SkewGuard-PoLR: Investigating Dirichlet-Uncertainty Gated Multi-Cluster Expansion for Prefix-Consensus Self-Consistency

FARS
Analemma
fars@analemma.ai

Abstract

Test-time scaling through self-consistency (SC) improves large language model reasoning but incurs substantial computational cost. The Path of Least Resistance (PoLR) reduces this cost by expanding only the dominant answer cluster after prefix-based sampling, yet reportedly suffers tail failures when the dominant cluster is incorrect. We propose SkewGuard-PoLR, which places a Dirichlet posterior over cluster proportions and triggers multi-cluster expansion when the credible set indicates uncertainty about cluster dominance. However, our experiments on AIME25 and GPQA-Diamond with QwQ-32B and DeepSeek-R1-Distill-Qwen-7B reveal that PoLR does not exhibit the reported tail failures: PoLR achieves 78.89% accuracy on AIME25, outperforming SC (77.78%) by 1.11 percentage points rather than underperforming by 10 points as previously reported. Consequently, SkewGuard-PoLR provides no accuracy improvement while incurring 17% higher computational cost. This negative result demonstrates that the tail failure assumption underlying our approach does not hold under current evaluation conditions, helping the community avoid similar directions.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 Introduction

Large language models (LLMs) have demonstrated remarkable reasoning capabilities when prompted to generate intermediate reasoning steps (Wei et al., 2022). Self-consistency (Wang et al., 2023) further improves reliability by sampling multiple reasoning traces and selecting the answer via majority vote. However, this approach incurs substantial computational cost, as it requires generating N complete traces for each problem. Recent work has shown that test-time compute can be allocated more efficiently (Snell et al., 2024), motivating methods that reduce the number of full traces while preserving accuracy.

Path of Least Resistance (PoLR) (Jindal et al., 2026) addresses this efficiency challenge by exploiting prefix consistency: reasoning traces often share similar early prefixes that correlate with final answers. PoLR samples short prefixes, clusters them, and expands only the dominant cluster to full traces, achieving 40–60% token reduction while often matching self-consistency accuracy. However, the original PoLR paper reports a notable failure case: on QwQ-32B with AIME25, PoLR achieves 66.7% accuracy compared to self-consistency's 76.7%, a 10 percentage point drop. This suggests that when the dominant cluster happens to contain incorrect reasoning paths, PoLR discards the correct paths entirely.

We hypothesize that such tail failures occur when cluster dominance is statistically unreliable—when multiple clusters have similar sizes, the "dominant" cluster selection becomes a high-variance decision. To address this, we propose SkewGuard-PoLR, which uses Bayesian uncertainty quantification to decide when to expand beyond the dominant cluster. Specifically, we model uncertainty over cluster proportions using a Dirichlet posterior and expand all clusters within a credible set, hedging against incorrect dominant cluster selection when dominance is ambiguous.

[1] https://gitlab.com/fars-a/skewguard-polr

Surprisingly, our experiments reveal that the underlying assumption does not hold: **PoLR does not exhibit the reported tail failures in our evaluation**. On QwQ-32B with AIME25, PoLR achieves 78.89% accuracy compared to self-consistency's 77.78%, a +1.11 percentage point improvement rather than the reported -10 percentage point degradation. Consequently, SkewGuard-PoLR provides no accuracy benefit while incurring 17–32% additional computational cost.

This paper makes three contributions. First, we propose SkewGuard-PoLR, a method that uses Dirichlet posterior uncertainty to expand multiple prefix clusters when dominance is ambiguous, providing a principled approach to hedge against incorrect cluster selection. Second, we demonstrate that PoLR's reported tail failures on AIME25 do not reproduce in our evaluation, with PoLR achieving +1.11 percentage points over self-consistency rather than the reported -10 percentage point degradation. Third, we provide comprehensive experiments across two models (QwQ-32B, DeepSeek-R1-Distill-Qwen-7B) and two benchmarks (AIME25, GPQA-Diamond), showing that uncertainty-based multi-cluster expansion is unnecessary when PoLR's tail failures do not manifest.

2 Related Work

Self-Consistency and Adaptive Stopping. Self-consistency (Wang et al., 2023) improves chain-of-thought reasoning (Wei et al., 2022) by sampling multiple reasoning traces and taking a majority vote over final answers. While effective, the N-fold increase in decoding cost has motivated adaptive stopping methods that terminate sampling when answer agreement is sufficient. Early-stopping self-consistency (ESC) (Li et al., 2024) monitors answer stability in a sliding window, while CGES (Aghazadeh et al., 2025) uses confidence-guided thresholds. ConSol (Lee et al., 2025) applies sequential probability ratio testing with explicit Type-I error control, and BEACON (Wan et al., 2025) frames adaptive sampling as Bayesian optimal stopping. These methods reduce compute on easy instances but require generating full traces before making stopping decisions.

Prefix-Based Efficiency Methods. An alternative approach exploits the observation that reasoning traces often share similar early prefixes that correlate with final answers. Path of Least Resistance (PoLR) (Jindal et al., 2026) clusters short prefixes and expands only the dominant cluster, achieving substantial token savings. Path-consistency (Zhu et al., 2025) similarly leverages prefix information to guide decoding. Unlike answer-level stopping methods, prefix-based approaches make allocation decisions before paying full-trace costs, enabling greater parallelism.

Test-Time Scaling. Recent work has established that allocating additional compute at inference time can be more effective than scaling model parameters (Snell et al., 2024). Comprehensive surveys (Zhang et al., 2025; Wang et al., 2025) categorize test-time scaling strategies including sampling-based aggregation, search-based methods like Tree of Thoughts (Yao et al., 2023), and iterative refinement approaches (Madaan et al., 2023). Methods such as Slim-SC (Hong et al., 2025) and reasoning-aware self-consistency (Wan et al., 2024) reduce redundancy by pruning or weighting traces based on similarity or quality signals.

Positioning. Our work extends PoLR with Dirichlet-based uncertainty quantification over cluster dominance, aiming to expand multiple clusters when the dominant cluster is uncertain. However, our experiments reveal that the prerequisite condition—PoLR exhibiting tail failures when the dominant cluster is incorrect—does not reproduce in our evaluation, rendering the proposed extension unnecessary.

3 Method

3.1 Background: Path of Least Resistance

Self-consistency (Wang et al., 2023) samples N independent reasoning traces from a language model M given input x, then selects the final answer by majority vote. While effective, this incurs N-fold decoding cost. Path of Least Resistance (PoLR) (Jindal et al., 2026) reduces this cost by exploiting prefix consistency: reasoning traces often share similar early prefixes that correlate with final answers.

PoLR operates in four steps. First, it samples N short prefixes $\{p_i\}_{i=1}^{N}$ of length L_p tokens (typically 256). Second, it embeds each prefix using TF-IDF bag-of-words encoding and clusters them via agglomerative hierarchical clustering with cosine similarity, yielding clusters $\mathcal{C} = \{C_1, \dots, C_m\}$. Third, it identifies the dominant cluster $C^* = \arg\max_{C_j \in \mathcal{C}} |C_j|$ and expands only prefixes in C^* to full reasoning traces. Finally, it applies majority voting over the extracted answers from expanded traces.

The token efficiency of PoLR is $\eta = 1 - \frac{N \cdot \ell_p + K \cdot (\ell_f - \ell_p)}{N \cdot \ell_f}$, where $K = |C^*|$ is the number of expanded traces, ℓ_p is the prefix length, and ℓ_f is the average full trace length. When $K \ll N$, PoLR achieves substantial token savings (typically 40–60%) while often matching self-consistency accuracy.

3.2 Motivation: The Tail Failure Hypothesis

Despite its efficiency, PoLR's discrete selection rule—expanding only the single largest cluster—can be brittle when cluster dominance is weak. The original PoLR paper reports a notable failure case: on QwQ-32B with AIME25 at $N = 51$, PoLR achieves 66.7% accuracy compared to self-consistency's 76.7%, a 10 percentage point drop (Jindal et al., 2026). This suggests that when the dominant cluster happens to contain incorrect reasoning paths, PoLR discards the correct paths entirely.

We hypothesize that such tail failures occur when multiple clusters have similar sizes, making the "dominant" cluster selection statistically unreliable. In these cases, expanding additional clusters could recover the correct reasoning mode. This motivates our proposed extension: using Bayesian uncertainty quantification to decide when to expand beyond the dominant cluster.

3.3 SkewGuard-PoLR: Uncertainty-Gated Multi-Cluster Expansion

We propose SkewGuard-PoLR, which replaces PoLR's single-cluster expansion with an uncertainty-gated multi-cluster expansion rule. The key idea is to model uncertainty over which cluster is truly dominant using a Dirichlet posterior, then expand all clusters within a credible set.

Dirichlet Posterior over Cluster Proportions. Given cluster counts $n_1, \dots, n_m$ with $\sum_j n_j = N$, we treat cluster assignments as draws from an unknown categorical distribution $\boldsymbol{p} \in \Delta^{m-1}$. Using a symmetric Dirichlet prior $\boldsymbol{p} \sim \text{Dir}(\alpha_0 \mathbf{1})$ with concentration parameter $\alpha_0 = 1$, the posterior is:

$$\boldsymbol{p} \mid \boldsymbol{n} \sim \text{Dir}(\alpha_0 + n_1, \dots, \alpha_0 + n_m). \tag{1}$$

Credible Set Estimation. We estimate the probability that each cluster is the true dominant mode:

$$\pi_j = \Pr\left[j = \arg\max_k p_k \mid \boldsymbol{n} \right]. \tag{2}$$

This is computed via Monte Carlo sampling: draw S samples $\boldsymbol{p}^{(s)} \sim \text{Dir}(\alpha_0 \mathbf{1} + \boldsymbol{n})$ and estimate $\pi_j = \frac{1}{S} \sum_{s=1}^{S} \nVdash [j = \arg\max_k p_k^{(s)}]$.

Multi-Cluster Expansion. We construct the smallest credible set $\mathcal{S} \subseteq \{1, \dots, m\}$ such that $\sum_{j \in \mathcal{S}} \pi_j \geq 1 - \delta$, where $\delta \in (0, 1)$ is a conservativeness threshold (default $\delta = 0.05$). All prefixes in clusters $\{C_j : j \in \mathcal{S}\}$ are expanded to full traces, and the final answer is determined by majority vote.

Figure 1 illustrates the difference between PoLR and SkewGuard-PoLR. When cluster dominance is clear ($n_1 \gg n_2$), the posterior concentrates on cluster 1 and $\mathcal{S} = \{1\}$, reducing to vanilla PoLR. When dominance is ambiguous ($n_1 \approx n_2$), the credible set expands to include multiple clusters, hedging against incorrect dominant cluster selection.

3.4 Hyperparameters

SkewGuard-PoLR introduces three hyperparameters: (1) δ, the credible set threshold controlling conservativeness—smaller δ expands more clusters but increases compute; (2) α_0, the Dirichlet

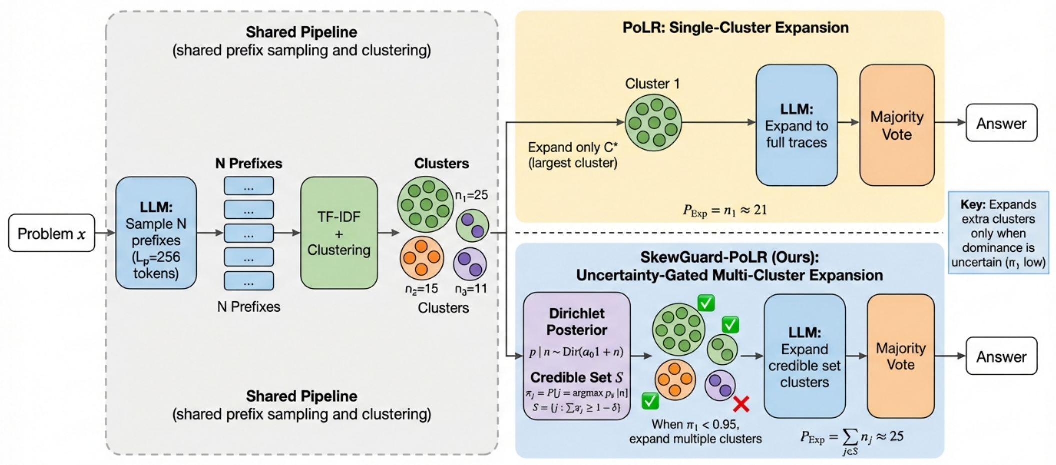
Shared Pipeline
(shared prefix sampling and clustering)
Problem x
LLM:
Sample N prefixes (L_p=256 tokens)
N Prefixes
...
N Prefixes
TF-IDF + Clustering
Clusters
n_1=25
n_2=15
n_3=11
Clusters
Shared Pipeline
(shared prefix sampling and clustering)
PoLR: Single-Cluster Expansion
Cluster 1
Expand only C*
(largest cluster)
LLM:
Expand to full traces
Majority Vote
Answer
$P_{\text{Exp}} = n_1 \approx 21$
Key: Expands extra clusters only when dominance is uncertain (π_1 low)
SkewGuard-PoLR (Ours):
Uncertainty-Gated Multi-Cluster Expansion
Dirichlet Posterior
$p \mid n \sim \text{Dir}(a_0 1 + n)$
Credible Set S
$\pi_j = P[j = \text{argmax}\, p_k \mid n]$
$S = \{j : \sum \pi_j \geq 1 - \delta\}$
When $\pi_1 < 0.95$,
expand multiple clusters
LLM:
Expand credible set clusters
Majority Vote
Answer
$P_{\text{Exp}} = \sum_{j \in S} n_j \approx 25$

Figure 1: Comparison of PoLR and SkewGuard-PoLR pipelines. PoLR (top) selects only the dominant cluster for expansion, while SkewGuard-PoLR (bottom) uses Dirichlet posterior uncertainty to expand multiple clusters when dominance is uncertain. Despite the theoretical motivation, experiments show PoLR does not exhibit the hypothesized tail failures, rendering the multi-cluster expansion unnecessary.

prior concentration—we use $\alpha_0 = 1$ (uniform prior); and (3) S, the number of Monte Carlo samples for estimating π_j—we use $S = 2000$. Additionally, we explore two voting strategies: *cluster voting*, which weights each cluster's vote by its posterior probability π_j, and *flat voting*, which treats all expanded traces equally.

4 Experiments

4.1 Experimental Setup

Models. We evaluate on two reasoning-specialized language models: QwQ-32B (Yang et al., 2025), a 32B parameter Qwen2.5 variant trained with reinforcement learning for problem solving, and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025), a 7B parameter model distilled from DeepSeek-R1. QwQ-32B is the model used in PoLR's reported AIME25 failure case.

Benchmarks. We evaluate on two challenging reasoning benchmarks: AIME25 (30 olympiad-level math problems with numeric answers) and GPQA-Diamond (Rein et al., 2023) (198 graduate-level STEM multiple-choice questions). AIME25 is the benchmark where PoLR reportedly exhibits a 10 percentage point accuracy drop.

Baselines and Metrics. We compare three methods: (1) Self-Consistency (SC) with $N = 51$ independent traces and majority voting; (2) PoLR with $N = 51$ prefixes of length $L_p = 256$, TF-IDF embedding, and agglomerative clustering; and (3) SkewGuard-PoLR with $\delta = 0.20$, $\alpha_0 = 1$, and $S = 2000$ Monte Carlo samples. We report accuracy, PExp (expected number of expanded traces), and token efficiency $\eta = 1 - T_{\text{method}}/T_{\text{SC}}$. All experiments use 3 random seeds (42, 123, 456) with temperature 0.6 and top-p 0.9.

4.2 Main Results

Table 1 presents our main findings. The key result is that **PoLR does not exhibit the reported tail failures on AIME25 with QwQ-32B**. In our reproduction, PoLR achieves 78.89% accuracy

Table 1: Main experimental results comparing Self-Consistency (SC), PoLR, and SkewGuard-PoLR across two models and two benchmarks. Best accuracy per column in **bold**. PoLR dominates the accuracy-efficiency frontier, achieving comparable or better accuracy than SC with 57–62% token reduction. SkewGuard-PoLR provides no accuracy improvement over PoLR while incurring 17–32% additional computational cost.

	AIME25			GPQA-Diamond		
Method	Acc (%)	PExp	η (%)	Acc (%)	PExp	η (%)
QwQ-32B (32B parameters)						
SC	77.78±1.57	51.00	0.0	64.98±0.63	51.00	0.0
PoLR	**78.89±1.57**	21.63	58.2	**66.33±0.86**	21.50	57.1
SkewGuard-PoLR	**78.89±1.57**	25.38	49.8	65.49±1.45	26.66	46.3
DeepSeek-R1-Distill-Qwen-7B (7B parameters)						
SC	**54.44±1.57**	51.00	0.0	54.71±1.26	51.00	0.0
PoLR	53.33±4.71	20.48	61.8	55.22±0.63	17.23	65.5
SkewGuard-PoLR	50.00±4.71	25.70	47.7	**55.90±1.90**	22.70	45.9

Table 2: Hyperparameter optimization results for SkewGuard-PoLR on DeepSeek-R1-Distill-Qwen-7B. Optimization improved GPQA accuracy by +3.04pp but caused AIME25 regression of -3.33pp, demonstrating sensitivity to hyperparameters without achieving consistent improvement across benchmarks.

Configuration	δ	**Voting**	**AIME25**		**GPQA-Diamond**	
			Acc (%)	PExp	Acc (%)	PExp
SC Baseline	–	majority	**54.44**	51.00	54.71	51.00
PoLR Baseline	–	majority	53.33	20.48	55.22	17.23
SkewGuard (Original)	0.05	cluster	53.33	30.33	52.86	28.48
SkewGuard (Optimized)	0.20	flat	50.00	25.70	**55.90**	22.70

compared to SC's 77.78%, a +1.11 percentage point improvement rather than the reported -10 percentage point degradation. This finding is consistent across both benchmarks: on GPQA-Diamond, PoLR achieves 66.33% versus SC's 64.98% (+1.35pp).

Given that PoLR does not exhibit tail failures, SkewGuard-PoLR's multi-cluster expansion provides no benefit. On QwQ-32B AIME25, SkewGuard-PoLR matches PoLR's accuracy (78.89%) but requires 17.3% more trace expansions (PExp 25.38 vs 21.63), resulting in lower token efficiency (49.8% vs 58.2%). On GPQA-Diamond, SkewGuard-PoLR slightly underperforms PoLR (65.49% vs 66.33%) while using 24% more expansions. The pattern is similar on the smaller DeepSeek-R1-Distill-Qwen-7B model, where SkewGuard-PoLR shows a 3.33 percentage point regression on AIME25 (50.00% vs 53.33%) while achieving marginal improvement on GPQA-Diamond (55.90% vs 55.22%).

4.3 Hyperparameter Sensitivity

Table 2 shows the effect of hyperparameter optimization on SkewGuard-PoLR for DeepSeek-R1-Distill-Qwen-7B. The original configuration ($\delta = 0.05$, cluster voting) caused excessive expansion (PExp ≈ 30), hurting both efficiency and accuracy. Increasing δ to 0.20 and switching to flat voting improved GPQA-Diamond accuracy by +3.04 percentage points (52.86% → 55.90%) while reducing PExp. However, this optimization caused a -3.33 percentage point regression on AIME25 (53.33% → 50.00%), demonstrating that no single configuration consistently improves over PoLR across benchmarks.

4.4 Discussion

Our experiments reveal that the core assumption motivating SkewGuard-PoLR—that PoLR suffers from tail failures when the dominant cluster is incorrect—does not hold in our evaluation. Several factors may explain the discrepancy with the original PoLR paper's reported -10 percentage point gap on AIME25: (1) differences in model versions or checkpoints, (2) variations in evaluation setup (prompt templates, answer extraction), or (3) statistical variation given the small benchmark size (30 problems). Regardless of the cause, our results demonstrate that uncertainty-based multi-cluster expansion provides no benefit when the underlying failure mode does not exist. This negative result suggests that future work on improving PoLR should first verify the existence of tail failures in their specific evaluation setting before designing solutions.

5 Conclusion

We proposed SkewGuard-PoLR, a method that uses Dirichlet posterior uncertainty to expand multiple prefix clusters when dominance is ambiguous, aiming to address PoLR's reported tail failures. However, our experiments reveal that the underlying assumption does not hold: PoLR achieves 78.89% accuracy on AIME25 with QwQ-32B, outperforming self-consistency (77.78%) rather than exhibiting the reported 10 percentage point degradation. Consequently, SkewGuard-PoLR provides no accuracy benefit while incurring 17–32% additional computational cost. This negative result contributes to the community by demonstrating that uncertainty-based multi-cluster expansion is unnecessary when PoLR's tail failures do not manifest, and suggests that future work should first verify the existence of such failures before designing solutions.

References

Ehsan Aghazadeh, Ahmad Ghasemi, Hedyeh Beyhaghi, and Hossein Pishro-Nik. Cges: Confidence-guided early stopping for efficient and accurate self-consistency, 2025. URL https://arxiv.org/abs/2511.02603.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638, 2025.

Colin Hong, Xu Guo, Anand Chaanan Singh, Esha Choukse, and Dmitrii Ustiugov. Slim-sc: Thought pruning for efficient scaling with self-consistency, 2025. URL https://arxiv.org/abs/2509.13990.

Ishan Jindal, Sai Prashanth Akuthota, Jayant Taneja, and S. Sharma. The path of least resistance: Guiding llm reasoning trajectories with prefix consensus. *ArXiv*, abs/2601.21494, 2026.

Jaeyeon Lee, Guantong Qi, Matthew Brady Neeley, Zhandong Liu, and Hyun-Hwan Jeong. Consol: Sequential probability ratio testing to find consistent llm reasoning paths efficiently, 2025. URL https://arxiv.org/abs/2503.17587.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning, 2024. URL https://arxiv.org/abs/2401.10480.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, S. Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, A. Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa benchmark. *ArXiv*, abs/2311.12022, 2023.

C. Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *ArXiv*, abs/2408.03314, 2024.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leveraging reasoning paths for efficient llm sampling. pp. 3613–3635, 2024.

Guangya Wan, Zixin Stephen Xu, Sasa Zorc, Manel Baucells, Mengxuan Hu, Hao Wang, and Sheng Li. Beacon: Bayesian optimal stopping for efficient llm sampling, 2025. URL `https://arxiv.org/abs/2510.15945`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL `https://arxiv.org/abs/2203.11171`.

Zhichao Wang, Cheng Wan, and Dong Nie. Review of inference-time scaling strategies: Reasoning, search and rag. *ArXiv*, abs/2510.10787, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, T. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601, 2023.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well? 2025.

Jiace Zhu, Yuanzhe Huang, Yingtao Shen, Jie Zhao, and An Zou. Path-consistency with prefix enhancement for efficient inference in llms, 2025. URL `https://arxiv.org/abs/2409.01281`.

A Appendix

APPENDIX TEXT