GenSelect: A Generative Approach to Best-of-N

Shubham Toshniwal [1] **Ivan Sorokin** [1] **Aleksander Ficek** [1] **Ivan Moshkov** [1] **Igor Gitman** [1]

Abstract

Generative reward models with parallel sampling have enabled effective test-time scaling for reasoning tasks. Current approaches employ pointwise scoring of individual solutions or pairwise comparisons. However, pointwise methods underutilize LLMs' comparative abilities, while pairwise methods scale inefficiently with larger sampling budgets. We introduce GenSelect, where the LLM uses long reasoning to select the best solution among N candidates. This leverages LLMs' comparative strengths while scaling efficiently across parallel sampling budgets. For math reasoning, we demonstrate that reasoning models, such as `QwQ` and `DeepSeek-R1-0528`, excel at GenSelect, outperforming existing scoring approaches with simple prompting.

1. Introduction

Test-time scaling has emerged as a powerful paradigm for enhancing the performance of large language models on reasoning tasks (OpenAI, 2024; DeepSeek-AI, 2025). This scaling approach leverages two complementary mechanisms: *sequential scaling*, which utilizes long reasoning models, and *parallel sampling*, which produces multiple independent candidate solutions (Snell et al., 2025). Central to the parallel sampling strategy is the critical task of identifying the highest-quality response from the generated candidates. Typical candidate evaluation methods typically rely on rule-based methods (Hassid et al., 2025) or discriminative reward models (Cobbe et al., 2021; Liu et al., 2025b). Recent work has demonstrated the efficacy of generative reward models (GenRMs), which offer another axis for leveraging test-time computation (Zhang et al., 2025a; Mahan et al., 2024). Moreover, recent findings suggest that these generative verification and reward models also benefit from extended reasoning (Chen et al., 2025; Guo et al., 2025).

These generative reward models primarily operate through two distinct evaluation paradigms: *pointwise evaluation*, where models assess individual candidate outputs in isolation (Zhang et al., 2025a), and *pairwise comparison*, where models evaluate pairs of candidates relative to one another (Mahan et al., 2024). While pointwise GenRMs are easily compatible with parallel sampling strategies, they are constrained by the inherent limitations of LLMs in verification tasks (Tyen et al., 2024; Zhao et al., 2025). Conversely, pairwise GenRMs leverage the comparative strengths of LLMs but extending them to larger candidate pools is non-trivial (Liu et al., 2024). The computational complexity becomes particularly prohibitive when performing exhaustive pairwise comparisons, requiring $\mathcal{O}(N^2)$ evaluations for ranking N candidates (Jiang et al., 2023). Approximations such as pairwise knockout tournaments can reduce the computation to $\mathcal{O}(N)$ but increase latency by a factor of $\mathcal{O}(log_2 N)$ (Liu et al., 2025a). This computational burden is further amplified when using reasoning-based generative reward models for such pairwise comparisons.

To address these limitations, we propose GenSelect, where the LLM is tasked with performing the best-of-N judgment given the N candidate solutions. This generalization of GenRM from binary to N-ary comparison allows for a far efficient application of reasoning-based LLMs to larger parallel sampling budgets. To extend GenSelect to even larger sampling budgets, beyond the context window size limitations, we propose a N-ary knockout tournament similar to PairJudge-RM (Liu et al., 2025a).

Our experiments with `QwQ` and `DeepSeek-R1` demonstrate that reasoning models are remarkably adept at the GenSelect task out of the box. Our results show that GenSelect substantially outperforms prior approaches on various competition-level math reasoning benchmarks. We also find that the GenSelect performance is relatively stable across different inference setups, and thus, N-way comparisons with large N allow for highly efficient scaling to higher parallel sampling budgets without performance degradation.

[*]Equal contribution [1]NVIDIA. Correspondence to: Shubham Toshniwal <stoshniwal@nvidia.com>.

The second AI for MATH Workshop at the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. Copyright 2025 by the author(s).

2. Related Work

In this section, we discuss some of the popular reward model work in the context of the mathematical reasoning task. For brevity, we will limit our discussion of reward models for general use cases.

2.1. Discriminative Reward Models

Cobbe et al. (2021) was one of the first works to use outcome-based reward models (ORMs) where they showed performance benefits with the use of verifiers for selecting the best solution. Follow-up work by Yang et al. (2024) and Liu et al. (2025b) demonstrated continued gains with the use of the latest ORMs, even with the dramatic performance improvement of generators.

Besides ORMs, there has been a rich line of work in process reward models (PRMs), where the task is to teach the model to grade based on both the outcome and the process used to arrive at the answer. While earlier work relied on costly human annotations (Lightman et al., 2023), recent work utilizes automatic labels via repeated rollouts from different points in the solution trajectory (Zhang et al., 2025b).

2.2. Generative Reward Models

Instead of training models to output numerical scores through specialized classification heads, GenRMs leverage the text generation capabilities of large language models by representing correctness decisions using the log probability of specific tokens, typically ‘Yes’ or ‘No’, conditioned on a prompt consisting of the question, corresponding solution, and instruction to judge the correctness of the solution. GenRMs enable two ways of verification/scoring test-time scaling via chain-of-thought reasoning and parallel sampling (Zhang et al., 2025a).

Recent work has explored the use of training reasoning models for the task of verification via reinforcement learning (RL). Shi & Jin (2025) trained such a model for math solution verification, and Guo et al. (2025) trained general verifier models, both models achieving great success with extended reasoning. While the proposed GenSelect method also has a natural RL formulation, we focus on demonstrating the out-of-the-box capabilities of current reasoning models.

Apart from scoring a standalone solution, GenRMs have also been used and developed for comparing two inputs (Mahan et al., 2024). For the math reasoning task, Zhao et al. (2025) note that while *frontier* LLMs can be *weak* verifiers, their capability to identify errors improves when verification is done via comparison of responses. As with *pointwise* GenRMs, long reasoning models trained via RL aid the binary classification task of picking the preferred response as well (Chen et al., 2025). While GenRMs, which compare two responses, are an excellent fit for popular reward model benchmarks, using these models for the Best-of-N (BoN) task is non-trivial. Exhaustive pairwise comparisons can be computationally very costly (Jiang et al., 2023), especially with the new reasoning models, and approximations such as pairwise knockout tournaments can reduce the computation cost at the cost of latency (Liu et al., 2025a).

3. Methodologies

In this section, we provide a brief overview of relevant baselines and contrast them with GenSelect. To make the discussion concrete, we consider an input problem X and a set of corresponding solutions $\{Y_1, \cdots, Y_N\}$. The goal is to select a solution that accurately answers the input problem. We assume access to the function $\texttt{Ans}(Y)$, which extracts the final answer from a given solution trajectory.

Additionally, we utilize the function $\texttt{Summary}(Y)$, which produces a faithful summary of a long reasoning solution, including any backtracking and self-verification steps. Our initial experiments revealed no significant benefit from using complete reasoning traces compared to solution summaries in solution representations. We therefore adopt solution summaries for all methods, which has the added advantage of compatibility with shorter context window models. While reasoning models naturally produce summaries (the solution portion following the thinking section), our preliminary experiments with solutions generated by `QwQ` indicated modest benefits from generating fresh summaries using `Qwen2.5-32B-Instruct` (prompt details in Appendix A.1).

3.1. Baselines

Pass@N. This is the oracle baseline, which selects any of the solutions that reach the ground truth answer. This serves as the upper bound for all the solution ranking methods, including GenSelect.

Majority Voting/Self-Consistency. Proposed by (Wang et al., 2023), this method selects the most common answer from the solution candidates.

$$\text{Majority}(\{\texttt{Ans}(Y_1), \ldots, \texttt{Ans}(Y_N)\})$$

Given that this approach only uses the final answer for aggregation, this is a shallow approach for aggregating multiple responses.

Discriminative Reward Model. A discriminative RM assigns floating-point scores to candidate solutions, which

GenRM Prompt

```
You will be given a challenging math problem followed by a solution. Your task is to
 systematically analyze this solution to determine whether it is mathematically
sound and correct.

Input Format:
Problem: A complex mathematical word problem at advanced high school or college
level
Solution: A detailed solution concluding with an answer in \boxed{{}} notation

YOUR TASK

Problem: {problem}

Solution:
{generation}

Evaluation Process:

1. Mathematical Accuracy Check
- Verify all computational steps for arithmetic errors
- Check algebraic manipulations and equation solving
- Validate any formulas, theorems, or mathematical principles used
- Ensure proper mathematical notation and \boxed{{}} format

2. Logical Reasoning Analysis
- Examine the logical progression from problem statement to solution
- Identify any gaps or jumps in reasoning
- Verify that each step follows logically from the previous ones
- Check that the approach appropriately addresses the problem's requirements

3. Completeness and Method Evaluation
- Evaluate handling of edge cases or special conditions if applicable
- Determine if the chosen method is appropriate for the problem type

Your response should include:
1. Step-by-step verification of the mathematical work
2. Analysis of the logical structure and reasoning
3. Identification of any errors, omissions, or weaknesses

End your evaluation with exactly:
Judgment: [Yes/No]
where "Yes" means the solution is mathematically sound and correct, and "No" means
the solution contains significant errors or is incorrect.
```

Figure 1. Prompt used for GenRM.

can then be used to select the highest-scoring solution or perform weighted majority voting (Cobbe et al., 2021; Yang et al., 2024). In our experiments, we find that weighted majority voting performs better than selecting the highest-scoring solution, a finding corroborated by Wu et al. (2025).

As a baseline, we compare against the `Qwen2.5-Math-RM-72B` model by Yang et al. (2024), one of the best-performing outcome-based reward models from prior work.

Generative Reward Model. We establish a GenRM baseline by zero-shot prompting the `QwQ` model using our verification prompt (prompt in Figure 1). This prompt-based approach serves two purposes: it addresses the unavailability of publicly released GenRM models from prior work (Zhang et al., 2025a; Shi & Jin, 2025), while enabling direct comparison with GenSelect under equivalent conditions. This setup facilitates a fair and meaningful comparison between absolute scoring (GenRM) and comparative scoring (GenSelect) paradigms using the same underlying reasoning model.

GenRMs enable straightforward test-time scaling of verification. We sample multiple verifications from `QwQ` at a temperature of 0.6.

3.2. GenSelect

The predominant approach for BoN inference has been to: (a) assign a score to the N solutions, either via a discriminative or generative classifier, and use this score to select the answer, or (b) perform pairwise comparisons and deduce the *best* solution via a sequence of such binary comparisons. Rather than limiting the model to one or two candidate solutions at a time and going a roundabout way to selecting the best candidate, GenSelect tackles the BoN task head-on by directly evaluating all N candidate solutions simultaneously and having the LLM perform an N-ary comparison to identify the *best* response (see prompt in Figure 2). We show that current reasoning models are already adept at this task.

In practice, context window limitations present scalability challenges for $N > 16$. For instance, `QwQ-32B` has a maximum context of 40,960 tokens, which includes both the prompt and response tokens. To scale GenSelect to larger sampling budgets, we employ an N-ary knockout tournament approach, following the methodology established in PairJudge-RM (Liu et al., 2025a). Although N-ary and binary knockout tournaments maintain equivalent theoretical computational complexity of $\Theta(N)$ and latency of $\Theta(logN)$, N-ary tournaments demonstrate substantially superior practical efficiency. For example, when evaluating 64 candidate solutions, binary comparisons require six sequential rounds comprising 63 total comparisons, whereas 16-way comparisons require merely two rounds with five total comparisons, representing a significant reduction in computational overhead. This becomes even more evident with the overhead of reasoning models performing long reasoning to determine the best solution in each comparison. We also demonstrate that GenSelect exhibits stable performance across various values of N, indicating that efficiency gains can be achieved with larger N without performance degradation.

Problem source	# of Problems
AIME 2024	30
AIME 2025	30
HMMT Nov 2024	62
HMMT Feb 2024	68
HMMT Feb 2025	66
Total	256

Table 1. Composition of Comp-Math-24-25.

4. Experimental Setup and Results

4.1. Evaluation Benchmark

The AIME competitions are a popular benchmark for math reasoning, but they consist of only 30 questions, resulting in high variance in performance measurements. To reduce this variance, we combine problems from the American Invitational Mathematics Examinations (AIME) and Harvard-MIT Mathematics Tournaments (HMMT) for the years 2024 and 2025. We exclude proof-based questions and those awarding partial credit based on estimated accuracy. We refer to this dataset as `Comp-Math-24-25`, which consists of 256 problems, as detailed in Table 1.

4.2. Evaluation Details

We evaluate two of the most popular open-weight reasoning models, namely, `QwQ-32B` and `DeepSeek-R1-0528`, in our experiments. For both models, we use a sampling temperature of 0.6 to generate solutions, and when the model is used as a verifier/reward model.

4.3. Results

Comparison with Baselines. Table 3 presents a comprehensive comparison of various Best-of-N selection methods alongside our proposed GenSelect approach. The underperformance of `Qwen2.5-Math-RM-72B` relative to majority voting can be attributed to distribution mismatch,

GenSelect Prompt

```
You will be given a challenging math problem followed by {num_solutions} solutions.
Your task is to systematically analyze these solutions to identify the most
mathematically sound approach.

Input Format:
Problem: A complex mathematical word problem at advanced high school or college
level
Solutions: Detailed solutions indexed 0-{max_idx}, each concluding with an answer in
 \boxed{{}} notation

YOUR TASK

Problem: {problem}

Solutions:
{solutions}

Evaluation Process:

1. Initial Screening
- Group solutions by their final answers
- Identify and explain mathematical contradictions between different answers
- Eliminate solutions with clear mathematical errors

2. Detailed Analysis
For remaining solutions, evaluate:
- Mathematical precision and accuracy
- Logical progression of steps
- Completeness of mathematical reasoning
- Proper use of mathematical notation, including \boxed{{}}
- Handling of edge cases or special conditions
- For solutions containing and addressing errors, evaluate the error identification
and correction methodology.

3. Solution Comparison
Compare viable solutions based on:
- Efficiency of approach
- Clarity of mathematical reasoning
- Sophistication of method
- Robustness of solution (works for all cases)

Your response should include:
1. Brief analysis of conflicting answers
2. Detailed evaluation of mathematically sound solutions
3. Justification for eliminating incorrect solutions
4. Clear explanation for selecting the best approach

End your evaluation with exactly:
Judgment: [IDX]
where IDX is the index 0-{max_idx} of the best solution.
```

Figure 2. The prompt used for GenSelect includes 0-indexed solution candidates, and the model must reference the best solution by its corresponding index in the final judgment.

	DeepSeek-R1-0528		QwQ-32B	
N	**Generation** $(2N)$	**Generation** (N) **+ GenSelect** (N)	**Generation** $(2N)$	**Generation** (N) **+ GenSelect** (N)
4	**82.8**	82.4	66.4	**69.5**
8	84.4	**85.9**	66.8	**69.5**
16	84.4	**87.1**	68.0	**71.9**

Table 2. Comparison of performance with `DeepSeek-R1-0528` and `QwQ-32B` when spending inference compute budget entirely on solution generation, followed by majority voting vs dividing the inference budget equally into solution and GenSelect generation.

BoN Method	**Accuracy (in %)**
Pass@64	85.2
Majority@64	68.4
Qwen2.5-Math-RM-72B	66.8
QwQ GenRM	69.1
QwQ GenSelect@1	**72.1**
QwQ GenSelect@8	**73.4**

Table 3. Accuracy on Comp-Math-24-25 for solutions generated by `QwQ-32B`. For both GenRM and GenSelect, we use the `QwQ-32B` model itself to score the candidate solutions. For GenSelect, we conduct a 16-way competition followed by a 4-way competition to determine the best solution.

as the model was trained on a different data distribution. Additionally, prior research has demonstrated that discriminative reward models tend to degrade relative to majority voting at large sampling budgets. While QwQ GenRM achieves modest improvements over the majority baseline, these gains are substantially exceeded by QwQ GenSelect@1, which demonstrates that leveraging the comparative strengths of LLMs for selecting the best solution is more effective than scoring each solution individually and then selecting the best one. The performance advantage becomes even more pronounced when the GenSelect pipeline is repeated eight times with majority voting applied to the resulting outputs, achieving 73.4% accuracy. This substantial improvement underscores that current reasoning models, like `QwQ`, are adept at GenSelect with simple prompting.

Stability of GenSelect. In this ablation, we compare the performance of GenSelect in different inference setups. In particular, we change the value of N from $\{2, 4, 8, 16\}$. Note that when $N = 2$, for a candidate solution set of 64 solutions, we require six rounds of comparison to decide the best solution, whereas just two rounds of comparisons are required with $N = 8$ and 16.

Table 4 presents the performance of these different setups,

N	**GenSelect@1**	**GenSelect@8**
2	72.1	73.4
4	72.6	73.0
8	72.3	73.4
16	72.1	73.4

Table 4. Comparison of GenSelect performance with different values of N which determines the N-ary comparisons performed.

and from the results, it is pretty evident that GenSelect is relatively stable across the different inference setups. This suggests that we can make N large (up to context window limits), and accelerate the best solution selection pipeline without performance degradation.

Generation vs Verification. Given an inference compute budget, are we better off spending it on generating more solutions, or spending some of it on verification, or in our case, on GenSelect. To answer this question, we compare two scenarios: (a) $2N$ solution generations followed by majority voting, and (b) N solution generations followed by N GenSelect generations. Note that while the two systems may use a similar amount of computation, the latency of the second one would most likely be worse than the first one, since GenSelect pipeline can only start once the solution generations have finished.

Table 2 presents results for this inference compute allocation comparison for `DeepSeek-R1-0528` and `QwQ-32B`. For the stronger model, `DeepSeek-R1-0528`, we see that focusing on generation-only for lower sampling budgets is preferred, but for $N >= 8$, allocating equal inference compute to GenSelect is preferred. For the weaker `QwQ-32B` model, allocating equal compute to GenSelect is advantageous for all sampling budgets in our study.

Final Results. Table 5 presents the results for both `DeepSeek-R1-0528` and `QwQ-32B`, along with their Self-GenSelect counterparts. For both models, we see a

Model	Comp-Math-24-25			
	AIME24	AIME25	HMMT-24-25	All
DeepSeek-R1-0528	85.9 (93.3)	80.5 (86.7)	67.5 (82.1)	71.2 (84.0)
+ Self GenSelect@32	93.3	90.0	85.7	87.1
QwQ-32B	78.1 (86.7)	67.2 (76.7)	56.1 (64.3)	60.0 (68.4)
+ Self GenSelect@32	90.0	73.3	70.4	73.0

Table 5. All models are evaluated with a maximum of 32K output tokens, temperature of 0.6, and top-p 0.95. We present metrics as pass@1 (maj@64) where pass@1 is an average accuracy across 64 generations and maj@64 is the result of majority voting. For HMMT, we use the LLM-judge setup of (Toshniwal et al., 2025) to verify the answers. For GenSelect, we use $N = 8$, which requires two rounds of scoring eight solutions each. We repeat GenSelect 32 times with random solution permutations and perform majority voting over the answers selected by GenSelect.

significant improvement in performance over the majority baseline. In particular, we see the gains are predominantly in the HMMT-24-25 split of Comp-Math-24-25.

5. Conclusion

In this work, we introduced GenSelect, a tournament-based Best-of-N selection method that leverages the comparative evaluation capabilities of large language models. Our experimental evaluations for mathematical reasoning tasks demonstrate that GenSelect consistently outperforms established baselines including majority voting, discriminative reward models, and generative reward models. The method exhibits remarkable stability across different tournament configurations and addresses practical scalability limitations through efficient N-ary knockout tournaments that respect context window constraints while significantly reducing computational overhead.

Our analysis of inference compute allocation reveals that GenSelect's effectiveness varies with model capability, becoming advantageous for stronger models at moderate sampling budgets while benefiting weaker models across all tested configurations. The simplicity of GenSelect's implementation—requiring only straightforward prompting without specialized training—combined with its consistent performance gains, makes it an immediately applicable technique for enhancing mathematical reasoning systems. Future work could explore extending GenSelect to other reasoning domains and general tasks. The GenSelect formulation is also suitable for reinforcement learning, and future work can leverage RL to learn the comparative capabilities of larger reasoning language models in smaller LLMs.

References

Chen, X., Li, G., Wang, Z., Jin, B., Qian, C., Wang, Y., Wang, H., Zhang, Y., Zhang, D., Zhang, T., Tong, H., and Ji, H. RM-R1: Reward Modeling as Reasoning, 2025. URL `https://arxiv.org/abs/2505.02387`.

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. Training Verifiers to Solve Math Word Problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Guo, J., Chi, Z., Dong, L., Dong, Q., Wu, X., Huang, S., and Wei, F. Reward Reasoning Model, 2025. URL `https://arxiv.org/abs/2505.14674`.

Hassid, M., Synnaeve, G., Adi, Y., and Schwartz, R. Don't Overthink it. Preferring Shorter Thinking Chains for Improved LLM Reasoning, 2025. URL `https://arxiv.org/abs/2505.17813`.

Jiang, D., Ren, X., and Lin, B. Y. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *ACL*, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's Verify Step by Step, 2023. URL `https://arxiv.org/abs/2305.20050`.

Liu, Y., Zhou, H., Guo, Z., Shareghi, E., Vulic, I., Korhonen, A., and Collier, N. Aligning with Human Judgement: The Role of Pairwise Preference in Large Language Model Evaluators. *arXiv preprint arXiv:2403.16950*, 2024.

Liu, Y., Yao, Z., Min, R., Cao, Y., Hou, L., and Li, J. PairJudge RM: Perform Best-of-N Sampling with Knockout

Tournament, 2025a. URL https://arxiv.org/abs/2501.13007.

Liu, Z., Chen, Y., Shoeybi, M., Catanzaro, B., and Ping, W. AceMath: Advancing Frontier Math Reasoning with Post-Training and Reward Modeling, 2025b. URL https://arxiv.org/abs/2412.15084.

Mahan, D., Phung, D. V., Rafailov, R., Blagden, C., Lile, N., Castricato, L., Fränken, J.-P., Finn, C., and Albalak, A. Generative Reward Models, 2024. URL https://arxiv.org/abs/2410.12832.

OpenAI. OpenAI o1 System Card, 2024. URL https://arxiv.org/abs/2412.16720.

Shi, W. and Jin, X. Heimdall: test-time scaling on the generative verification, 2025. URL https://arxiv.org/abs/2504.10337.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. In *ICLR*, 2025.

Toshniwal, S., Du, W., Moshkov, I., Kisacanin, B., Ayrapetyan, A., and Gitman, I. OpenMathInstruct-2: Accelerating AI for Math with Massive Open-Source Instruction Data. In *ICLR*, 2025.

Tyen, G., Mansoor, H., Carbune, V., Chen, P., and Mak, T. LLMs cannot find reasoning errors, but can correct them given the error location. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of ACL*, 2024.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Wu, Y., Sun, Z., Li, S., Welleck, S., and Yang, Y. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models, 2025. URL https://arxiv.org/abs/2408.00724.

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. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement, 2024. URL https://arxiv.org/abs/2409.12122.

Zhang, L., Hosseini, A., Bansal, H., Kazemi, M., Kumar, A., and Agarwal, R. Generative Verifiers: Reward Modeling as Next-Token Prediction. In *ICLR*, 2025a.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The Lessons of Developing Process Reward Models in Mathematical Reasoning, 2025b. URL https://arxiv.org/abs/2501.07301.

Zhao, E., Awasthi, P., and Gollapudi, S. Sample, Scrutinize and Scale: Effective Inference-Time Search by Scaling Verification, 2025. URL https://arxiv.org/abs/2502.01839.

A. Prompts

A.1. Solution Summary Prompt

Summary Prompt

```
I will give you a math problem and a long solution to that problem exploring
different approaches, making mistakes along the way, correcting them, switching
around and so on. But eventually that solution gets to the right approach and solves
 the problem. Your task is to write a clean version of the final correct solution
without all the exploration. Cover all the details of the final solution.

Problem:
{problem}

Solution:
{generation}

Now write a clean version of the final correct solution without all the exploration
but cover all the details of the final solution.
```