reference solutions into sub-questions and corresponding solutions to enable process supervision. However, their approach relies on an auxiliary model for data collection, which is trained using outputs from a more capable language model. More recently, GenRM (Zhang et al. 2025a) employed reference-guided grading to train verifiers using synthetically generated rationales as references. However, GenRM is not a process supervision (PRM)

model and relies on rationales from a more capable model than the one trained as the reward model. More recently, ThinkPRM (Khalifa et al. 2025) and R-PRM (She et al. 2025) utilized a more capable model to generate synthetic verification rationales, which were subsequently filtered to retain only those whose step annotations aligned with human-labeled steps in the PRM800K dataset (Lightman et al. 2024). While these approaches maximize the utility of available data, they depend on step labels for initialization or filtering, labels that are the end goal of process supervision itself and may not exist for other datasets or domains.

To address these gaps, we propose **Single-Pass Annotation with Reference-Guided Evaluation (SPARE)**, a general-purpose framework that enables automatic process supervision through step-level evaluation of model responses by leveraging intermediate steps from reference reasoning traces. SPARE strikes a balance between leveraging available data and maintaining broad applicability across domains, using only a single model throughout the process. Concretely, it introduces a generic, structured evaluation scheme that (i) emphasizes explicit reasoning during step evaluation, and (ii) supports multi-step alignment between model outputs and references. It accomplishes this by encoding instance-agnostic alignment and evaluation criteria in the system prompt, complemented by in-context exemplars that illustrate how to apply these guidelines to specific instances. This design enables single-pass evaluation with additive scaling relative to the token lengths of the response and reference. In summary, our contributions are:

- We introduce SPARE, a general, single-pass, and structured reference-guided evaluation framework for process annotation, which emphasizes explicit reasoning and multi-step alignment. Notably, SPARE is agnostic to the source of reference solutions, assuming they are of high quality. In this work, we reuse existing reference solutions from standard supervised fine-tuning (SFT) datasets without requiring additional reasoning traces.
- We utilize SPARE annotations to improve LLM reasoning via: (i) training Reward Models (RMs) for ranking and aggregating multiple generations, and (ii) fine-tuning models in an offline reinforcement learning (RL) setup for greedy-decoding during inference.
- We evaluate SPARE across four benchmarks—GSM8K, MATH, MuSiQue-Ans, and SpARP—demonstrating consistent improvements over baselines, and out-of-distribution generalization on PROCESSBENCH with high data efficiency. It substantially reduces annotation cost versus tree-search methods, with manual analysis revealing complementary precision-recall characteristics.

## 2 Related Work

**Reasoning abilities of LLMs.** Reasoning remains a challenging area for the Large Language Models (LLMs). Various prompting techniques, such as chain-of-thought, few-shot prompting and their variants (Wei et al. 2022b; Kojima et al. 2022; Yao et al. 2023; Hao et al. 2023) elicited reasoning capabilities in LLMs. Importance of individual steps while prompting (Fu et al. 2023; Zhou et al. 2023) was soon

found to be crucial in successfully solving multi-step reasoning problems. While prompt-only techniques show promising results, their performances are constrained by and sensitive to prompt design and nature of tasks (Ye and Durrett 2022). Consequently, explicitly finetuning with high-quality reasoning traces for improving LLM reasoning capabilities has become popular (Yu et al. 2024; Luo et al. 2025).

**Outcome and Process Supervision.** Supervised finetuning quickly results in saturation, leading to the search for other advanced techniques and better supervision signals. Outcome supervision (Cobbe et al. 2021; Yu, Gao, and Wang 2024) relies on signal based on the final answer, and hence, is easier to obtain. Process supervision offers advantages in the form of fine-grained feedback from individual reasoning steps, however, early work (Uesato et al. 2022; Pan et al. 2023; Lightman et al. 2024) relied on time-consuming and costly human annotation. To alleviate this problem, several recent approaches have emerged for automating process supervision. Monte-Carlo Tree Search (MCTS) based approaches (Wang et al. 2024a,b; Luo et al. 2024; Zhang et al. 2024) target obtaining process annotation by several continuations from intermediate steps whose correctness are evaluated based on the final step. Parallel work has explored leveraging reference reasoning traces, either ground-truth or synthetic, for supervision. Prior approaches decompose solutions into steps for alignment (Li et al. 2023; Khalifa et al. 2023), often relying on dataset-specific models with limited generalization. Others, like AutoPRM (Chen et al. 2024), use sub-question decomposition but depend on auxiliary models trained with outputs from stronger LLMs. GenRM (Zhang et al. 2025a), ThinkPRM (Khalifa et al. 2025), and R-PRM (She et al. 2025) use synthetic rationales from more capable models for training verifiers, but do not provide a general-purpose, reference-guided process supervision framework. In contrast to these efforts, our work proposes a unified, single-pass, and structured evaluation framework for automatic process annotation, enabling flexible alignment and multi-step comparison with reference solutions. We further demonstrate its effectiveness across both fine-tuning and verification settings.

## 3 Our Approach

### 3.1 Single-Pass Annotation with Reference-Guided Evaluation (SPARE)

We propose **Single-Pass Annotation with Reference-Guided Evaluation (SPARE)** as a unified LLM-driven framework for fine-grained evaluation of model-generated reasoning steps with respect to a given context and reference solution. SPARE *jointly* infers with explicit reasoning (i) the alignment of each output step with relevant context and reference steps, and (ii) its correctness label. Concretely, given:

- a reference reasoning path  $\mathcal{R} = \{r\}_{i=1}^m$  (with  $m$  steps),
- a model-generated output  $\mathcal{O} = \{o\}_{i=1}^n$  (with  $n$  steps),
- a contextual question  $\mathcal{C} = \{c\}_{i=1}^s$  (with  $s$  sentences), and
- a system prompt  $\mathcal{S}$  defining evaluation guidelines,

An answer or outcome annotation  $y \in \mathbb{R}$  is a score indicating a measure of correctness of the model’s output. Most commonly,  $y = \mathbb{I}(o_n = r_m)$ ; i.e., the output’s answer matches with the reference reasoning answer. In contrast, a process annotation  $\mathcal{Y} = \{y \mid y \in \mathbb{R}\}_{i=1}^n$  is a sequence of scalar scores assigned to the corresponding steps  $o_i$ .

We devise an evaluation sequence  $\mathcal{E} = \{\varepsilon\}_{i=1}^n$  where each step  $o_i$  is annotated with alignment and correctness information, such that each  $\varepsilon_i$  consists of a structured tuple:

$$\varepsilon = (e, c^+, o^+, r^+, \epsilon, y_i)$$

where  $e$  is a natural language explanation justifying the evaluation  $y_i \in \{-1, +1\}$ , while referring to:

- $c^+ \subseteq \mathcal{C}$  is the subset of context sentences relevant to  $o_i$ ,
- $o^+ \subseteq \mathcal{O} \setminus o_i$  contains other output steps related to  $o_i$ ,
- $r^+ \subseteq \mathcal{R}$  are the reference steps relevant to  $o_i$ ,
- $\epsilon$  is a (possibly empty) list of error categories.

**Joint Alignment and Evaluation via In-Context Learning (ICL).** The core innovation of SPARE lies in its single-pass framework that leverages the reasoning and evaluative capabilities of large language models (LLMs) through In-Context Learning (ICL) to *jointly* infer, first the step alignment and then the step correctness, in a single generation. This approach parallels Natural Language Inference (NLI) with evidence localization, i.e., not merely determining whether a hypothesis (or a key fact) is entailed by a premise (or a document), but also identifying which textual components support that entailment. Similar strategies have proven effective in fine-grained summarization evaluation (Song et al. 2024).

We extend this paradigm of localized evidence to multi-step reasoning evaluation by enabling accurate and context-sensitive step alignment through:

- **Relevancy directives** – Terms, concepts or natural language descriptions embedded in the system prompt ( $\mathcal{S}$ ), which guide the LLM in evaluating the relevancy between two steps, say  $o_i$  and  $r_j$ . These directives are broadly applicable across instances and include criteria such as semantic overlap, computational or numeric similarity, entity or variable consistency, and structural or format alignment.
- **Few-shot exemplars** that ground the instance-agnostic generic alignment criteria to instance-specific alignment explanation. The exemplars are created highlighting reasoning across varying granularity and surface forms, including both single- and multi-step alignment scenarios.

Conditioned on these instructions and exemplars, the LLM is prompted to jointly:

- Reason and identify the aligned step(s) in the reference solution  $\mathcal{R}$ , context sentences  $\mathcal{C}$ , or peer output steps in  $\mathcal{O}$  for each generated step  $o_i$ .
- Reason and explain the correctness label  $y_i$  of the aligned step, optionally specifying the error type (e.g., NUMERIC, COMPREHENSION) when applicable.

**Alignment Scenarios.** To accommodate differences in reasoning granularity ( $n \neq m$ ), we incorporate detailed guidance in the system prompt and design few-shot exemplars that capture the following flexible alignment scenarios:

1. *One-to-one* – Most simple alignment where one output step aligns directly and completely with *at most* one step, making it sufficient for evaluation. The alignment can take one of the forms: (i) a single reference reasoning step ( $o_i \mapsto r_j$ ), (ii) a single context sentence ( $o_i \mapsto c_k$ ), (iii) follows directly from or complements another output step ( $o_i \mapsto o_l$ ), or (iv) no alignment at all ( $o_i \mapsto \emptyset$ ).
2. *One-to-many* – An output step requires alignment with *at least* two steps for its evaluation. Such an alignment is *necessarily* required for:
  - i) *Composite output steps* – The model output step  $o_i$  omits minor intermediary steps or merges multiple steps into one. Its correctness must be evaluated against multiple reference steps  $r_j$  and  $c_k$ .
  - ii) *Composite reference steps* – The model output step  $o_i$  is simple while reference steps are composite. Its correctness must be evaluated in conjunction with at least one other output step  $o_l$  and at least one reference step  $r_j$  or context sentence  $c_k$ .

In summary, our SPARE framework defines step correctness through LLM-mediated joint alignment and evaluation, where steps are contextualized within the broader reasoning structure through explicit reference to supporting evidence. Combined with structured explanations, SPARE accommodates surface form variations without penalizing alternate but valid solution paths, enabling step-level automatic process supervision. A complete example for the MATH dataset, including system prompt, LLM output, and example failure modes are shown in Appendix A. While LLM-based approaches may inherit model biases and errors, we mitigate these in SPARE through broad relevancy directives and diverse few-shot exemplars that balance correctness, reasoning diversity, and topic coverage. We note that annotation quality depends on reference solution quality; we therefore evaluate SPARE with existing clean references to establish its efficacy. For noisy or synthetically obtained references, we expect errors to remain localized to affected steps due to SPARE’s local multi-step alignment, with robustness further improvable through cross-reference consistency checks. We leave experimentation under noise, as well as extensions to multilingual and multimodal contexts, for future work.

### 3.2 Training Approach

#### SPARE-based Process Reward Model (SPARE-PRM)

We utilize the step-level evaluations  $y_i$  obtained through SPARE as direct reward signals to train process reward models. The SPARE-PRM is trained in a stepwise classification

Dataset	Aggregation / Ranking					Model Finetuning			
	SC	ORM (BoN)	ORM + SC	SPARE (BoN)	SPARE + SC	SFT-1	SFT-2	SFT-1 + Out.	SFT-1 + SPARE
GSM8K	74.9	79.7	79.8	<u>80.0</u>	<b>80.3</b>	<b>70.4</b>	<b>70.4</b>	69.0	<u>69.8</u>
MATH*	23.4	20.2	<u>23.8</u>	20.9	<b>24.1</b>	21.2	22.1	<u>23.1</u>	<b>23.4</b>
MuSiQue-Ans	19.7 / 25.2	33.4 / 45.4	34.8 / 44.5	<b>34.9 / 45.5</b>	32.1 / 40.4	23.6 / 32.5	26.3 / 35.1	<u>38.2 / 49.9</u>	<b>38.9 / 50.5</b>
SpaRP-S	25.4 / 34.4	<u>41.7 / 49.8</u>	<u>41.7 / 49.8</u>	<b>43.7 / 50.0</b>	39.6 / 46.9	23.2 / 35.0	<u>39.9 / 47.1</u>	<u>39.2 / 49.8</u>	<b>40.1 / 51.0</b>

Table 1: Llama-3 8B Instruct performance. **Bold** means best; underline means second-best. **Aggregation/Ranking** on N=20 generations from Llama-3 8B SFT iteration 1. **SC** means Self-Consistency, **BoN** means Best-of-N sampling. Metrics averaged over 3 independent runs. **Finetuning** results use greedy decoding. \* indicates BoN / SC results reported on MATH-500.

setting, using the following cross-entropy loss:

$$\mathcal{L}_{PRM} = - \sum_{i=1}^n \left( y_i \log \sigma(r_\theta(\mathcal{C}, o_{1:i})) + (1 - y_i) \log (1 - \sigma(r_\theta(\mathcal{C}, o_{1:i}))) \right) \quad (1)$$

where  $o_{1:i}$  is the sub-sequence of output  $\mathcal{O}$  till the  $i^{th}$  step. Unlike ORMs which predict a single solution score for  $\mathcal{O}$ , PRMs generate a probability sequence  $\mathcal{P} = \{p_i\}_{i=1}^n$  for each step  $o_i \in \mathcal{O}$ . While min and prod aggregation are commonly used (Lightman et al. 2024; Wang et al. 2024a), we adopt the last function to aggregate step-wise probabilities, following recent findings (Wang et al. 2024b; Zhang et al. 2025b), as it yields superior downstream performance.

**SPARE-based Finetuning (SPARE-ORPO)** We propose SPARE-based fine-tuning to enhance model reasoning capabilities. The step-by-step process annotations  $\mathcal{Y} = \{y_i\}_{i=1}^n$ , derived using the SPARE framework, can be effectively integrated with both online and offline Reinforcement Learning (RL). For ease of implementation, training stability, and resource efficiency, we use Odds Ratio Preference Optimization (ORPO) for preference training over *chosen* and *rejected* pairs ( $\mathcal{O}_w, \mathcal{O}_l$ ).

In SPARE-ORPO, we compute mean step annotation  $\bar{y} = \frac{1}{n} \sum y_i$  as reasoning score and combine it with answer correctness  $y$  to form score tuple  $(y, \bar{y})$  for preference pairs, where  $y_w = 1$ ,  $y_l = -1$ , and  $\bar{y}_w > \bar{y}_l$ . Thus chosen solutions exceed rejected solutions in both reasoning quality and answer accuracy. Conversely, Outcome-ORPO uses preference pairs ( $\mathcal{O}_w, \mathcal{O}_l$ ) based solely on answer correctness, i.e.,  $y_w = 1$  and  $y_l = -1$ .

## 4 Experiment Results

### 4.1 Experimental Set-up

**Datasets.** We conduct extensive experiments over a suite of reasoning datasets<sup>1</sup>:

- **Mathematical Reasoning.** We use two mathematical datasets, GSM8K (Cobbe et al. 2021), which is a collection of grade school math word problems, and

<sup>1</sup>We use 90:10 train/dev splits for GSM8K and MATH, and 80:20 for MuSiQue-Ans, as these lack official dev-sets.

Model	SC	ORM+SC	PRM+SC	$\Delta$ RM
SPARE-Llama3-8B:				
N=20	23.4	23.8	24.1	0.3
N=256	30.5	31.2	32.4	1.2
SPARE-Qwen2.5-3B (N=20):				
Qwen2.5-3B Gen.	31.4	33.8	34.6	0.8
Qwen2.5-3B Gen. <sup>†</sup>	66.6	67.6	68.8	1.2
Qwen2.5-32B Gen.	64.6	65.6	66.0	0.4
MS-Mistral-7B:				
SFT Gen.	35.1	38.0	38.3	0.3
Process RL Gen.	42.3	43.1	43.5	0.4
MS-DeepSeek-67B	45.4	47.0	48.1	1.1
R-MCTS*-Mistral-7B	35.1	38.0	39.0	1.0

Table 2: SPARE performance across setups contextualized against PRMs with comparable training paradigms. N: generation count; MS: Math-Shepherd; R-MCTS\*: Rest-MCTS\*. Results for external models from their publications at N=256 generations. Results with <sup>†</sup> reported on generation length of 2048 using pre-trained model.

MATH (Hendrycks et al. 2021), which contains high school competition-level math problems across seven diverse topics. Following standard practice in the verification setting, we use the MATH-500 subset (Lightman et al. 2024) for test-time evaluation involving multiple generations.

- **Question-Answering.** We use MuSiQue-Ans dataset (Trivedi et al. 2022), a challenging multi-hop question-answering dataset constructed by composing six diverse reasoning graphs of sub-questions from five different sources.
- **Spatial Reasoning.** We use the small SpaRP (Rizvi, Zhu, and Gurevych 2024), i.e., SpaRP-S dataset, which comprises four textual spatial reasoning sub-datasets covering various spatial characterizations. SpaRP requires spatial relation composition to deduce relations between two objects when their direct relation is not provided in the context.

**Models.** We conduct our main experiments across all four datasets for both reward model training and instruct-model fine-tuning using the Llama3-8B Instruct model. Due to computational constraints, we limit additional experiments

to selected datasets. To assess generalization across different experimental setups and out-of-distribution datasets, we also report results using Qwen2.5 models.

**Metrics.** We report the accuracy<sup>2</sup> for GSM8K and MATH datasets, accuracy and F1 for the MuSiQue-Ans dataset, and the accuracy and macro-F1 for the SpaRP dataset.

**Parameter Setting.** All models are fine-tuned using the HuggingFace TRL library with QLoRA ( $\alpha=16$ , dropout = 0.1, rank  $r=64$ ). Training is performed with an effective batch size of 32, learning rate of  $1e-4$ , a cosine scheduler with a warm up ratio of 0.03, and a maximum sequence length of 512, which is also used at inference unless otherwise specified. Experiments are conducted on 8 NVIDIA A100 GPUs (40 GB each). The number of training samples for the reward models are: 40,350 for GSM8K, 40,500 for MATH, 10K for MuSiQue, and 16K for SpaRP.

**Implementation Details and Baselines.** We begin with a single-epoch supervised fine-tuning (SFT) on the training split. Next, for each problem in the training and dev-set, we generate  $N = 20$  solutions from the fine-tuned model using a temperature of 1. These solutions are then annotated using final answers for outcome supervision and the SPARE framework for process supervision.

We employ the same pretrained models<sup>3</sup> for reference-guided step annotations using our SPARE framework. To account for problem diversity, we manually construct structured step-by-step evaluation exemplars per dataset—ranging from 6 for SpaRP to 56 for MATH—balanced for final answer correctness and covering all topics (MATH), sub-datasets (SpaRP), or reasoning graphs (MuSiQue-Ans). Each dataset is evaluated in a 5-shot setting, with exemplars selected randomly while ensuring both positive and negative examples, and using dataset-specific evaluation guidelines as system prompts. See Appendix A for an example.

In the *verification* scenario, we use process annotations from the SPARE framework to train SPARE-PRMs, predicting special tokens for correct and incorrect steps as a classification objective (Section 3.2) at special end-of-step tokens. We benchmark these models against outcome reward models (ORMs) and majority-voted self-consistency. To ensure balanced training, we randomly sample equal numbers of positive and negative examples. Evaluation metrics for both ORMs and PRMs are reported under two settings: (a) weighted aggregation (i.e., RM-weighted self-consistency) and (b) no aggregation, i.e., Best-of-N (BoN) sampling considering only the highest-scoring solution. Further training details and hyperparameters are provided in Appendix B.

In the *finetuning* scenario, we evaluate our SPARE-ORPO iteration trained on preference pairs formed using both outcome supervision and the mean reasoning scores of the step-by-step annotations (Section 3.2). We benchmark SPARE-

<sup>2</sup>Exact Match for GSM8K; competition math metric from evaluate library for MATH; Accuracy and F1 from official repositories for MuSiQue-Ans and from `scikit-learn` for SpaRP.

<sup>3</sup>Performance could be further improved using specialized evaluators like Prometheus 2 (Kim et al. 2024) or larger LLMs.

ORPO against Outcome-ORPO and second iteration of Supervised Fine-Tuning (SFT) with an equivalent number of training instances. The training hyperparameter details are provided in Appendix C.

## 4.2 Results and Discussion

**SPARE Improves Reward Model Training and Adapts to Diverse Reasoning Traces.** Table 1 shows that SPARE-PRM performs the best across all four datasets, outperforming both ORMs and the majority-voted Self-Consistency (SC). The improvements of the best SPARE-PRM ranked or aggregated strategy over the best Outcome baselines are statistically significant ( $p < 0.05$ ) under one-tailed paired  $t$ -test, with a maximum *relative* improvement of 4.8% accuracy on SpaRP-S. On the challenging MATH-500 dataset, it attains a relative improvement of 1.3%, reaching an accuracy of 24.1%. Notably, this improvement is consistent across difficulty levels and especially significant for more difficult problems (Figure 1a). The performance scales with increasing number of generated solutions (Figure 1b), consistently outperforming the baselines.

Table 2 presents additional results on the MATH dataset across varied settings, including different generation counts ( $N=20$  and 256), model families (LLaMA3 and Qwen2.5), and heterogeneous generator-RM configurations (e.g., Qwen2.5-32B generator with Qwen2.5-3B RM), different generation lengths (512 and 2048). SPARE demonstrates consistent performance across these setups, achieving up to a 1.2% absolute improvement over the ORM baseline. For context, results from Math-Shepherd (Wang et al. 2024a) and Rest-MCTS\* (Zhang et al. 2024), which follow similar methodologies of data construction from scratch and rely on a single model, show comparable gains over ORM.

Finally, Table 1 demonstrates that on datasets with limited reasoning variation (e.g., MuSiQue-Ans, SpaRP-S), SPARE-PRM with Best-of-N (BoN) sampling performs best, while self-consistency (SC) aggregation underperforms even ORMs. In contrast, on datasets with diverse reasoning forms (e.g., GSM8K, MATH-500), SC aggregation boosts SPARE-PRM’s performance, in comparison to both BoN and ORM baselines. Distributional analyses (Figure 2) further confirm this adaptability. On SpaRP-S, SPARE-PRM exhibits wider score spread and lower mean score for correct answers, reducing SC effectiveness. However, on MATH-500, its probability mass skews higher for correct answers, enabling SC to recover stronger performance.

**SPARE Helps in Fine tuning.** We report the performance of fine tuning LLM followed by greedy decoding in Table 1. SPARE-ORPO achieves the best performance across three of the four datasets, with a maximum relative improvement of 2.4% in F1 score on the SpaRP-S dataset compared to the Outcome-ORPO models. On the challenging MATH dataset, it attains a relative improvement of 1.3%, reaching an accuracy of 23.4%. The improvements of the SPARE models over Outcome baselines are statistically significant ( $p < 0.05$ ) under one-tailed paired  $t$ -test. This underscores the effectiveness of SPARE in reasoning step annotation and identifying *superior* preference pairs than outcome-only

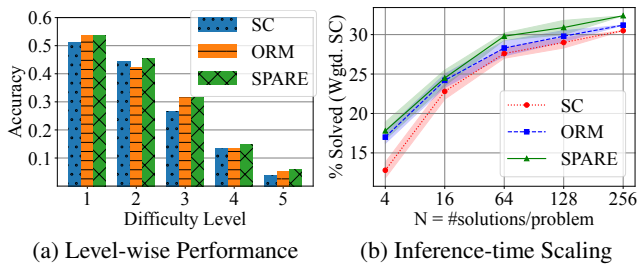


Figure 1: Llama-3 8B Instruct performance across SC, ORM-weighted, and SPARE PRM-weighted consistency by difficulty level, and candidate scaling on MATH-500.

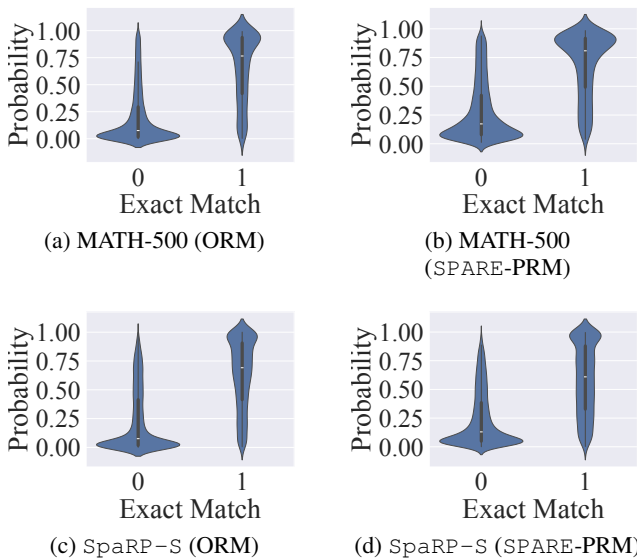


Figure 2: Distribution-plots of ORM and SPARE-PRM probabilities for correct and incorrect answers of Math-500 and SpaRP-S datasets.

preference pairs. Both these ORPO models significantly outperform the SFT models trained on the ground-truth reasoning traces, except for the GSM8K dataset. We attribute this to the saturation of performance on GSM8K, particularly as we used the same hyperparameters across all datasets (see parameter details in Section 4.1 and Appendix C.

**SPARE Exhibits Data-Efficient Out-of-Distribution Generalization.** Table 5 reports a fine-grained evaluation of SPARE-PRMs on the MATH, OlympiadBench, and OmniMATH subsets of the PROCESSBENCH (Zheng et al. 2025) benchmark, measuring the earliest *error* detection, full-solution *correctness*, and their harmonic mean (F1). For comparison, results from the leading PRMs within each model family and size, as well as a PRM trained on human-annotated data (PRM800K), are included from the original benchmark.

SPARE-PRMs achieve the highest accuracy on full-solution correctness identification, significantly outperforming other PRMs on out-of-distribution (OOD) subsets such

Aggregation	GSM8K Acc. ( $\uparrow$ )	MATH-500 Acc. ( $\uparrow$ )
Self-Consistency (SC)	83.1	33.6
+ ORM	86.7	35.1
+ Math-Shepherd	87.7	<b>35.4</b>
+ SPARE-PRM	<b>87.8</b>	<b>35.4</b>

Table 3: Performance comparison on mathematical datasets with Math-Shepherd (Wang et al. 2024a) MCTS approach. Accuracy averaged over 3 sampling groups.

Annot. Method	Avg. #Token	Avg. Time (s)
MCTS Rollouts (A)	10,569.1	40.5
SPARE (B)	4,591.3	15.5
Speed-up (A/B)	2.3	2.6
Efficiency ( $100 \times B/A$ )	43.5	38.5

Table 4: SPARE efficiency versus MCTS for process labeling on MATH in terms of average total tokens and runtime under identical compute.

as OlympiadBench and Omni-MATH. They also consistently match or outperform the PRM trained on human-labeled data. In terms of F1 and earliest error detection, SPARE-Qwen2.5-3B remains comparable on OOD datasets, achieving the highest F1 score on Omni-MATH dataset. Additionally, both Math-Shepherd and SPARE-Qwen2.5-3B exhibit greater robustness to distributional shift, as their error and F1 scores degrade less sharply from MATH to the OOD datasets compared to other top-performing models. Notably, SPARE achieves these results with high data efficiency, using only  $\sim 16\%$  of the training samples compared to both the human-labeled PRM800K and Deepseek-8B synthetic data, and just  $\sim 9\%$  relative to the MCTS-based Math-Shepherd model.

**SPARE is Compute-Efficient and Competitive with MCTS Methods.** For direct comparison with MCTS-based approaches, we follow the experimental setup of Math-Shepherd (Wang et al. 2024a), using Mistral-7B:MetaMATH for solution generation (256 samples) and a Mistral-7B-based PRM for SPARE. As shown in Table 3, SPARE-PRM slightly outperforms Math-Shepherd on GSM8K and performs comparably on Math-500 under weighted aggregation. Both methods surpass standard baselines such as self-consistency and ORM.

Table 4 highlights the computational efficiency of SPARE relative to the MCTS-based annotation used in Math-Shepherd. On the MATH dataset, SPARE reduces the average number of total tokens by  $2.3\times$  and runtime on identical compute setup (as outlined earlier in Section 4.1) by  $2.6\times$ , achieving an overall efficiency gain of  $\sim 40\%$  across both metrics. This efficiency arises from SPARE’s single-pass annotation process, in contrast to MCTS-based methods that require extensive search and repeated model inferences, significantly increasing the computational overhead.

Model	# Train	MATH			Olymp.Bench			Omni-MATH		
		Error	Correct	F1	Error	Correct	F1	Error	Correct	F1
Math-Shepherd-7B*	440K	18.0	82.0	29.5	<u>15.0</u>	71.1	<b>24.8</b>	<b>14.2</b>	73.0	<u>23.8</u>
RLHFlow-Deepseek-8B*	250K	<u>21.4</u>	80.0	<u>33.8</u>	10.1	51.0	16.9	10.9	51.9	16.9
Skywork-7B*	–	<b>43.8</b>	62.2	<b>53.6</b>	<b>17.9</b>	31.9	<u>22.9</u>	<u>14.0</u>	41.9	21.0
SPARE-Llama3-8B	40.5K	6.1	<b>91.6</b>	11.4	3.3	<b>87.6</b>	6.4	2.8	<u>82.2</u>	5.4
SPARE-Qwen2.5-3B	40.5K	16.0	<u>89.2</u>	27.1	11.1	<u>85.0</u>	19.6	<u>14.0</u>	<b>83.8</b>	<b>23.9</b>
Qwen-2.5-Math-7B-PRM800K (Human)	250K	48.0	90.1	62.6	35.7	87.3	50.7	29.8	86.1	44.3

Table 5: Fine-grained evaluation comparison of SPARE trained PRMs on PROCESSBENCH (Zheng et al. 2025). Best values in **bold**, second best in underline. Results marked with \* are from the PROCESSBENCH paper.

Dataset	Annot. Method	Label Acc.
GSM8K	DIVERSE-NLI (Llama-based*)	75.6
	MCTS (Math-Shepherd*)	85.0
	MCTS (Ours)	87.3
	SPARE	87.5
MATH	MCTS (Ours)	76.4
	SPARE	76.2

Table 6: Step-Label Accuracy of different automatic annotation processes on GSM8K and MATH dataset. Results marked with \* are from Wang et al. (2024a).

**SPARE Aligns Well with Manual Step Annotations and Exhibits Complementary Behavior to MCTS-Based Annotation.** We assessed the annotation accuracy of SPARE on 56 manually labeled MATH examples, balanced for answer correctness and spanning all seven topics, and 30 examples from GSM8K. Annotations were generated using Llama3-8B in a 5-shot setting, with the target excluded from in-context examples to prevent data leakage. Exemplars were randomly sampled per instance, and each annotation process was repeated ten times. Table 6 reports label accuracies for SPARE, our MCTS implementation, and prior baselines from Math-Shepherd (Wang et al. 2024a), where our MCTS follows their setup to ensure fair comparison. On GSM8K, SPARE, MCTS (ours), and MCTS (Math-Shepherd) achieve comparable high accuracy ( $\sim 85\%$ ), substantially outperforming DIVERSE-NLI, a directly comparable LLM-based reference-guided approach to SPARE. While DIVERSE-NLI performs alignment externally to single-steps without explicit reasoning, SPARE jointly performs step alignment (including multi-step) and label prediction via explicit reasoning, yielding more reliable annotations, particularly as  $\sim 20\%$  of labeled steps required multi-step alignment. Additionally, despite the increased difficulty of MATH dataset, both SPARE and MCTS maintain strong performance ( $\sim 76\%$ ).

Although overall label accuracies for MCTS and SPARE are comparable, the class-wise precision-recall analysis in Figure 3 reveals that for both single- and multi-aligned steps, SPARE achieves high recall for correct steps and high precision for incorrect ones (each  $> 80\%$ ), while MCTS excels in precision for correct steps and recall for incorrect ones (each

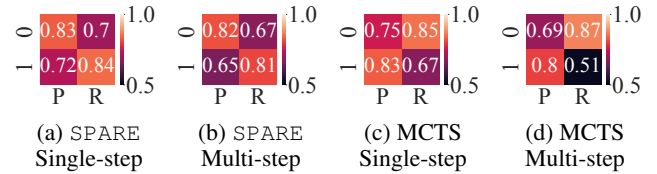


Figure 3: Precision (P) and Recall (R) of SPARE and MCTS annotations relative to human annotations for ground-truth single-step and multi-step alignments.

$> 80\%$ ). These complementary strengths suggest potential gains from combining SPARE and MCTS, for example via ensemble annotation. The observed performance drop from single- to multi-aligned steps reflects the increased difficulty of correct assessment of multi-aligned steps.

## 5 Conclusion

We present **Single-Pass Annotation with Reference-Guided Evaluation (SPARE)**, a structured framework that enables per-step annotation in a single pass by evaluating each solution step against one or multiple reference steps with explicit reasoning. Our experimental results demonstrate that fine-tuning a base model and training a reward model with SPARE lead to improved reasoning performance under both greedy decoding and ranking/aggregation of multiple solutions. Furthermore, we observe consistent improvements across four datasets spanning mathematical reasoning, multi-hop compositional question answering, and spatial reasoning. SPARE shows (i) data-efficient out-of-distribution generalization on PROCESSBENCH, (ii) competitive performance with greater compute efficiency compared to tree search-based annotation methods, and (iii) strong alignment with human annotations with complementary precision-recall characteristics. These findings highlight the potential of reference-guided automatic process supervision as a promising approach for enhancing LLM reasoning capabilities.

## Acknowledgments

This work has been funded by the Collaboration Lab with Nexplore “AI in Construction” (AICO). We further thank our anonymous reviewers and Chen Cecilia Liu, Rima

Hazra, and Tim Baumgärtner for their fruitful discussions and helpful feedback.

## 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*.
- BigBench-Team. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.
- Chen, Z.; Zhao, Z.; Zhu, Z.; Zhang, R.; Li, X.; Raj, B.; and Yao, H. 2024. AutoPRM: Automating Procedural Supervision for Multi-Step Reasoning via Controllable Question Decomposition. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1346–1362. Mexico City, Mexico: Association for Computational Linguistics.
- Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; Garcia, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omer-nick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Diaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2023. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24(240): 1–113.
- 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. arXiv:2110.14168.
- Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2023. Complexity-Based Prompting for Multi-step Reasoning. In *The Eleventh International Conference on Learning Representations*.
- Hao, S.; Gu, Y.; Ma, H.; Hong, J.; Wang, Z.; Wang, D.; and Hu, Z. 2023. Reasoning with Language Model is Planning with World Model. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 8154–8173. Singapore: Association for Computational Linguistics.
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Khalifa, M.; Agarwal, R.; Logeswaran, L.; Kim, J.; Peng, H.; Lee, M.; Lee, H.; and Wang, L. 2025. Process Reward Models That Think. arXiv:2504.16828.
- Khalifa, M.; Logeswaran, L.; Lee, M.; Lee, H.; and Wang, L. 2023. GRACE: Discriminator-Guided Chain-of-Thought Reasoning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 15299–15328. Singapore: Association for Computational Linguistics.
- Kim, S.; Suk, J.; Longpre, S.; Lin, B. Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; and Seo, M. 2024. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 4334–4353. Miami, Florida, USA: Association for Computational Linguistics.
- Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.
- 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 Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5315–5333. Toronto, Canada: Association for Computational Linguistics.
- 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 *The Twelfth International Conference on Learning Representations*.
- Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; Tang, Y.; and Zhang, D. 2025. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. arXiv:2308.09583.
- Luo, L.; Liu, Y.; Liu, R.; Phatale, S.; Guo, M.; 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. arXiv:2406.06592.
- Pan, S.; Lialin, V.; Muckatira, S.; and Rumshisky, A. 2023. Let’s Reinforce Step by Step. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

- Rizvi, M. I.; Zhu, X.; and Gurevych, I. 2024. SpaRC and SpaRP: Spatial Reasoning Characterization and Path Generation for Understanding Spatial Reasoning Capability of Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4750–4767. Bangkok, Thailand: Association for Computational Linguistics.
- She, S.; Liu, J.; Liu, Y.; Chen, J.; Huang, X.; and Huang, S. 2025. R-PRM: Reasoning-Driven Process Reward Modeling. arXiv:2503.21295.
- Song, H.; Su, H.; Shalyminov, I.; Cai, J.; and Mansour, S. 2024. FineSurE: Fine-grained Summarization Evaluation using LLMs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 906–922. Bangkok, Thailand: Association for Computational Linguistics.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.
- Trivedi, H.; Balasubramanian, N.; Khot, T.; and Sabharwal, A. 2022. ↻ MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics*, 10: 539–554.
- Uesato, J.; Kushman, N.; Kumar, R.; Song, F.; Siegel, N.; Wang, L.; Creswell, A.; Irving, G.; and Higgins, I. 2022. Solving math word problems with process- and outcome-based feedback. arXiv:2211.14275.
- Wang, P.; Li, L.; Shao, Z.; Xu, R.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2024a. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9426–9439. Bangkok, Thailand: Association for Computational Linguistics.
- Wang, Z.; Li, Y.; Wu, Y.; Luo, L.; Hou, L.; Yu, H.; and Shang, J. 2024b. Multi-step Problem Solving Through a Verifier: An Empirical Analysis on Model-induced Process Supervision. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 7309–7319. Miami, Florida, USA: Association for Computational Linguistics.
- Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022a. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022b. 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*, volume 35, 24824–24837. Curran Associates, Inc.
- Yao, S.; Yu, D.; Zhao, J.; Shafraan, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. R. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Ye, X.; and Durrett, G. 2022. The Unreliability of Explanations in Few-shot Prompting for Textual Reasoning. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.
- Yu, F.; Gao, A.; and Wang, B. 2024. OVM, Outcome-supervised Value Models for Planning in Mathematical Reasoning. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, 858–875. Mexico City, Mexico: Association for Computational Linguistics.
- Yu, L.; Jiang, W.; Shi, H.; YU, J.; Liu, Z.; Zhang, Y.; Kwok, J.; Li, Z.; Weller, A.; and Liu, W. 2024. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. In *The Twelfth International Conference on Learning Representations*.
- Zhang, D.; Zhou, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024. ReST-MCTS\*: LLM Self-Training via Process Reward Guided Tree Search. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- 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*.
- 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 Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Findings of the Association for Computational Linguistics: ACL 2025*, 10495–10516. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-256-5.
- Zheng, C.; Zhang, Z.; Zhang, B.; Lin, R.; Lu, K.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025. ProcessBench: Identifying Process Errors in Mathematical Reasoning. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1009–1024. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.
- Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q. V.; and Chi, E. H. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations*.