Early Signals, Strong Decisions: Prefix-Guided Sampling for Parallel Test-Time Scaling

Jie Ren
MIT CSAIL, MIT FutureTech
rjie@mit.edu

Jonathan Rosenfeld
MIT CSAIL, MIT FutureTech
jonsr@csail.mit.edu

Neil Thompson
MIT CSAIL, MIT FutureTech
neil_t@mit.edu

Abstract

Parallel test-time scaling (TTS) has been shown to enhance reasoning in large language models by generating and aggregating multiple independent reasoning paths. Internal confidence scores can further improve this process by identifying paths of varying quality. However, existing methods remain computationally inefficient, as low-quality paths are fully expanded before being evaluated. In this work, we aim to improve the use of internal confidence signals for better model performance and computational efficiency. We begin by offering a new perspective on the role of confidence in aggregation, showing that it is most impactful for questions near the decision boundary. Notably, when amplified by multi-sample aggregation, even small confidence signals derived from short prefixes can meaningfully influence outcomes in these boundary cases. Building on this insight, we introduce Prefix-Guided Sampling (PreG), a method that reallocates computation by first generating short prefixes, ranking them using internal confidence scores, and then completing only the most promising candidates. This strategy reduces token usage while maintaining gains in path quality. We provide theoretical analysis demonstrating that, under a fixed compute budget, PreG strengthens the effective signal by widening the decision boundary margin. Empirically, our method consistently outperforms standard parallel TTS approaches across benchmarks, while significantly improving token efficiency and reducing overall computation. Our code and dataset is avaiable at github.com/renjie3/PreG.

1 Introduction

Large language models (LLMs) have demonstrated remarkable reasoning capabilities, particularly when augmented with test-time scaling (TTS) techniques that allocate additional computation during inference [1–6]. Among these, parallel scaling has emerged as a powerful paradigm [2, 7–9]: multiple independent reasoning paths are sampled for the same problem, each producing an answer, and the answers are aggregated via mechanisms such as majority voting [2], leading to improved performance.

These methods can be further improved by estimating the confidence of each path and assigning greater weight to high-confidence paths. Recent work has explored various confidence signals, including external supervision such as process reward models (PRMs) [10, 11] and verifier models [12, 13], as well as internal confidence signals [14–21]. The latter is particularly appealing because it requires no additional computation and generalizes across models. For example, DeepConf (DC) uses the concentration of prediction distributions across tokens [14], while Self-Certainty (SCert) utilizes the divergence between vocabulary distributions and a uniform prior at each token position [15].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Despite the effectiveness of internal signal methods, these approaches suffer from a critical inefficiency [22, 23]. Each reasoning path is typically generated to completion before its confidence is assessed, as suggested in prior studies that confidence signals become more reliable in the later tokens of reasoning, closer to the final answer [14, 24]. This inefficiency becomes increasingly pronounced as the number of paths grows, leading to substantial redundant token generation without proportional performance gains.

This raises a fundamental question: *Can we predict the confidence of a reasoning path from its prefix, before it is fully generated?* If feasible, such an approach would substantially reduce the cost of identifying unpromising paths and improve the quality of the final path set. However, this problem is inherently challenging. Early-stage tokens are often incomplete and ambiguous, making prefix-based confidence scores unreliable. As a result, this direction has received limited attention [14].

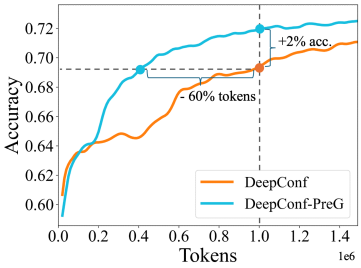


Figure 1: DC-PreG vs. vanilla DC

In contrast to the common view that prefixes are uninformative, we show that even limited prefix information can be effectively leveraged when amplified through the aggregation process in parallel TTS. To verify this, we first provide a deeper understanding of the interaction between confidence scores and aggregation. Aggregation enhances the most probable answer: when correct paths dominate, it reinforces the right answer; when incorrect paths dominate, it amplifies the error. If there is a large gap between the numbers of correct and incorrect paths, confidence scores have little impact. But when counts are similar, confidence becomes critical—even a slight boost to correct answers gets amplified through aggregation. This can accelerate convergence and even reverse the direction of amplification from incorrect to correct paths. Thus, even weak signals may suffice to influence the final outcome through aggregation.

This insight motivates us to ask whether prefix signals, despite their incompleteness, can serve as a useful signal after amplification. Surprisingly, we found that even very short prefixes, covering only around 20% of the full reasoning path, already contain sufficient signal to influence uncertain samples (Section 3), suggesting that early signals may be far more actionable than previously believed. Under a fixed token budget, this is even more efficient than full generation. Building on this, we propose **Prefix-Guided Sampling** (PreG), a compute-efficient framework that reallocates computation during TTS. Instead of expanding all sampled paths to completion, PreG first allocates part of the budget to generate short prefixes, estimates their potential by internal confidence, and selectively continues generation only for a subset of candidates. This approach enables early identification of low-quality paths, reducing unnecessary computation while preserving strong candidates.

Through theoretical analysis, we show that allocating tokens to prefixes remains highly worthwhile, even with fewer full reasoning paths. This makes it an effective and principled mechanism for reallocating compute. Empirically, as shown in Figure 1, our method consistently outperforms vanilla confidence-based parallel TTS approaches [14, 15] in reasoning tasks [25] and achieves substantial gains in token efficiency, reducing token usage by a large margin. Overall, PreG demonstrates that leveraging early prefix signals is not only feasible but also a fundamentally more efficient paradigm for test-time scaling.

2 Related Works

Test-time scaling. Test-time scaling has emerged as a powerful paradigm for enhancing LLM reasoning by allocating additional computation at inference time, without modifying model parameters [1–5]. The methods broadly fall into two complementary categories: sequential scaling and parallel scaling. Sequential scaling extends computation along the depth dimension. by generating longer, iteratively refined reasoning chains. Chain-of-Thought prompting [1] generates longer by eliciting step-by-step reasoning, while other works like Self-Refine [26] encouraged models to revise their outputs through internal feedback. Parallel scaling improves performance along the width dimension by sampling multiple independent reasoning paths and aggregating their outputs [2, 7–9].

Internal confidence. Internal confidence is increasingly used to improve reasoning. A simple source is P(True), obtained by directly querying the model [27, 28]. Beyond this, more stable estimates aggregate over the full token sequence, using metrics such as likelihood [29], Gini Impurity [30, 15],

entropy [31], DC [14], SCert [15], Tokur [21] and DTR [19]. In addition, CoRefine proposes treating the token confidence sequence as time-series data and training a classifier on top of it [24].

Efficient TTS. Prior work improves TTS efficiency via early stopping, either based on the generated paths themselves—such as ASC [32], ESC [33], and CGES [34]—or through inter-path comparison, where methods like Slim-SC [35] and [23] exploit latent consistency among early paths to prune candidates. However, these approaches require sometimes partially sequential sampling or dynamic decision-making, limiting system-level scalability, and often sacrifice accuracy in the process.

3 When Confidence Scores Matter: Aggregation Amplifies Small Gains

In this section, we first formalize the problem, then discuss how the role of confidence scores and why prefix confidences serve as effective confidence estimates.

3.1 Problem Statement and Definitions

Given an input problem Q and an LLM, we consider solving the problem by parallel test-time scaling methods that independently generate a set of reasoning paths under a fixed token budget B.

Parallel testing-time scaling. Given Q, we define a path generated by the model as $S = (s_1, s_2, \ldots, s_T)$, where each token $s_t \in \mathcal{V}$, and T denotes the sequence length. Under a fixed token budget B, a parallel TTS method produces a set of reasoning paths $\mathcal{S}_B = \{S_1, S_2, \ldots, S_M\}$, where M is the number of complete path and depends on how the budget is allocated. (If all the budget is used for full path generation, we denote the path number as $M_{\text{base}} = B/T$.) Each complete path yields an answer y_i, and a final answer is obtained via an aggregation function: $\hat{y} = A(y_1, y_2, \ldots, y_M)$. Let y^* denote the ground-truth answer. The objective is to maximize the probability of correct prediction (i.e., $\hat{y} = y^*$) under a fixed token budget B.

Confidence-based parallel TTS. In this work, we focus on internal confidence scores without external models or verifiers, and aggregate answers via weighted voting:

$$\hat{y} = \arg\max_y W(S_i) \sum_{i=1}^{M} \mathbf{1}[y_i = y],$$

where $W(\cdot)$ is the confidence-based aggregation function. **Self-Consistency** (SC) [2] is a special case where $W(\cdot) = 1$ (i.e. unweighted majority voting), which is the most foundamental method.

Margin. By denoting the aggregated score as $\mathrm{Conf}(y) = W(S_i) \sum_{i=1}^{M} \mathbf{1}[y_i = y]$ and the strongest incorrect answer as $y_w = \arg\max_{y \neq y^*} \mathrm{Conf}(y)$, we define the margin as

$$\Delta = \mathrm{Conf}(y^*) - \mathrm{Conf}(y_w) = \sum_{i=1}^{M} Z_i, \tag{1}$$

where $Z_i = W(S_i)\big(\mathbf{1}[y_i = y^*] - \mathbf{1}[y_i = y_w]\big)$ is the per-path contribution. The prediction is correct iff $\Delta > 0$: $\Pr(\hat{y} = y^*) = \Pr(\Delta > 0)$.

3.2 A Margin-Based Analysis of Aggregation in Parallel TTS

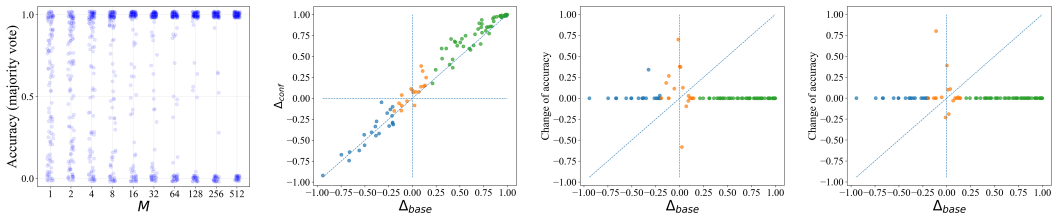
In this subsection, we first investigate what factors influence the final aggregated accuracy. The above margin decomposition $\Delta = \sum_{i=1}^{M} Z_i$ reveals that the aggregated outcome in parallel TTS depends on how per-path contributions accumulate. To characterize this process, we analyze how prediction accuracy relates to the statistical properties of these contributions, summarized in the following theorem which serves as the theoretical foundation for our subsequent analysis.

Proposition 1 (Accuracy via margin)**.** *Assume that $\{Z_i\}_{i=1}^{M}$ are independent, bounded random variables with $\mathbb{E}[Z_i] = \mu, \mathrm{Var}(Z_i) = \sigma^2$. Then, for large M, by the central limit theorem, the margin admits the Gaussian approximation*

$$\Delta \approx \mathcal{N}(M\mu, M\sigma^2),$$

and the prediction accuracy is approximated by

$$\Pr(\hat{y} = y^*) \approx \Phi\Big(\sqrt{M}\,\mu/\sigma\Big).$$



(a) SC (b) SCert-Borda (c) SCert-Borda (d) SCert-Borda (prefix)

Figure 2: Studies on the interaction between confidence scores and aggregation using Qwen3-8B [36] (a) shows problem accuracy distribution of SC as M increases. The point is jittered to show density. Prefix length in (d) is 4096 which is 22% of full path length in (b)(c). (b)(c)(d) fixes $M = 200$.

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.

Proof. A complete proof is provided in Appendix A.1. □

Proposition 1 shows that accuracy is determined by the signal-to-noise ratio (SNR) $\sqrt{M}\mu/\sigma$, where μ is the expected per-path advantage of correct answer and $\sqrt{M}$ reflects the amplification of aggregation. Assuming σ approximately fixed for simplicity, accuracy improves monotonically with M and μ. Crucially, the direction of amplification is governed by the sign of μ: when $\mu > 0$, aggregation reinforces the correct answer; when $\mu < 0$, it amplifies the strongest incorrect answer instead.

To verify this, we run experiments using SC in Figure 2a. Each point represents the estimated accuracy of a single problem under M aggregated paths, where accuracy is estimated by repeating the aggregation 100 times. As M increases, per-problem accuracies polarize toward either 1.0 or 0, consistent with the sign-driven amplification predicted by the theorem.

3.3 Confidence Scores Are Most Effective Near the Decision Boundary

As shown above, accuracy is governed by μ and M. Obviously, μ varies across problems: for example, easier problems tend to have positive μ, while harder ones can have negative μ. We denote μ under SC as μ_{base}. After using confidence scores μ_{base} is shifted to μ_{conf}. In this subsection, we examine how confidence scores influence μ and for how this shift influences TTS.

Confidence scores are most effective near the decision boundary. Figure 2b plots μ_{conf} against μ_{base} for individual problems. For problems with positive or near-zero μ_{base} (green and orange points), confidence scores consistently increase μ, with some near-zero points even flipping sign. For strongly negative $\mu_{\text{base}} < -0.2$ (blue), the shift is noisy with no clear trend.

Nevertheless, whether μ improves or worsens in this region has little consequence for accuracy. As shown in Figure 2c, which reports the change in accuracy after introducing confidence scores across different values, problems with $\mu_{\text{base}} < -0.2$ or $\mu_{\text{base}} > 0.2$ show almost no change in accuracy. This is because when $|\mu_{\text{base}}|$ is large, aggregation over even a moderate M is sufficient to converge to the dominant answer, and the outcome becomes insensitive to shifts in μ.

In contrast, for problems near the decision boundary (i.e., μ_{base} is around 0), the shift in μ is important in two ways. First, it enlarges the small positive margins, making the correct answer more stable under the noise in sampling process. Second, it can flip the sign of μ, reversing the direction of amplification from incorrect to correct. These boundary problems are exactly where confidence scores provide the greatest benefit. As confirmed in Figure 2c, accuracy improvements after introducing confidence scores are concentrated among problems with $\mu_{\text{base}} \approx 0$. More importantly, even the shift is small, the aggregation over M paths can amplify it.

Prefix signals are sufficient near the boundary. Since small shifts in μ can be disproportionately amplified by aggregation, a natural question is whether prefix-based confidence scores can provide sufficient signal. Prefixes carry less information than full paths (see Appendix D.1 for a quantitative analysis). Nevertheless, the experiment in Figure 2d shows that they can achieve competitive influence to accuracy after aggregation. Using the same experimental setup as Figure 2c, we replace the full-path confidence score with prefix score. Full-path score improves accuracy by 1.71% on average,

while prefix score achieves 0.91%. The prefix recovers **53%** of the accuracy gain from full-path scoring, but only uses **22%** of the tokens. This suggests that prefix signals are not only actionable but also substantially more token-efficient than full-path generation. This motivates the token reallocation framework we introduce in the following Section 4.

4 Prefix-Guided Sampling

Building on the above analysis, we propose a principled framework for parallel test-time scaling that reallocates tokens toward prefix exploration.

4.1 Method Framework

We propose **Prefix-Guided Sampling** (PreG), a simple yet effective framework that is both model-agnostic and score-agnostic, consistently improving performance across aggregation strategies. As shown in Figure 3, PreG operates in two stages: prefix generation and path completion.

Stage 1: Prefix generation. We first generate M_{prefix} prefixes in parallel, each of length k. For each prefix, we compute an internal confidence score. (All the definitions of the confidence score we tested in this work can be found in Appendix B. We additionally introduce a new confidence score, *Conf-Lens*, derived from the logit lens [37].) The prefixes are then ranked by confidence scores to identify the most potential candidates.

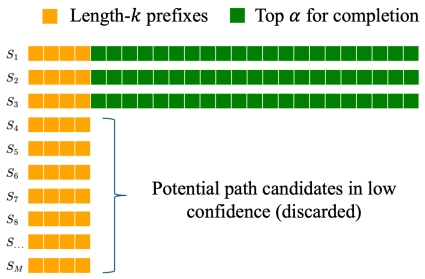


Figure 3: PreG framework

Stage 2: Path selection and completion. We select the top fraction α of prefixes and continue generation to complete paths of length T, yielding $M_{\text{comp}} = \alpha M_{\text{prefix}}$ complete paths. Under the token budget B, the prefix stage consumes $M_{\text{prefix}} \cdot k$ tokens and the completion stage consumes $M_{\text{comp}} \cdot (T - k)$ tokens, giving

$$M_{\text{comp}} = \frac{\alpha B}{k + \alpha(T - k)}.$$

These complete paths are then aggregated via

$$\hat{y} = \arg\max_{y} \sum_{i=1}^{\alpha M_{\text{prefix}}} W(S_i^{\text{comp}}) \cdot \mathbf{1}[y_i = y],$$

where $W(\cdot)$ is determined by the confidence score of the completed path S_i^{comp} and strategies such as majority voting, weighted voting, Borda voting [15], or top-10% voting [14, 21].

Why PreG works. Compared to vanilla parallel TTS, PreG improves μ in two ways: it filters out low-quality prefixes, and it explores a larger pool of $M_{\text{prefix}} > M_{\text{base}} = B/T$ candidates before selecting the best α fraction. The cost is fewer completed paths. However, this is naturally mitigated by the aggregation mechanism. When $|\mu_{\text{base}}|$ is large, aggregation already converges to the dominant answer quickly, so the reduction in M has little effect on accuracy. When $\mu_{\text{base}} \approx 0$, the improvement in μ is amplified, sometimes even flipping the direction of amplification.

System efficiency. Beyond algorithmic gains, PreG also offers practical advantages at the system level: all prefixes are generated in parallel to a fixed length, requiring no dynamic stopping conditions, cross-path comparisons, or prefill operations, making it better suited for parallel system design.

Finite-sample regime. When M is small, the signal-to-noise ratio $\sqrt{M}\mu/\sigma$ can be close to zero even when $\mu > 0$, meaning aggregation is dominated by sampling noise. In this regime, reducing M through filtering can cause a large accuracy drop unless the gain in μ is sufficiently large. Empirically, however, our method already outperforms the vanilla baseline at values of M well below those typical in standard parallel sampling setups.

4.2 Analysis on k and α

Proposition 1 shows that accuracy is governed by the signal-to-noise ratio (SNR) $S = \sqrt{M_{\text{comp}}}\mu/\sigma$. Under a fixed budget B, both M and μ depend on the hyperparameters α and k, creating a trade-off: increasing α yields more complete paths but weaker filtering, while increasing k improves signal quality but reduces M_{comp}. The following propositions characterize the optimal choices.

Optimal prefix length k. Assume $\mu(k) = \mu_0 + \beta k$ grows linearly with prefix length. Then $V(k) = \sqrt{M(k)}\,\mu(k)$ has no interior maximum over $k \in [k_{\min}, k_{\max}]$, and the optimal prefix length satisfies $k^* \in \{k_{\min}, k_{\max}\}$. A complete derivation is provided in Appendix A.2.

For simplicity, we assumed $\mu(k) = \mu_0 + \beta k$ grows linearly with prefix length. However, as shown in Appendix D.1, the signal grows very slowly and is near zero for the first 1000–2000 tokens, suggesting that $k_{\min}$ should not be set too small. In this work, the smallest k we evaluate is 4000 tokens.

Optimal filtering ratio α. Under a fixed budget B, increasing α yields more complete paths but admits more low-quality ones, while decreasing α improves path quality but reduces M_{comp}. Taking $\alpha \to 0$ collapses the SNR regardless of μ; taking $\alpha = 1$ reduces to the vanilla method. Unless the vanilla method outperforms PreG, the optimal α^* lies between the two extremes.

These theoretical findings are consistent with our empirical results in Section 5.4.

5 Experiments

In this section, we evaluate PreG across multiple reasoning benchmarks, confidence scores and aggregation strategies. We further ablate the finite-sample regime, analyze the sensitivity to the two key hyperparameters, and evaluate system-level efficiency. We first present the experimental settings.

LLMs and Datasets. We evaluate two reasoning models: Qwen3-8B (Qwen) [36] and Llama-3.1-Nemotron-Nano-4B-v1.1 (Llama) [38]. We benchmark on four math competition datasets [25], AIME25, BRUMO25, HMMT25-Feb, and CMIMC25, as well as GPQA-Diamond [39], a dataset spanning biology, physics, and chemistry.

Baselines. We consider vanilla parallel TTS methods, early-stopping methods and PRM. The first category includes Self-Consistency and six methods based on six token-level confidence scores, likelihood (LL) [29], entropy [31], Gini Impurity [30, 15], Diff-token, Conf-Lens, DeepConf (DC) [14], and Self-Certainty (SCert) [15]. They are combined with four aggregation strategies, majority voting, weighted voting, Borda voting [15] and top-10% voting (W-10%) [14, 21]. Early-stopping methods include ESC [33], ST-BoN [23], and CGES [34]. The PRM method uses weighted voting. The PRMs are Qwen2.5-Math-PRM-7B [40] for math, and Skywork-Reward-V2-Qwen3-8B [41] for GPQA-D.

Implementation and evaluation. We pre-generate 512 paths per problem at temperature $= 1.1$ on H200s. For each evaluation, we sample from these paths without replacement over 100 random seeds to simulate the TTS process. Confidence scores in vanilla methods are computed over the last 2048 tokens of each path; prefix scores in PreG use the first 4096 tokens ($\sim$22% of Qwen output and $\sim$18% of Llama output). We set $\alpha = 0.3$ for all PreG configurations. In this setting, we have $M_{\text{prefix}} = 2.2M_{\text{base}}$ and $M_{\text{comp}} = 0.66M_{\text{base}}$ for Qwen, and $M_{\text{prefix}} = 2.31M_{\text{base}}$ and $M_{\text{comp}} = 0.69M_{\text{base}}$ for Llama. Performance of a parallel TTS method is reported as accuracy averaged over 100 attempts. Effectiveness of a confidence score is measured by AUROC.

5.1 Main Results

We first compare methods under a fixed token budget, evaluating at three budget levels, 100k, 300k, and 1000k tokens, where the corresponding M_{base} is approximately 5.4, 16.2, and 54.1 for Qwen and 4.6, 13.8, and 46.1 for Llama. Due to space constraints, the main results include three representative confidence scores (LL, DC, and SCert) and the two best-performing aggregation strategies among the baselines: Borda voting and W-10%. Borda voting assigns weights based on score rankings; W-10% retains the top 10% of M_{base} paths and applies weighted voting. PreG uses the same aggregation strategies. For W-10% in PreG, this results in two rounds of path filtering: the first uses the prefix

Table 1: Accuracy under fixed budgets (100k, 300k, 1000k tokens). Results exceeding and falling below the corresponding vanilla baseline are underlined in green and red, respectively. The best result per dataset-budget combination is shown in **bold**.

QWen3-8B		AIME25 (100k/300k/1000k)	BRUMO (100k/300k/1000k)	HMMT (100k/300k/1000k)	CMIMC (100k/300k/1000k)	GPQA-D (100k/300k/1000k)
Pass@1		67.94	69.64	45.30	46.14	58.90
Self-Consistency		76.53 / 78.43 / 79.40	76.93 / 78.70 / 79.90	52.37 / 56.10 / 57.33	52.78 / 55.50 / 56.33	68.90 / 69.87 / 70.75
PRM		78.17 / 78.43 / 78.97	78.17 / 78.93 / 80.27	56.60 / 57.00 / 58.10	55.17 / 55.53 / 56.55	67.88 / 68.43 / 69.41
ESC		**78.33** / 78.77 / **81.00**	77.87 / 79.47 / 80.73	55.87 / 57.13 / 58.03	55.10 / 55.85 / 56.23	69.24 / 70.17 / NA
ST-BoN		75.40 / 78.70 / 80.60	75.13 / 78.00 / 79.73	49.40 / 53.73 / 55.13	49.48 / 54.65 / 56.13	67.77 / 67.34 / 68.92
CGES		70.43 / 75.10 / NA	75.20 / 78.17 / NA	51.40 / 53.13 / NA	52.18 / 54.10 / NA	NA / NA / NA
Borda	LL	75.30 / 77.87 / 79.33	75.87 / 78.87 / 79.27	50.93 / 55.73 / 56.37	50.25 / 55.30 / 57.65	69.61 / 70.12 / 70.81
	LL-PreG	74.90 / 77.43 / 79.30	78.43 / 81.43 / 82.77	53.27 / 57.80 / 60.00	51.42 / 56.95 / 58.43	69.07 / 69.39 / 69.67
	DC	76.27 / 78.43 / 78.83	77.43 / 80.33 / 81.63	55.37 / 59.00 / 60.47	53.65 / 58.87 / 60.72	69.58 / 70.12 / 70.58
	DC-PreG	76.20 / **79.07** / 79.93	79.77 / 82.07 / 83.03	56.63 / 59.90 / 61.47	**55.45** / 60.68 / 63.35	69.63 / 69.95 / NA
	SCert	76.33 / 78.33 / 78.70	77.77 / 80.93 / 82.33	56.10 / 58.87 / 60.03	54.47 / 58.73 / 60.15	69.46 / 69.81 / 70.32
	SCert-PreG	76.40 / 78.67 / 80.60	**80.73** / **82.70** / 83.40	**57.10** / 59.00 / 60.20	55.20 / **60.78** / 62.55	**69.62** / **70.05** / **70.45**
W-10%	LL	67.83 / 65.60 / 73.10	70.60 / 69.60 / 75.43	46.47 / 46.37 / 51.43	46.00 / 44.62 / 48.98	66.63 / 67.89 / 68.36
	LL-PreG	67.03 / 71.30 / 76.27	72.57 / 76.97 / 81.23	48.50 / 52.40 / 57.60	46.52 / 47.80 / 55.20	66.79 / 68.41 / 68.71
	DC	72.70 / 71.90 / 76.50	74.87 / 75.70 / 80.93	53.53 / 55.60 / 60.23	52.68 / 54.43 / 58.95	67.27 / 68.51 / 69.49
	DC-PreG	71.63 / 75.70 / 78.57	76.73 / 80.20 / 82.07	52.93 / 58.73 / 61.37	51.18 / 56.73 / 64.72	67.03 / 69.03 / 69.83
	SCert	72.37 / 72.53 / 76.80	76.00 / 77.47 / 80.77	54.53 / 57.53 / 60.93	54.30 / 55.33 / 59.55	67.57 / 68.44 / 69.28
	SCert-PreG	72.40 / 75.73 / 78.70	77.80 / 81.50 / **83.47**	53.50 / **60.53** / **62.67**	52.90 / 58.77 / **66.17**	67.18 / 68.92 / 69.61
Llama-Nemotron-4B		AIME25 (100k/300k/1000k)	BRUMO (100k/300k/1000k)	HMMT (100k/300k/1000k)	CMIMC (100k/300k/1000k)	GPQA-D (100k/300k/1000k)
Pass@1		56.41	63.76	41.11	37.88	52.51
Self-Consistency		65.73 / 68.93 / 70.10	66.60 / 66.40 / 65.90	44.53 / 47.87 / 48.23	42.10 / 48.10 / 52.35	59.81 / 60.34 / 59.78
PRM		67.63 / 69.30 / 69.97	67.20 / 66.50 / 65.97	47.77 / 48.30 / 48.50	46.40 / 47.75 / 51.15	60.82 / 61.43 / 59.36
ESC		67.60 / 68.83 / 70.53	67.33 / 66.60 / 65.03	47.27 / 47.50 / 48.20	**46.72** / 47.98 / 53.10	58.18 / 58.70 / 59.63
ST-BoN		63.80 / 68.03 / 68.80	66.53 / 66.77 / 65.50	43.80 / 47.30 / 46.77	40.65 / 45.45 / 49.53	58.55 / 58.97 / 59.54
CGES		61.00 / 65.10 / NA	63.73 / 62.47 / NA	42.87 / 45.00 / NA	42.95 / 44.27 / NA	NA / NA / NA
Borda	LL	67.23 / 71.10 / 72.23	67.87 / 68.07 / 67.63	47.33 / 50.37 / 52.10	43.48 / 49.00 / 53.30	60.48 / 61.59 / 61.90
	LL-PreG	67.97 / 71.13 / 73.50	68.47 / **69.47** / 70.30	47.73 / **50.93** / 53.37	43.60 / 48.60 / 53.02	**61.43** / **62.56** / 62.50
	DC	67.50 / **71.83** / 73.13	67.60 / 68.00 / 67.47	47.87 / 50.67 / 52.03	43.80 / 48.98 / 53.57	60.39 / 61.47 / 61.77
	DC-PreG	68.20 / 71.47 / 74.63	68.00 / 69.10 / 69.23	**48.57** / 50.60 / 53.13	44.82 / **49.75** / **54.15**	61.22 / 62.18 / **62.23**
	SCert	67.73 / 71.73 / 73.30	67.73 / 67.93 / 67.33	47.47 / 50.70 / 51.63	43.40 / 49.15 / 53.50	60.21 / 61.11 / 61.27
	SCert-PreG	**68.60** / 71.70 / **74.93**	**68.57** / 68.80 / 68.17	48.50 / 50.77 / 52.50	44.90 / 49.55 / 53.53	60.66 / 61.57 / 61.47
W-10%	LL	63.87 / 64.37 / 70.80	66.40 / 67.07 / 70.67	45.80 / 47.90 / 50.17	42.53 / 43.40 / 48.90	59.83 / 60.27 / 62.89
	LL-PreG	62.53 / 67.63 / 74.73	67.10 / 68.57 / **71.50**	45.17 / 49.50 / **53.63**	41.68 / 43.87 / 49.50	59.93 / 62.21 / 63.50
	DC	64.83 / 66.33 / 72.27	67.20 / 67.27 / 70.53	47.20 / 49.33 / 51.70	43.10 / 44.65 / 50.75	60.05 / 60.71 / 63.15
	DC-PreG	63.87 / 69.07 / 73.30	67.20 / 68.60 / 71.13	46.23 / 50.90 / 53.20	42.93 / 46.57 / 52.38	59.86 / 62.09 / 63.26
	SCert	65.47 / 67.23 / 71.83	67.20 / 67.47 / 70.17	46.90 / 49.97 / 52.93	42.63 / 44.40 / 51.50	59.77 / 60.87 / 62.49
	SCert-PreG	63.90 / 69.20 / 73.67	66.57 / 69.30 / 70.33	46.60 / 49.97 / 52.87	42.00 / 46.05 / 51.80	59.69 / 61.91 / 62.15

score to retain M_{comp} paths, and the second uses the full score to retain the same number as the vanilla baseline (10% of M_{base}), ensuring a fair comparison. We report the results in Table 1.

Overall, our method achieves strong and consistent improvements across datasets and models. Specifically, (1) our approach attains the best performance among all compared methods on the majority of datasets and model configurations, and (2) it yields stable and significant gains over the corresponding vanilla confidence-based baseline. We now discuss from the following perspectives.

Performance Across Different Datasets. Our method improves performance on all datasets, with gains particularly stable when the token budget exceeds 300k (smaller budgets are analyzed in Section 5.5). (1) Among the 15 dataset–budget combinations, PreG achieves the best result in 13 cases under Qwen and 14 cases under LLaMA. Improvements can be substantial: on CMIMC (1000k budget), SCert-PreG (W-10%) outperforms all baselines by 5.4% under Qwen, where SC itself gains only $\sim$10% over Pass@1. (2) Across all settings, our method consistently outperforms the corresponding vanilla confidence methods by a large margin.

Different Voting Methods. PreG improve both Borda and W-10% but exhibit different behaviors. PreG with Borda is more stable across budgets, outperforming the vanilla confidence baselines in most configurations and yielding at least 2% improvement on Qwen in most cases. PreG with W-10% performs poorly under small budgets—roughly matching the baseline at 100k for Qwen and even degrading slightly for LLaMA. But it is more effective at 1,000k budgets, achieving the best result among all methods in 5 of 8 dataset–model math combinations. We attribute this to the aggressive filtering of W-10%: at 100k, only 1 path survives to vote, making results highly unstable. At large budgets, this aggressiveness becomes an advantage: if correct paths are the majority, discarding low-confidence paths does not hurt; if incorrect paths dominate, the filtering is more likely to remove

Table 2: Token efficiency. Underline colors indicate >50% token savings, 0–50% token savings, and no savings.

Qwen3	Score	AIME25	BRUMO	HMMT	CMIMC	GPQA-D
Borda	LL	144.25%	41.83%	46.64%	50.87%	NA
	DC	57.98%	20.50%	62.53%	54.90%	121.10%
	SCert	55.07%	23.07%	63.08%	46.83%	61.76%
W-10%	LL	35.27%	33.87%	25.89%	33.64%	55.26%
	DC	33.25%	44.09%	45.95%	47.92%	58.27%
	SCert	54.18%	49.66%	51.29%	47.81%	62.54%
Llama	Score	AIME25	BRUMO	HMMT	CMIMC	GPQA-D
Borda	LL	44.11%	29.36%	57.76%	114.34%	19.02%
	DC	54.25%	29.23%	68.82%	84.91%	20.55%
	SCert	42.16%	29.43%	53.62%	91.52%	26.02%
W-10%	LL	47.65%	59.69%	32.04%	62.70%	56.50%
	DC	56.18%	40.28%	73.58%	67.53%	71.65%
	SCert	54.33%	42.47%	91.36%	78.13%	NA

Table 3: Confidence signal effectiveness.

Qwen	Math	GPQA-D
PRM	0.6148	0.5122
LL	0.5462	0.5660
LL-PreG	0.5345	0.5055
DC	0.7260	0.5651
DC-PreG	0.5415	0.5481
SCert	0.7409	0.5814
SCert-PreG	0.5471	0.5359
Llama	Math	GPQA-D
PRM	0.5959	0.5532
LL	0.6229	0.5670
LL-PreG	0.5151	0.5394
DC	0.6576	0.5785
DC-PreG	0.5241	0.5360
SCert	0.6624	0.5811
SCert-PreG	0.5187	0.5278

them. Thus, at large budgets, W-10% can amplify the signal from high-quality paths more effectively. We also report the results of other aggregation in Appendix D.3.

Different Confidence Scores. PreG achieves better absolute performance when paired with stronger confidence scores, but yields more pronounced relative improvements on weaker scores. DC and SCert are the two strongest confidence scores, and the best overall results concentrate around these methods. For weaker scores (represented by LL), PreG consistently closes the gap with vanilla DC and SCert, often matching DC-PreG and SCert-PreG. The results of other scores are in Appendix D.3.

Comparison with early stopping and PRM-based methods. PreG outperforms all three early stopping baselines, demonstrating more effective utilization of the token budget. Our method also surpasses the PRM baseline evaluated in our experiments.

5.2 Token Efficiency

In Table 2, we compare token efficiency against the vanilla confidence methods. We measure efficiency as the ratio of tokens required by PreG versus the corresponding vanilla method at a target accuracy. The target accuracy is defined as the accuracy achieved by the vanilla method at a 1,000k token budget. (If the vanilla method saturates before 1000k tokens, we use the actual token budget at which the target accuracy is first reached.)

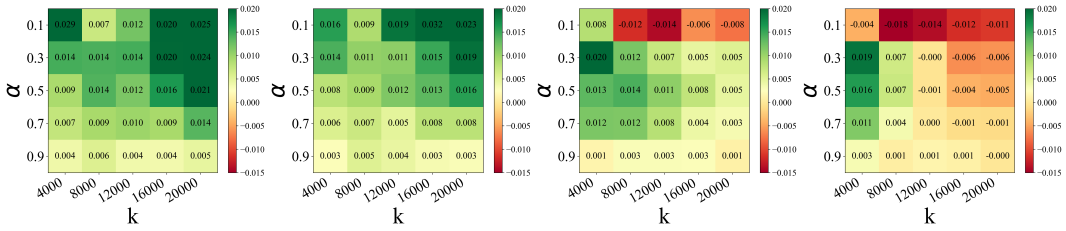
Across all 60 configurations, PreG requires fewer tokens than the vanilla method in 54 cases, and in 27 of these requires less than 50% of the tokens. Token savings are most pronounced on BRUMO, where up to $\sim$80% of tokens can be saved. On mathematical datasets, Qwen saves over 40% in all but one configuration (AIME, Borda, LL); LLaMA similarly achieves over 40% savings in the majority of cases. Overall, these results demonstrate PreG's effectiveness as a token-efficient inference strategy.

5.3 Confidence Signal Effectiveness

Table 3 compares the AUROC of different confidence scores, where "Math" denotes averaged results across four mathematical datasets. AUROC is first computed within each problem. The results are averaged over problems that are neither fully correct nor fully incorrect.

Comparing full generation versus prefix, prefix confidence is substantially weaker: SCert achieves an AUROC of 0.7409 on full-length paths but only 0.5471 on prefixes. Combined with the strong performance gains reported earlier, this supports our analysis that parallel TTS amplifies the signal near the decision boundary—demonstrating that even weak prefix signals can be effectively leveraged.

On GPQA-Diamond, AUROC scores are smaller than math data, which explains why adding confidence signals sometimes degrades performance. This confirms a key limitation of our method: its effectiveness is strongly tied to the quality of the underlying confidence signal. However, when that signal is reliable, our method substantially improves both performance and token efficiency.



(a) DC (Borda) (b) SCert (Borda) (c) DC (W-10%) (d) SCert (W-10%)

Figure 4: Effect of k and α on Accuracy (Qwen, Math Datasets)

Table 4: Multi-GPU inference efficiency on four H200 GPUs using Qwen3-8B.

Method	Traces per problem	Tokens (M)	Latency (min)	Avg. KV (%)	Peak KV (%)
DC-Vanilla	32	6.43	58.5	37.2	93.8
DC-PreG	$80 \rightarrow 24$	6.95	**40.6**	**31.3**	**74.7**

5.4 Hyperparameters

We analyze the effect of varying α and k. Consistent with our theoretical analysis, the optimal α should lie away from 0 and 1, and the optimal k tends to occur at the boundary of the search range. In Figure 4, we evaluate $k \in \{4\text{k}, 8\text{k}, 12\text{k}, 16\text{k}, 20\text{k}\}$ and $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ Each cell reports the change in accuracy relative to the vanilla, with positive values indicating improvement.

Effect of k. Across all four figures, the optimum occurs at either k_{min} or k_{max}. However, k_{max} is unstable—as shown in Figures 4c–4d—and the maximum possible k varies unpredictably across samples, so we favor k_{min}. **Effect of α.** For Borda (Figures 4c–4d), $\alpha = 0.3$ yields the largest gains. For W-10% (Figures 4a–4b), $\alpha = 0.1$ performs best; we did not explore finer granularity below 0.1, because $\alpha \rightarrow 0$ is not desirable as Section 4.2. The key finding is that larger $\alpha \rightarrow 1$ is always worse, confirming that prefix-based path selection via PreG is consistently beneficial.

5.5 Small budget B

To better illustrate how performance varies with budget, we plot the detailed accuracy curves in Figure 5, averaged across four math datasets.

For SCert, our method consistently outperforms the baseline across all budgets. For W-10%, our method shows clear improvement beyond approximately 120k tokens for Qwen ($M_{\text{base}} \approx 6.48$) and 236k tokens for LLaMA ($M_{\text{base}} \approx 10.92$). These path counts are well below those used in typical parallel TTS settings. Also, as the curves show, the region below these budgets corresponds to the fastest phase of accuracy growth, where practitioners would rarely stop. Moreover, below these thresholds, PreG drops only marginally below the vanilla baseline.

An interesting observation is that the vanilla W-10% exhibits a pronounced plateau between 200k–400k tokens where accuracy grows slowly, whereas our method largely eliminates this plateau. This directly supports our motivation for allocating budget to explore more prefixes: by surfacing high-quality candidates earlier, PreG accelerates convergence and removes the stagnation region.

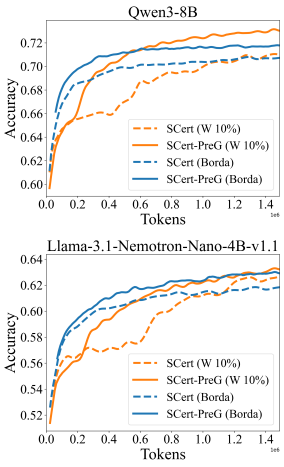


Figure 5: Tokens vs. Acc.

5.6 System-level Inference Efficiency

We further evaluate the system-level efficiency of PreG in a multi-GPU inference setting. Specifically, we use Qwen3-8B with vLLM on four NVIDIA H200 GPUs, maintaining the same distributed configuration for both methods.

As shown in Table 4, DC-PreG reduces end-to-end latency by 30.6%, from 58.5 to 40.6 minutes. Average KV-cache utilization decreases from 37.2% to 31.3%, while peak utilization decreases from 93.8% to 74.7%. Notably, these improvements are achieved despite generating 8.1% more tokens overall.

This improvement is consistent with PreG's sequence-length allocation: it explores more short prefixes but completes fewer long trajectories, reducing long-lived KV-cache occupancy and mitigating long-tail decoding overhead. These results demonstrate that PreG's compute-allocation benefits can translate into improved latency and KV-cache efficiency in multi-GPU inference.

6 Conclusions

We proposed Prefix-Guided Sampling (PreG), a compute-efficient framework for parallel TTS that generates short prefixes, filters low-quality candidates via internal confidence scores, and completes only the most promising paths. Theoretically, we show that weak prefix signals can be amplified through aggregation near the decision boundary. Empirically, PreG consistently outperforms vanilla confidence-based TTS methods while reducing token usage by up to 80%.

Broader Impacts and Limitations. This work may also enable more scalable misuse of LLMs, such as large-scale harmful content generation. Our method's effectiveness is bounded by the quality of the underlying confidence signal, limiting gains when the signal is weak. Under very small token budgets, PreG may not outperform the vanilla baseline.

Acknowledgments and Disclosure of Funding

This work was supported by Coefficient Giving. The authors have no competing interests to declare.

References

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[2] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

[3] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[5] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the association for computational linguistics: ACL 2023*, pages 1049–1065, 2023.

[6] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, 2025.

[7] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

[8] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[9] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for llm problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

[11] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[13] Yegon Kim, Seungyoo Lee, Chaeyun Jang, Hyungi Lee, and Juho Lee. Parallel test-time scaling with multi-sequence verifiers. *arXiv preprint arXiv:2603.03417*, 2026.

[14] Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.

[15] Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[16] Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, et al. Efficiently scaling llm reasoning with certaindex. *arXiv preprint arXiv:2412.20993*, 2024.

[17] Anton Korikov, Pan Du, Scott Sanner, and Navid Rekabsaz. Batched self-consistency improves llm relevance assessment and ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32675–32691, 2025.

[18] Joshua Ong Jun Leang, Zheng Zhao, Aryo Pradipta Gema, Sohee Yang, Wai-Chung Kwan, Xuanli He, Wenda Li, Pasquale Minervini, Eleonora Giunchiglia, and Shay B Cohen. Picsar: Probabilistic confidence selection and ranking for reasoning chains. *arXiv preprint arXiv:2508.21787*, 2025.

[19] Wei-Lin Chen, Liqian Peng, Tian Tan, Chao Zhao, Blake JianHang Chen, Ziqian Lin, Alec Go, and Yu Meng. Think deep, not just long: Measuring llm reasoning effort via deep-thinking tokens. *arXiv preprint arXiv:2602.13517*, 2026.

[20] Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for llm reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[21] Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, et al. Tokur: Token-level uncertainty estimation for large language model reasoning. *arXiv preprint arXiv:2505.11737*, 2025.

[22] Hongwei Chen, Yishu Lei, Dan Zhang, Bo Ke, Danxiang Zhu, Xuyi Chen, Yuxiang Lu, Zhengjie Huang, Shikun Feng, Jingzhou He, et al. Matryoshkathinking: Recursive test-time scaling enables efficient reasoning. *arXiv preprint arXiv:2510.10293*, 2025.

[23] Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding. *arXiv preprint arXiv:2503.01422*, 2025.

[24] Chen Jin, Ryutaro Tanno, Tom Diethe, and Philip Teare. Corefine: Confidence-guided self-refinement for adaptive test-time compute. *arXiv preprint arXiv:2602.08948*, 2026.

[25] Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvalddson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026. URL https://arxiv.org/abs/2605.00674.

[26] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

[27] Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. Calibrating reasoning in language models with internal consistency. *Advances in Neural Information Processing Systems*, 37:114872–114901, 2024.

[28] Xizhong Yang, Haotian Zhang, Huiming Wang, and Mofei Song. Believe your model: Distribution-guided confidence calibration. *arXiv preprint arXiv:2603.03872*, 2026.

[29] Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111, 2025.

[30] Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

[31] Aman Sharma and Paras Chopra. The sequential edge: Inverse-entropy voting beats parallel self-consistency at matched compute. *arXiv preprint arXiv:2511.02309*, 2025.

[32] Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, 2023.

[33] 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. In *The Twelfth International Conference on Learning Representations*.

[34] Ehsan Aghazadeh, Ahmad Ghasemi, Hedyeh Beyhaghi, and Hossein Pishro-Nik. Cges: Confidence-guided early stopping for efficient and accurate self-consistency. *arXiv preprint arXiv:2511.02603*, 2025.

[35] Colin Hong, Xu Guo, Anand Chaanan Singh, Esha Choukse, and Dmitrii Ustiugov. Slim-sc: Thought pruning for efficient scaling with self-consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34488–34505, 2025.

[36] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[37] nostalgebraist. Interpreting GPT: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. LessWrong blog post.

[38] Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

[39] 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 q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[40] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

[41] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

A Full proofs

A.1 Proof of Proposition 1

We prove Proposition 1 in two steps. First, we show that the correctness of the final prediction is exactly characterized by the sign of the aggregated margin. Second, we derive a Gaussian approximation to the distribution of the margin.

Step 1: Exact characterization by the margin. The final prediction is defined as

$$\hat{y} = \arg\max_{y} \mathrm{Conf}(y).$$

By definition, the prediction is correct if and only if the score of the correct answer y^* exceeds the score of every incorrect answer:

$$\hat{y} = y^* \quad \Longleftrightarrow \quad \mathrm{Conf}(y^*) > \max_{y \neq y^*} \mathrm{Conf}(y).$$

Equivalently,

$$\hat{y} = y^* \quad \Longleftrightarrow \quad \mathrm{Conf}(y^*) - \max_{y \neq y^*} \mathrm{Conf}(y) > 0.$$

Using the definition of the aggregated margin,

$$\Delta = \mathrm{Conf}(y^*) - \max_{y \neq y^*} \mathrm{Conf}(y),$$

we obtain

$$\hat{y} = y^* \quad \Longleftrightarrow \quad \Delta > 0.$$

Therefore,

$$\Pr(\hat{y} = y^*) = \Pr(\Delta > 0).$$

Step 2: Margin as a sum of signed per-path contributions. Let

$$y_w = \arg\max_{y \neq y^*} \mathrm{Conf}(y)$$

denote the strongest incorrect answer under the aggregation rule. Consider the signed contribution of the i-th path:

$$Z_i = W(S_i)\left(\mathbf{1}[y_i = y^*] - \mathbf{1}[y_i = y_w]\right).$$

Then the difference between the aggregated score of the correct answer and that of the strongest incorrect answer can be written as

$$\mathrm{Conf}(y^*) - \mathrm{Conf}(y_w) = \sum_{i=1}^{M} W(S_i)\left(\mathbf{1}[y_i = y^*] - \mathbf{1}[y_i = y_w]\right) = \sum_{i=1}^{M} Z_i.$$

Since

$$\Delta = \mathrm{Conf}(y^*) - \max_{y \neq y^*} \mathrm{Conf}(y) = \mathrm{Conf}(y^*) - \mathrm{Conf}(y_w),$$

we have

$$\Delta = \sum_{i=1}^{M} Z_i.$$

Step 3: Mean and variance of the margin. By assumption,

$$\mathbb{E}[Z_i] = \mu, \qquad \mathrm{Var}(Z_i) = \sigma^2.$$

Since the Z_i are independent, the mean and variance of Δ are

$$\mathbb{E}[\Delta] = \mathbb{E}\left[\sum_{i=1}^{M} Z_i\right] = \sum_{i=1}^{M} \mathbb{E}[Z_i] = M\mu,$$

and

$$\mathrm{Var}(\Delta) = \mathrm{Var}\left(\sum_{i=1}^{M} Z_i\right) = \sum_{i=1}^{M} \mathrm{Var}(Z_i) = M\sigma^2.$$

Step 4: Gaussian approximation. Because Z_i are independent and bounded, the central limit theorem implies that for sufficiently large M,

$$\frac{\Delta - M\mu}{\sqrt{M}\sigma} \approx \mathcal{N}(0,1).$$

Equivalently,

$$\Delta \approx \mathcal{N}(M\mu, M\sigma^2).$$

Step 5: Approximate prediction accuracy. Using the exact characterization from Step 1,

$$\Pr(\hat{y} = y^*) = \Pr(\Delta > 0).$$

Under the Gaussian approximation,

$$\Pr(\Delta > 0) = \Pr\left(\frac{\Delta - M\mu}{\sqrt{M}\sigma} > \frac{-M\mu}{\sqrt{M}\sigma}\right).$$

Let

$$Z = \frac{\Delta - M\mu}{\sqrt{M}\sigma} \approx \mathcal{N}(0,1).$$

Then

$$\Pr(\Delta > 0) \approx \Pr\left(Z > -\frac{\sqrt{M}\mu}{\sigma}\right).$$

By symmetry of the standard normal distribution,

$$\Pr\left(Z > -\frac{\sqrt{M}\mu}{\sigma}\right) = \Phi\left(\frac{\sqrt{M}\mu}{\sigma}\right).$$

Therefore,

$$\Pr(\hat{y} = y^*) \approx \Phi\left(\frac{\sqrt{M}\mu}{\sigma}\right).$$

This completes the proof.

A.2 Derivation for Prefix-Length Selection

We prove that, under a linear margin growth assumption, the optimal prefix length is attained at one of the endpoints of the feasible interval.

Step 1: Objective. Recall that the effective voting strength is defined as

$$V(k) = \sqrt{M(k)}\Delta(k).$$

Under a fixed budget, the number of completed paths after prefix filtering is

$$M(k) = \frac{\alpha B}{k + \alpha(T-k)} = \frac{\alpha B}{\alpha T + (1-\alpha)k}.$$

Assume that the margin grows linearly with the prefix length:

$$\Delta(k) = \Delta_0 + \beta k.$$

Therefore,

$$V(k) = \sqrt{\frac{\alpha B}{\alpha T + (1-\alpha)k}}(\Delta_0 + \beta k).$$

Since $\sqrt{\alpha B}$ is independent of k, maximizing $V(k)$ is equivalent to maximizing

$$\tilde{V}(k) = \frac{\Delta_0 + \beta k}{\sqrt{\alpha T + (1-\alpha)k}}.$$

Step 2: Derivative. Let

$$A(k) = \Delta_0 + \beta k, \qquad C(k) = \alpha T + (1-\alpha)k.$$

Then

$$\tilde{V}(k) = A(k)C(k)^{-1/2}.$$

Taking derivative with respect to k,

$$\tilde{V}'(k) = \beta C(k)^{-1/2} - \frac{1}{2}A(k)C(k)^{-3/2}(1-\alpha).$$

Factoring out $C(k)^{-3/2} > 0$, we obtain

$$\tilde{V}'(k) = C(k)^{-3/2}\left[\beta C(k) - \frac{1-\alpha}{2}A(k)\right].$$

Therefore, the sign of $\tilde{V}'(k)$ is determined by

$$G(k) = \beta C(k) - \frac{1-\alpha}{2}A(k).$$

Substituting $A(k)$ and $C(k)$, we get

$$G(k) = \beta\left(\alpha T + (1-\alpha)k\right) - \frac{1-\alpha}{2}\left(\Delta_0 + \beta k\right).$$

Equivalently,

$$G(k) = \alpha\beta T + \frac{1-\alpha}{2}\beta k - \frac{1-\alpha}{2}\Delta_0.$$

Thus $G(k)$ is an affine function of k.

Step 3: Shape of the objective. Since $G(k)$ is affine in k, the derivative $\tilde{V}'(k)$ can change sign at most once over the interval

$$k \in [k_{\min}, k_{\max}].$$

Therefore, $\tilde{V}(k)$ is either monotone over the interval, or it decreases first and then increases. In either case, $\tilde{V}(k)$ cannot have a strict interior maximum.

Hence the maximum of $\tilde{V}(k)$, and therefore the maximum of $V(k)$, must be attained at one of the two endpoints:

$$k^* \in \{k_{\min}, k_{\max}\}.$$

B Definitions of Used Confidence Score in this Work

Let $S = (s_1, s_2, \ldots, s_T)$ be a generated reasoning path, where each $s_t \in \mathcal{V}$ is a token. Let $p_t(v) = P(s_t = v \mid s_{<t})$ denote the model's predicted probability distribution over vocabulary $\mathcal{V}$ at position t, and let $|\mathcal{V}|$ denote the vocabulary size.

Likelihood (LL). The likelihood score aggregates the log-probabilities of the generated tokens:

$$\mathrm{LL}(S) = \exp\left(\frac{1}{T}\sum_{t=1}^{T}\log p_t(s_t)\right). \tag{2}$$

Exp makes sure LL is not negative. If it negative it cannot be used for weighted voting.

Entropy. The entropy score measures the average uncertainty of the model's predictions, with lower entropy indicating higher confidence:

$$\mathrm{Entropy}(S) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{v\in\mathcal{V}} p_t(v)\log p_t(v). \tag{3}$$

A lower entropy corresponds to a higher-confidence path; in practice, the negative entropy $\exp(-\mathrm{Entropy}(S))$ is used as the confidence score.

Gini Impurity. Gini Impurity measures the concentration of the predicted distribution, analogous to the Gini coefficient:

$$\mathrm{Gini}(S) = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \sum_{v\in\mathcal{V}} p_t(v)^2\right). \tag{4}$$

A lower Gini Impurity indicates a more peaked (confident) distribution; the negative value $-\mathrm{Gini}(S)$ is used as the confidence score.

Diff-token. Diff-token measures the gap between the top-1 and top-2 predicted probabilities at each step, rewarding distributions where one token is clearly preferred:

$$\mathrm{DiffToken}(S) = \frac{1}{T}\sum_{t=1}^{T}\left(p_t^{(1)} - p_t^{(2)}\right), \tag{5}$$

where $p_t^{(1)} \geq p_t^{(2)}$ are the largest and second-largest probabilities at position t.

Conf-Lens. Conf-Lens is a new confidence score we introduce, derived from the logit lens [37]. The logit lens projects intermediate hidden states $h_t^{(\ell)}$ at layer ℓ through the final unembedding matrix to obtain an intermediate predicted probability $\tilde{p}_t^{(\ell)}(s_t)$ for the generated token s_t. We measure confidence by aggregating these intermediate probabilities over a set of later layers $\mathcal{L}$:

$$\mathrm{ConfLens}(S) = \exp\left(\frac{1}{T}\sum_{t=1}^{T}\log\left(\frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\tilde{p}_t^{(\ell)}(s_t)\right)\right), \tag{6}$$

where $\mathcal{L}$ denotes the set of later layers used. This can be interpreted as the geometric mean over token positions of the layer-averaged intermediate probability assigned to each generated token. A higher score indicates that the later intermediate layers consistently assign high probability to the generated tokens, reflecting greater internal agreement and confidence. We include Conf-Lens only to evaluate whether PreG generalizes across different types of confidence signals, rather than as a core contribution.

DeepConf (DC). DeepConf measures the concentration of predicted distributions by aggregating token-level confidence based on the peakedness of the softmax output:

$$\mathrm{DC}(S) = \frac{1}{T}\sum_{t=1}^{T}\left(-\frac{1}{k}\sum_{j=1}^{k}\log p_t(j)\right). \tag{7}$$

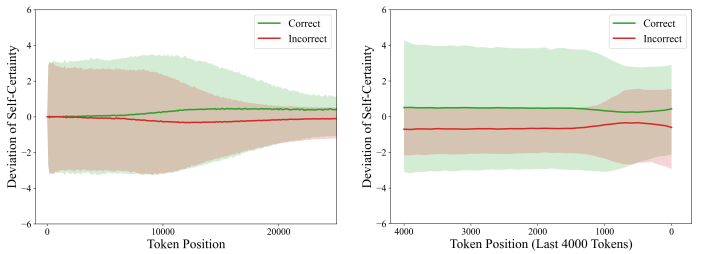
Self-Certainty (SCert). Self-Certainty is a KL-divergence-inspired distributional confidence metric defined as:

$$\mathrm{SCert}(S) = -\frac{1}{TV}\sum_{t=1}^{T}\sum_{j=1}^{V}\log\big(V\cdot p_t(j)\big), \tag{8}$$

where $V = |\mathcal{V}|$ is the vocabulary size and $p_t(j)$ denotes the predicted probability of the j-th vocabulary token at position t. This is equivalent to the KL divergence $D_{\mathrm{KL}}(p_t\|\mathcal{U})$ up to a constant, where $\mathcal{U}$ is the uniform distribution over $\mathcal{V}$.

Table 5: License information of assets

Asset	License	Link
Qwen/Qwen3-8B	Apache license 2.0	https://huggingface.co/Qwen/Qwen3-8B
nvidia/Llama-3.1-Nemotron-Nano-4B-v1.1	nvidia-open-model-license	https://huggingface.co/nvidia/Llama-3.1-Nemotron-Nano-4B-v1.1
Qwen/Qwen2.5-Math-PRM-7B	qwen	https://huggingface.co/Qwen/Qwen2.5-Math-PRM-7B
Skywork/Skywork-Reward-V2-Qwen3-8B	Apache license 2.0	https://huggingface.co/Skywork/Skywork-Reward-V2-Qwen3-8B
lchen001/AIME2025	Apache license 2.0	https://huggingface.co/datasets/lchen001/AIME2025
MathArena/brumo_2025	CC-BY-NC-SA-4.0	https://huggingface.co/datasets/MathArena/brumo_2025
MathArena/hmmt_feb_2025	CC-BY-NC-SA-4.0	https://huggingface.co/datasets/MathArena/hmmt_feb_2025
MathArena/cmimc_2025	CC-BY-NC-SA-4.0	https://huggingface.co/datasets/MathArena/cmimc_2025
Idavidrein/gpqa	CC-BY-4.0	https://huggingface.co/datasets/Idavidrein/gpqa



(a) (b)

Figure 6: Results on prefix scores.

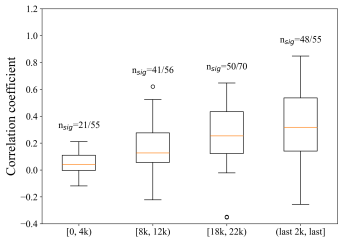


Figure 7: Correlation coefficients

C Licenses of Assets

In Table 5, we present the license information of all the assets including the data resources and the model we use.

D Additional Experiment Results

D.1 Studies on the internal signals from prefix

Similar to Self-Consistency, parallel methods which uses token-level confidence scores typically make decisions, such as evaluating path quality or aggregating paths, only after full path generation. Many early truncation methods also rely on signals from completed paths rather than prefix-based stopping within a single path. While full-path signals are more informative, they incur unnecessary cost on paths that yield incorrect answers. In this subsection, we conduct a preliminary study to explore how much prefix information can help reduce unnecessary tokens.

We first examine how confidence evolves along correct and incorrect paths. In Figure 6, we plot the Self-Certainty of Qwen3 on four different math datasets. For each question, we sample 512 responses from Qwen3-8B and compute token-level Self-Certainty, enabling a comparison between paths that lead to correct versus incorrect final answers. Since such comparisons are most meaningful within the same question, we normalize each token's score by subtracting the mean score at that token position across all samples for that question. Figure 6a shows the variation from the first token onward, while Figure 6b presents the results over the last 4,000 tokens.

The average length of Qwen3 reasoning paths on these problems is 2.1k tokens. From Figure 6a, we observe that Self-Certainty shows almost no difference at the very beginning. Starting from around 2,000 tokens, a noticeable gap in the mean emerges, which continues to widen until approximately 12,000 tokens. From Figure 6b, we further observe that the mean difference becomes larger in the final 4,000 tokens. This suggests that, although the difference is small, early tokens have already begun to diverge and start to provide informative signals that can distinguish potential correct and incorrect paths.

Compared to Figure 6, Figure 7 further illustrates how Self-Certainty can help identify correct answers. Following the use of internal signals, we aggregate token-level information by averaging over different segments, i.e., token ranges [0, 4k), [8k, 12k), [18k, 22k), and [last2k, last]. We then

Table 6: STD

Qwen3-8B		AIME25 (100k/300k/1000k)	BRUMO (100k/300k/1000k)	HMMT (100k/300k/1000k)	CMIMC (100k/300k/1000k)	GPQA-D (100k/300k/1000k)
Pass@1		67.94	69.64	45.30	46.14	49.80
Majority voting		3.30 / 2.69 / 2.68	3.67 / 2.98 / 2.33	4.88 / 3.06 / 2.31	4.03 / 3.02 / 1.52	1.77 / 1.25 / NA
PRM		2.64 / 2.81 / 2.89	3.48 / 2.78 / 2.09	3.62 / 2.96 / 1.96	3.36 / 3.11 / 1.83	1.83 / 1.35 / 1.09
ESC		2.69 / 2.78 / 2.47	3.58 / 2.61 / 2.03	3.92 / 2.94 / 2.59	3.26 / 2.35 / 1.52	1.58 / 1.29 / NA
ST-BoN		3.82 / 2.98 / 2.33	3.51 / 3.13 / 2.43	4.58 / 3.60 / 2.77	4.43 / 3.16 / 2.04	1.75 / 1.27 / 1.07
CGES		4.66 / 4.12 / NA	4.15 / 3.24 / NA	5.34 / 4.29 / NA	4.58 / 3.85 / NA	NA / NA / NA
Borda	LL	3.27 / 2.33 / 2.62	4.06 / 2.46 / 1.74	4.62 / 2.87 / 2.59	3.99 / 3.80 / 2.15	1.88 / 1.56 / 1.02
	LL-PreG	3.41 / 3.19 / 2.80	3.32 / 2.07 / 1.25	4.74 / 3.50 / 3.16	4.02 / 3.83 / 2.57	1.45 / 1.23 / 0.48
	DC	3.78 / 2.89 / 2.72	3.43 / 2.65 / 2.13	4.69 / 3.21 / 1.89	4.08 / 3.85 / 3.17	1.59 / 1.36 / 0.90
	DC-PreG	3.23 / 2.87 / 1.70	2.88 / 1.93 / 0.95	4.58 / 3.04 / 2.07	4.36 / 3.26 / 2.72	1.41 / 1.18 / 0.34
	SCert	3.45 / 3.07 / 2.66	3.83 / 2.59 / 1.73	5.19 / 3.44 / 2.08	3.73 / 3.65 / 2.87	1.75 / 1.25 / 0.86
	SCert-PreG	3.01 / 3.02 / 1.97	3.29 / 1.68 / 0.47	4.85 / 3.42 / 2.05	3.95 / 3.20 / 3.04	1.57 / 1.21 / 0.37
W-10%	LL	4.93 / 5.20 / 3.54	4.65 / 4.65 / 3.32	4.69 / 4.83 / 3.78	4.91 / 4.72 / 3.76	2.56 / 2.20 / 1.41
	LL-PreG	5.79 / 3.71 / 2.72	4.62 / 3.24 / 2.24	5.80 / 4.47 / 3.71	4.78 / 4.53 / 3.62	2.34 / 1.62 / 1.42
	DC	4.66 / 4.58 / 2.72	4.41 / 4.50 / 2.95	4.98 / 5.35 / 3.57	4.71 / 4.46 / 3.50	2.34 / 1.93 / 1.30
	DC-PreG	4.33 / 3.41 / 1.84	4.05 / 2.53 / 1.62	5.28 / 5.09 / 3.09	4.83 / 4.37 / 3.39	2.29 / 1.80 / 1.24
	SCert	4.86 / 4.83 / 2.83	4.50 / 4.35 / 2.70	5.15 / 4.96 / 3.62	4.73 / 4.58 / 4.11	2.28 / 1.84 / 1.30
	SCert-PreG	4.57 / 3.77 / 2.21	3.72 / 2.68 / 1.69	5.30 / 4.56 / 2.87	5.02 / 4.55 / 3.58	2.49 / 1.80 / 1.20
Llama-Nemotron-4B		AIME25 (100k/300k/1000k)	BRUMO (100k/300k/1000k)	HMMT (100k/300k/1000k)	CMIMC (100k/300k/1000k)	GPQA-D (100k/300k/1000k)
Pass@1		56.41	63.76	41.11	37.88	43.03
Majority voting		4.62 / 3.46 / 2.51	3.56 / 3.36 / 2.53	4.07 / 3.28 / 2.77	4.04 / 2.83 / 2.09	2.37 / 1.54 / NA
PRM		2.64 / 2.81 / 2.89	3.48 / 2.78 / 2.09	3.62 / 2.96 / 1.96	3.36 / 3.11 / 1.83	1.83 / 1.35 / 1.09
ESC		2.69 / 2.78 / 2.47	3.58 / 2.61 / 2.03	3.92 / 2.94 / 2.59	3.26 / 2.35 / 1.52	1.58 / 1.29 / NA
ST-BoN		3.82 / 2.98 / 2.33	3.51 / 3.13 / 2.43	4.58 / 3.60 / 2.77	4.43 / 3.16 / 2.04	1.75 / 1.27 / 1.07
CGES		4.66 / 4.12 / NA	4.15 / 3.24 / NA	5.34 / 4.29 / NA	4.58 / 3.85 / NA	NA / NA / NA
Borda	LL	4.08 / 3.43 / 2.63	3.58 / 3.69 / 3.03	4.67 / 3.56 / 3.04	3.62 / 2.87 / 1.76	2.56 / 1.77 / 1.21
	LL-PreG	4.85 / 3.63 / 2.60	3.45 / 3.36 / 2.83	4.59 / 2.79 / 2.38	4.05 / 3.19 / 1.81	2.53 / 1.55 / 1.09
	DC	3.75 / 3.69 / 2.57	3.74 / 3.68 / 2.87	4.80 / 3.46 / 3.09	3.42 / 2.87 / 1.70	2.51 / 1.88 / 1.25
	DC-PreG	4.36 / 3.87 / 2.40	3.30 / 3.29 / 3.05	4.25 / 3.87 / 2.35	4.48 / 2.93 / 1.51	2.44 / 1.43 / 1.08
	SCert	3.86 / 3.64 / 2.56	3.30 / 3.52 / 3.02	4.76 / 3.54 / 3.32	3.38 / 2.94 / 1.90	2.42 / 1.75 / 1.31
	SCert-PreG	4.17 / 3.84 / 2.03	3.34 / 3.22 / 2.80	4.38 / 3.23 / 2.51	4.01 / 3.25 / 1.74	2.17 / 1.44 / 0.99
W-10%	LL	5.33 / 5.29 / 4.22	4.23 / 4.40 / 3.06	5.11 / 4.89 / 3.66	4.15 / 4.34 / 3.94	2.63 / 2.76 / 2.19
	LL-PreG	5.66 / 4.04 / 3.20	4.05 / 3.54 / 3.18	4.82 / 4.01 / 2.63	4.91 / 3.27 / 3.08	2.43 / 2.09 / 1.53
	DC	5.11 / 4.98 / 4.02	3.52 / 3.66 / 3.01	4.87 / 4.92 / 3.48	3.94 / 4.49 / 4.01	2.72 / 2.72 / 2.09
	DC-PreG	5.77 / 4.92 / 3.21	3.70 / 3.75 / 3.13	4.61 / 4.52 / 2.94	5.09 / 4.09 / 2.97	2.38 / 2.38 / 1.80
	SCert	4.77 / 4.83 / 4.30	3.73 / 3.92 / 2.96	4.86 / 4.98 / 3.75	3.97 / 4.98 / 3.92	2.65 / 2.39 / 1.99
	SCert-PreG	5.46 / 4.43 / 2.96	4.04 / 3.72 / 3.51	5.01 / 4.28 / 2.91	4.64 / 4.13 / 2.81	2.29 / 2.02 / 1.71

compute, for each question, the correlation between sample-level scores and labels, yielding the relationship between samples and confidence scores.

In Figure 7, after removing questions with highly imbalanced labels, we present box plots of the correlation coefficients across all questions, along with the proportion of significant correlations (e.g., n_{sig}). We observe that later segments exhibit stronger correlations with the labels; however, even within the first 4k tokens, a clear correlation already emerges. Approximately 75% of questions show a positive correlation between confidence scores and labels in this early segment, with 38% ($n_{\text{sig}} = 21/55$) being statistically significant. These findings suggest that, even from prefixes, it is possible to select path candidates more likely to yield correct final answers, thereby improving overall performance.

D.2 STD of Main Results

The STD of main results are reported in Table 6.

D.3 Comparison with Other Aggregation Methods and Confidence Scores

The results of other voting methods on Qwen are reported in Table 7. The results of other confidence scores methods on Qwen are reported in Table 8.

Table 7: Other aggregation methods

Qwen		Math
Weighted voting	LL	64.61 / 67.29 / 68.27
	LL-PreG	62.63 / 67.75 / 69.59
	DC	65.71 / 67.90 / 68.53
	DC-PreG	64.84 / 68.59 / 69.96
	SCert	65.97 / 68.01 / 68.76
	SCert-PreG	65.42 / 68.74 / 70.38
Majority voting	Majority voting	64.65 / 67.18 / 68.24
	LL-PreG	62.51 / 67.66 / 69.54
	DC-PreG	62.72 / 68.08 / 69.71
	SCert-PreG	62.83 / 67.93 / 69.86

Table 8: Other confidences scores

		Math
Borda	Diff-token	63.05 / 67.05 / 68.18
	Diff-token-PreG	64.22 / 68.29 / 70.00
	Gini	63.15 / 67.06 / 68.21
	Gini-PreG	64.31 / 68.45 / 70.09
	Entropy	63.14 / 67.19 / 68.34
	Entropy-PreG	64.48 / 68.65 / 70.21
	Conf-Lens	64.43 / 67.76 / 68.59
	Conf-Lens-PreG	64.85 / 69.19 / 70.85
W-10%	Diff-token	57.66 / 57.01 / 62.39
	Diff-token-PreG	58.80 / 62.45 / 67.52
	Gini	57.76 / 56.73 / 62.37
	Gini-PreG	59.03 / 62.64 / 67.49
	Entropy	58.10 / 57.07 / 62.53
	Entropy-PreG	59.10 / 62.90 / 67.84
	Conf-Lens	60.26 / 60.05 / 64.98
	Conf-Lens-PreG	60.63 / 64.21 / 69.40

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that we propose a framework for parallel TTS.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The full proofs are presented in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full implementation details in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data are publicly available at `https://github.com/renjie3/PreG`, along with instructions for reproducing the experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide full implementation details in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide STD in Appendix D.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details of compute resources are provided in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release any models or datasets that pose a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models used in this paper are properly cited. The licenses are presented in Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The generated data, code, and documentation are publicly available at `https://github.com/renjie3/PreG`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for writing assistance and do not form part of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.