Reasoning under Uncertainty: Efficient LLM Inference via Unsupervised Confidence Dilution and Convergent Adaptive Sampling

Zhenning Shi[1,2,3], Yijia Zhu[5], Yi Xie[6], Junhan Shi[1], Guorui Xie[2],
Haotian Zhang[4], Yong Jiang[1,2*], Congcong Miao[3], Qing Li[2*]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China [3]Tencent, Shenzhen, China
[4]Peking University, Beijing, China [5]Xidian University, Xian, China
[6]Hunan Agricultural University, Changsha, China
{shizn23, shijh23}@mails.tsinghua.edu.cn, jiangy@sz.tsinghua.edu.cn, liq@pcl.ac.cn

Abstract

Large language models (LLMs) excel at complex reasoning tasks but often suffer from overconfidence and computational inefficiency due to fixed computation budgets and miscalibrated confidence estimates. We present a novel framework for computationally efficient, trustworthy reasoning under uncertainty, introducing two complementary techniques: Diversity-Aware Self-Signal Dilution (DASD) and Convergent Adaptive Weighted Sampling (CAWS). DASD operates in an unsupervised manner to dilute overconfident, semantically redundant reasoning paths, thereby producing better-calibrated internal confidence estimates. CAWS dynamically allocates computational resources at inference time by aggregating these signals and terminating computation once answer dominance and stability are achieved. Comprehensive experiments across three reasoning datasets demonstrate that our approach maintains accuracy levels while achieving over 70% reduction in inference cost, surpassing competitive baselines. Our framework provides a scalable, unsupervised solution for reliable and efficient LLM reasoning.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities across a range of complex reasoning tasks, including arithmetic problem-solving, commonsense inference, and multi-step question answering (Ahn et al., 2024; Li et al., 2025a). Beyond reasoning, LLMs have also been increasingly applied in other domains such as code and UI generation (Xiao et al., 2024; Wan et al., 2024; Xiao et al., 2025a,b; Tang et al., 2025). Despite these advancements, generating *accurate and trustworthy* answers during inference remains challenging. A central issue is that LLMs often produce outputs with uncertain reliability (Steyvers et al., 2025; Herrera-Poyatos et al., 2025; Wang et al., 2025a), while offering little signal to determine when an answer should be trusted, especially for inputs of varying difficulty and ambiguity.

To mitigate this, a common practice is to scale up inference-time computation via sampling-based strategies such as Best-of-N or Self-Consistency sampling. These methods aggregate multiple reasoning paths to increase the chance of correctness. However, regardless of complexity, they assign a fixed number of samples to each input. This leads to inefficiency: simple questions are oversampled, while hard ones are often underexplored, resulting in wasted computation and inconsistent outcomes (Chiang and Lee, 2024; Chen et al., 2025).

An alternative line of work leverages model-generated confidence scores to guide inference adaptively (Huang et al., 2025b; Kang et al., 2025). By terminating sampling early when the model exhibits high confidence, or re-ranking candidate answers using confidence signals, these methods aim to save computation on easy instances and allocate more effort to difficult ones. Yet, this line of work faces a fundamental challenge: *raw confidence scores from LLMs are frequently miscalibrated*, often reflecting surface fluency rather than semantic correctness (Zhang et al., 2024; Wang et al., 2025b). Moreover, the model often produces multiple reasoning paths that differ lexically but are logically redundant, which can make the answer appear more certain than it is.

To address these limitations, we propose a self-supervised framework that transforms the LLM's imperfect uncertainty signals into well-calibrated reasoning behavior. Our method introduces two complementary components operating at training and inference time, respectively, targeting both confidence quality and computational efficiency.

We propose *Diversity-Aware Self-Signal Dilution (DASD)*, a training-time confidence calibration algorithm that mitigates overconfidence by explicitly modeling semantic redundancy in the model's

[*] Yong Jiang and Qing Li are the corresponding authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32204–32218
November 4-9, 2025 ©2025 Association for Computational Linguistics

outputs. Given a set of sampled reasoning paths for a given input, DASD clusters answers based on their final prediction and estimates diversity within each cluster using lightweight LLM-based semantic comparisons. Clusters containing many redundant variants of the same flawed reasoning path are penalized via a soft dilution mechanism, yielding calibrated pseudo-confidence labels that better reflect the true informativeness of each answer. These labels are then used to fine-tune the model's self-evaluation behavior, improving its ability to generate well-calibrated internal confidence scores.

We design *Convergent Adaptive Weighted Sampling (CAWS)*, a test-time inference algorithm designed to efficiently and robustly aggregate uncertain model outputs. CAWS incrementally samples candidate answers and uses the model's self-estimated confidence to assign soft voting weights via a sigmoid mapping. Crucially, instead of relying on a fixed number of samples or a hard confidence threshold, CAWS monitors the running distribution of weighted answers and applies convergence-based stopping criteria: inference halts only when a leading answer has remained dominant and stable across multiple steps. This adaptive mechanism allows the model to save computation on easy instances while allocating more effort to hard or ambiguous ones, resulting in better accuracy-compute trade-off and improved robustness to residual miscalibration.

Together, these two modules form a unified framework for scalable, trustworthy LLM reasoning under uncertainty, where DASD improves the quality of confidence signals and CAWS leverages them effectively during inference. Through extensive evaluations across diverse reasoning benchmarks, including GSM8K (Cobbe et al., 2021), ARC-Challenge (Clark et al., 2018), and CommonsenseQA (Talmor et al., 2018), and across three representative models (LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2024), and DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025)), we demonstrate that our proposed method consistently outperforms fixed-budget and confidence-based baselines across all benchmarks, maintaining or improving accuracy while reducing inference costs by over 70%. Our analysis further highlights the effectiveness of confidence dilution, directly contributing to the improved reliability of adaptive inference.

Overall, our work provides a significant step toward bridging the efficiency-reliability gap in LLM inference, offering a scalable, unsupervised solution that advances practical deployment and reliable decision-making in reasoning scenarios.

Our contributions are as follows:

- We propose a self-supervised confidence calibration approach that explicitly accounts for semantic redundancy in model generations, producing more informative and trustworthy internal confidence signals.

- We develop *Diversity-Aware Self-Signal Dilution*, an unsupervised method that explicitly models intra-cluster semantic diversity to produce calibrated confidence without relying on annotations or external reward models.

- We introduce *Convergent Adaptive Weighted Sampling*, a novel stopping mechanism that dynamically halts inference based on stability and dominance, enabling robust compute allocation under imperfect confidence estimates.

- We demonstrate that our unified framework achieves substantial efficiency gains while maintaining accuracy across diverse reasoning tasks, establishing a new paradigm for efficient and reliable LLM inference[1].

2 Related Work

Test-time compute scaling. A common approach to improve LLM reasoning performance is to generate multiple outputs and aggregate them via fixed-budget strategies (Snell et al., 2024; Ji et al., 2025). Best-of-N decoding selects the output with the highest likelihood among N samples (Sun et al., 2024), while Self-Consistency (SC) sampling aggregates final answers from multiple reasoning chains to improve robustness (Chen et al., 2023b). Despite their empirical success, these methods statically allocate computational resources, regardless of input difficulty. This leads to inefficiency: trivial questions are oversampled, while ambiguous or complex ones may be underexplored (Tan et al., 2025; Liu et al., 2025a,b). Our work instead focuses on adaptive inference, adjusting the number of samples dynamically based on real-time confidence convergence.

[1] `https://github.com/sznnzs/Trustworthy-LLM-Reasoning`

Reward signal acquisition. Another line of work uses auxiliary reward models trained to assess response quality or human preference, often applied during reinforcement learning (Wu, 2025) or re-ranking of the decoding time (Huang et al., 2024a). Although effective in some settings, these models require labeled data, are expensive to train, and introduce substantial memory and latency overhead during inference (Zhong et al., 2024; Sheng et al., 2024). Moreover, recent studies have shown that reward models can be poorly calibrated and prone to over-rewarding fluent but incorrect completions (Huang et al., 2024b; Leng et al., 2024; Rita et al., 2024), thus limiting their reliability as a source of truth. In contrast, our approach avoids any form of external supervision and focuses on improving the utility of the native LLM uncertainty signals through unsupervised calibration.

Confidence-based adaptive inference. Several recent methods attempt to improve inference efficiency by leveraging scalar reward signals to guide decoding (Taubenfeld et al., 2025). For instance, early stopping strategies stop sampling once a candidate exceeds a fixed score threshold (Agrawal et al., 2024; Li et al., 2024), while other approaches use confidence-weighted voting to combine reasoning chains (Chen et al., 2023a; Razghandi et al., 2025). However, overreliance on uncalibrated confidence scores can lead to *reward hacking*—where models produce fluent yet incorrect outputs that game the scoring mechanism (Moskovitz et al., 2023; Miao et al., 2024b; Huang et al., 2025a). These issues make reward-based adaptive inference brittle and difficult to generalize. In contrast, our work unsupervisedly calibrates the internal confidence of the model and utilizes it robustly to ensure stable and trustworthy reasoning without external heuristics or task-specific scores.

3 Methodology

3.1 Problem Setup and Framework Overview

We consider open-ended question answering with a large language model (LLM), where given a natural language input x, the model generates a reasoning trace $y \sim p_\theta(\cdot \mid x)$, representing a full sequence of intermediate steps or explanations. A final answer $z = \texttt{parse}(y)$ (e.g., a choice or number) is extracted from the output.

At test time, multiple samples are typically drawn to improve answer quality and robustness. Our goal is to identify the most reliable answer while reducing the number of samples needed. To this end, we propose a unified two-stage framework. Figure 1 illustrates the overall pipeline.

- **Training stage:** We perform *Diversity-Aware Self-Signal Dilution*, a confidence calibration module that estimates intra-cluster semantic redundancy and softly downweights over-represented reasoning patterns. The resulting pseudo-labels are used to fine-tune the model's confidence prediction behavior in a fully self-supervised manner.

- **Inference stage:** At test time, we apply *Convergent Adaptive Weighted Sampling*, an algorithm that incrementally samples outputs, assigns confidence-based vote weights, and halts when answer distribution converges. This mechanism dynamically adjusts compute allocation based on real-time signal aggregation.

3.2 Self-Supervised Confidence Calibration

3.2.1 The Challenge of Overconfidence

While large language models demonstrate impressive reasoning capabilities, their self-assessed confidence scores often poorly align with actual answer correctness. LLMs typically exhibit *overconfidence*, assigning high confidence to incorrect answers, particularly when reasoning traces appear fluent or superficially coherent (Sun et al., 2025; Bodhwani et al., 2025). This miscalibration undermines the reliability of internal signals for adaptive inference decisions. The issue is further compounded by repetition in autoregressive decoding. Models frequently generate multiple variations of a reasoning path that differ in wording but follow nearly identical logic, converging on the same (possibly incorrect) final answer (Wang et al., 2022). These outputs are mistakenly treated as independent evidence, despite offering little meaningful diversity (Ginart et al., 2025). As a result, naive aggregation strategies such as majority voting or score averaging become prone to systematic overconfidence when faced with clusters of logically redundant reasoning chains (Chiang and Lee, 2024).

3.2.2 Redundancy-Aware Score Dilution

To mitigate overconfidence induced by semantic redundancy, we introduce a soft calibration mechanism that penalizes clusters of reasoning paths lacking semantic diversity. Our key intuition is

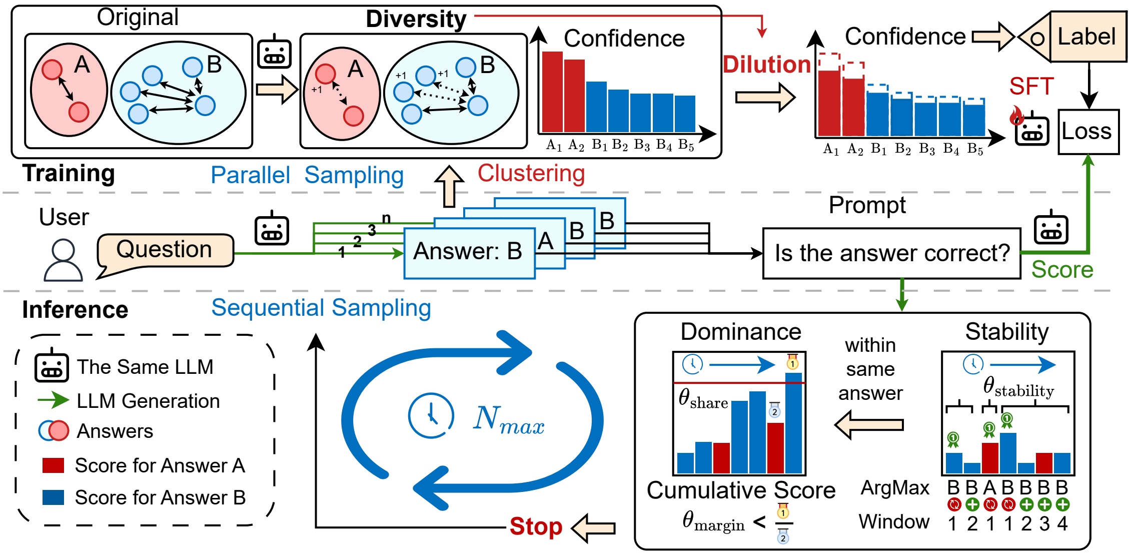


Figure 1: Overview of our two-stage framework. The training-time module, *Diversity-Aware Self-Signal Dilution (DASD)*, calibrates model confidence via semantic clustering and soft dilution. The inference-time module, *Convergent Adaptive Weighted Sampling (CAWS)*, leverages these calibrated scores to adaptively allocate compute through weighted sampling and convergence-based stopping.

that *a high number of similar answers should not be interpreted as strong evidence*, especially when those answers represent minor variations of the same flawed reasoning trace.

We define a score adjustment scheme that operates on batches of sampled outputs for a given input x. Let $\mathcal{Y} = \{y_1, \dots, y_N\}$ denote the set of reasoning paths sampled from the LLM. Each y_i is parsed into a final prediction $z_i = \texttt{parse}(y_i)$, and following (Huang et al., 2025b), its correctness score is obtained using a prompt-based verifier:

$$s_i = p_\theta(\texttt{Yes} \mid x, y_i, I) \tag{1}$$

where I is a fixed instruction that prompts the model to assess whether the answer is correct. The prompt design is detailed in Appendix B.

Step 1: Output Clustering. We group reasoning paths based on their parsed final answers. Let $\{z^{(1)}, \dots, z^{(K)}\}$ denote the set of distinct answers among $\{z_i\}$. Each cluster is defined as:

$$\mathcal{C}_k = \left\{ y_i \in \mathcal{Y} \mid z_i = z^{(k)} \right\}, \quad k = 1, \dots, K \tag{2}$$

Step 2: Intra-Cluster Diversity Estimation. To estimate semantic redundancy within each cluster, we select the highest-confidence path as representative:

$$y_k^{\text{rep}} = \arg\max_{y_i \in \mathcal{C}_k} s_i \tag{3}$$

For each other member $y_j \in \mathcal{C}_k \setminus \{y_k^{\text{rep}}\}$, we query the model with a pairwise prompt to estimate semantic similarity between y_j and the representative y_k^{rep}. We then compute the estimated number of distinct members as:

$$D_k = 1 + \sum_{y_j \in \mathcal{C}_k \setminus \{y_k^{\text{rep}}\}} \mathbb{I}\left[\texttt{sim}(y_j, y_k^{\text{rep}}) < \tau\right] \tag{4}$$

where τ is a predefined similarity threshold. The prompt used to elicit similarity scores is provided in Appendix B.

Step 3: Soft Dilution and Score Adjustment. To downweight low-diversity clusters, we define a dilution factor:

$$\alpha_k = \left(\frac{D_k}{|\mathcal{C}_k|}\right)^p, \quad p \in (0, 1) \tag{5}$$

We then compute the adjusted cluster score:

$$S_k^{\text{adj}} = \alpha_k \cdot \sum_{y_i \in \mathcal{C}_k} s_i \tag{6}$$

To produce a normalized confidence distribution over final answers, we compute for each sample:

$$\hat{R}(y_i) = \frac{S_k^{\text{adj}}}{\sum_{j=1}^{M} S_j^{\text{adj}}}, \quad \text{for } y_i \in \mathcal{C}_k \tag{7}$$

The resulting pseudo-scores $\hat{R}(y_i)$ reflect not only model-predicted correctness, but also the diversity of supporting reasoning. These signals serve as training targets for unsupervised calibration.

3.2.3 Confidence-Aware Fine-Tuning

The pseudo-confidence scores $\hat{R}(y_i)$, derived through diversity-aware dilution, serve as soft, self-supervised targets for calibrating the model's internal confidence estimates. We optimize a calibration loss $\mathcal{L}_{\text{calib}}$ to encourage alignment between the predicted scores and these targets.

For samples where the confidence target exceeds a threshold η, we additionally encourage output fidelity using a generation loss: $\mathcal{L}_{\text{gen}} = -\log p_\theta(y_i \mid x)$. The final objective combines both terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{calib}} + \lambda \cdot \mathbb{I}[\hat{R}(y_i) > \eta] \cdot \mathcal{L}_{\text{gen}} \quad (8)$$

Here, λ controls the influence of generation supervision relative to calibration. This procedure improves the model's ability to produce well-calibrated internal confidence scores without requiring any human annotations.

3.3 Convergent Adaptive Inference

3.3.1 Robustness with Imperfect Confidence

Even with improved calibration, confidence scores from large language models can remain noisy and imperfect. Consequently, inference strategies that rely on local signals, such as selecting the answer with the highest confidence or stopping early based on fixed thresholds, are often brittle and sensitive to outliers.

To address this, we design a robust inference-time algorithm that aggregates evidence across multiple generations. Our method incrementally samples candidate answers, assigns them soft voting weights derived from confidence, and monitors the evolving distribution over final predictions. Inference halts only when sufficient *convergence* is observed: one answer not only dominates the weighted vote but also remains stable across multiple sampling steps. This strategy defers commitment until the model has seen enough consistent evidence, improving robustness against noisy signals. It also enables adaptive compute allocation that naturally matches input difficulty.

3.3.2 Confidence-Weighted Voting

At each inference step, the model samples a candidate reasoning path $y^{(t)} \sim p_\theta(\cdot \mid x)$, which is parsed into a final answer $z^{(t)} = \texttt{parse}(y^{(t)})$. To aggregate outputs meaningfully, we assign each candidate a soft voting weight based on its predicted confidence score $s^{(t)} \in [0, 1]$.

To ensure robustness to raw score noise and reduce the impact of outliers, we apply a sigmoid transformation to the confidence score:

$$w^{(t)} = \sigma(s^{(t)}) = \frac{1}{1 + e^{s^{(t)}}} \quad (9)$$

This mapping amplifies differences near the decision boundary while saturating extreme values.

The cumulative score for each unique final answer z is computed as the sum of weights assigned to its supporting samples:

$$\texttt{Score}_t(z) = \sum_{i=1}^{t} \mathbb{I}[z^{(i)} = z] \cdot w^{(i)} \quad (10)$$

This formulation naturally accommodates noisy confidence estimates: rather than relying on a single high-scoring output, answers must accrue consistent support over time. The soft weighting avoids overcommitment to outlier samples and reflects the model's evolving belief distribution during sampling.

3.3.3 Convergence-Guided Stopping

To determine when to terminate inference, we monitor the evolution of the aggregated answer distribution and apply a convergence-guided stopping criterion. The intuition is to halt sampling only when one answer has both accumulated dominant support and remained stable over time.

Let $z_t^* = \arg\max_z \texttt{Score}_t(z)$ denote the current leading answer at step t. We define two convergence conditions:

- **Stability.** The same answer z_t^* must remain the top candidate for the last M steps.
- **Dominance.** The leading answer must either (a) account for a sufficient proportion of total votes or (b) surpass the second-best answer by a large margin:

$$\frac{\texttt{Score}_t(z_t^*)}{\sum_z \texttt{Score}_t(z)} \geq \theta_{\text{share}} \quad (11)$$

$$\text{or} \quad \frac{\texttt{Score}_t(z_t^*)}{\max_{z' \neq z_t^*} \texttt{Score}_t(z')} \geq \theta_{\text{margin}} \quad (12)$$

Inference halts as soon as both conditions are met. Otherwise, sampling continues until a maximum budget $N_{\max}$ is reached, at which point the most supported answer is returned:

$$z^* = \arg\max_z \texttt{Score}_{N_{\max}}(z) \quad (13)$$

This adaptive criterion balances efficiency and reliability. It allows early exit on confident inputs while allocating more compute to ambiguous or inconsistent cases. By grounding stopping decisions in aggregate signal convergence, CAWS avoids brittle reliance on raw scores or arbitrary thresholds.

4 Experiments

4.1 Experimental Setup

We evaluate our method using three open-source instruction-tuned language models: LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Yang et al., 2024), and DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025). Experiments are conducted on three representative reasoning benchmarks: GSM8K (Cobbe et al., 2021) for arithmetic reasoning, ARC-Challenge (Clark et al., 2018) for science question answering, and CommonsenseQA (Talmor et al., 2018) for commonsense inference. Prompt formats for both confidence estimation and semantic similarity judgment are provided in Appendix B. Full hyperparameter configurations are listed in Appendix 4.3.

4.2 Baselines Methods

We compare our approach against a set of widely adopted baselines that represent the dominant strategies for inference-time reasoning with LLMs:

- Best-of-N (Irvine et al., 2023; Song et al., 2024): Generates N independent reasoning paths and selects the one with the highest model score as the final answer.

- Self-Consistency (SC) (Wang et al., 2022; Chen et al., 2024): Samples multiple reasoning traces and determines the final answer via majority voting over their outputs.

- Weighted Self-Consistency (WSC) (Wu et al., 2024; Zeng et al., 2025): Extends SC by assigning each answer a weight based on its confidence, yielding a weighted aggregation.

- Early Stopping (Miao et al., 2024a; Qiao et al., 2025): Sequentially samples outputs and halts once a generated response exceeds a predefined confidence threshold.

- Adaptive Self-Consistency (ASC) (Aggarwal et al., 2023): Combines answer aggregation with confidence-based stopping, triggered by the score gap between top candidates.

Hyperparameter	Value
DASD soft dilution exponent p	0.1
Similarity threshold τ	9
Soft label threshold η	0.75
Loss weighting coefficient λ	0.1
Maximum samples N_{max}	64
Minimum samples N_{min}	5
Convergence window M	10
Confidence margin threshold θ_{margin}	3.0
Confidence share threshold θ_{share}	0.6
Training batch size	64
Learning rate	5×10^{-5}
Optimizer	AdamW
Training epochs	1

Table 1: Hyperparameters for DASD and CAWS.

4.3 Hyperparameter Settings

Table 1 lists the default hyperparameters used in our framework. These values were selected to reflect broadly applicable configurations that perform robustly across diverse datasets and models, without task-specific tuning. For all adaptive baselines that depend on confidence-based stopping criteria (e.g., Early Stopping, ASC), we adopt a unified threshold value of 0.9 across all datasets and models to ensure consistency and fair comparison with our method. We set the test split ratio to 0.1. DASD is performed for a single epoch, with a batch size of 64, a learning rate of 5×10^{-5}, and the AdamW optimizer. The loss weighting coefficient λ is set to 0.1, and the soft label threshold η is set to 0.75 (see Eq. 8). All methods are evaluated with a temperature setting of 0.7. While we prioritize generality and comparability in hyperparameter selection, further tuning on a per-task basis may improve accuracy or computational efficiency.

4.4 Accuracy and Efficiency Comparison

Table 2 shows that our framework achieves the best overall balance between accuracy and efficiency across all models and datasets. Compared to the Vanilla setting, incorporating DASD consistently improves accuracy across tasks, confirming the benefit of confidence dilution during training. Our inference-time module CAWS further enhances sample efficiency while preserving accuracy. For example, on GSM8K, CAWS achieves 92.6% accuracy with LLaMA and 94.6% with Qwen, matching SC but using less than 25% of the samples. Similar trends hold on ARC-Challenge and CommonsenseQA, where CAWS reduces compute by over 70% on average without hurting performance. This suggests that CAWS can reliably halt sampling ear-

Model	Method	ARC-Challenge				CommonSenseQA				GSM8K			
		Vanilla		DASD		Vanilla		DASD		Vanilla		DASD	
		ACC	Samples	ACC	Samples	ACC	Samples	ACC	Samples	ACC	Samples	ACC	Samples
DeepSeek	Early Stopping	67.5 ↑0.3	43.5 ↓20.5	67.8 ↑0.2	43.7 ↓20.3	53.0 ↓0.3	50.3 ↓13.7	53.0 ↑0.3	50.5 ↓13.5	89.2 ↑1.0	64.0	88.4 ↓0.2	64.0
	SC	67.2	64.0	67.6	64.0	53.3	64.0	52.7	64.0	88.2	64.0	88.6	64.0
	WSC	67.5 ↑0.3	64.0	67.8 ↑0.2	64.0	53.0 ↓0.3	64.0	53.0 ↑0.3	64.0	89.2 ↑1.0	64.0	88.4 ↓0.2	64.0
	Best-of-N	58.3 ↓8.9	64.0	56.9 ↓10.7	64.0	45.2 ↓8.1	64.0	40.9 ↓11.8	64.0	75.6 ↓12.6	64.0	68.9 ↓19.7	64.0
	ASC	65.7 ↓1.5	17.3 ↓46.7	66.7 ↓0.9	18.0 ↓46.0	52.4 ↓0.9	26.2 ↓37.8	51.6 ↓1.1	27.9 ↓36.1	87.4 ↓0.8	21.6 ↓42.4	87.3 ↓1.3	21.9 ↓42.1
	CAWS	67.2 ↑0.0	19.2 ↓44.8	67.8 ↑0.2	18.8 ↓45.2	52.4 ↓0.9	26.4 ↓37.6	52.7 ↑0.0	26.1 ↓37.9	87.9 ↓0.3	19.3 ↓44.7	88.3 ↓0.3	18.8 ↓45.2
Llama	Early Stopping	84.8 ↓3.0	2.6 ↓61.4	84.6 ↓3.3	2.0 ↓62.0	75.5 ↓3.1	2.6 ↓61.4	73.7 ↓5.6	2.0 ↓62.0	85.1 ↓6.9	12.6 ↓51.4	87.5 ↓4.9	17.4 ↓46.6
	SC	87.8	64.0	87.9	64.0	78.6	64.0	79.3	64.0	92.0	64.0	92.4	64.0
	WSC	88.3 ↑0.5	64.0	88.6 ↑0.7	64.0	78.7 ↑0.1	64.0	79.4 ↑0.1	64.0	92.2 ↑0.2	64.0	92.5 ↑0.1	64.0
	Best-of-N	86.3 ↓1.5	64.0	86.3 ↓1.6	64.0	78.7 ↑0.1	64.0	77.7 ↓1.6	64.0	85.0 ↓7.0	64.0	86.6 ↓5.8	64.0
	ASC	86.7 ↓1.1	10.7 ↓53.3	87.5 ↓0.4	11.6 ↓52.4	76.9 ↓1.7	16.2 ↓47.8	78.8 ↓0.5	15.9 ↓48.1	91.2 ↓0.8	19.5 ↓44.5	91.4 ↓1.0	17.0 ↓47.0
	CAWS	87.6 ↓0.2	15.1 ↓48.9	88.0 ↑0.1	14.6 ↓49.4	78.7 ↑0.1	17.5 ↓46.5	79.1 ↓0.2	17.1 ↓46.9	92.0 ↑0.0	16.6 ↓47.4	92.6 ↑0.2	15.9 ↓48.1
Qwen	Early Stopping	90.3 ↓1.0	41.1 ↓22.9	89.0 ↓2.6	1.6 ↓62.4	83.2 ↓0.3	60.6 ↓3.4	82.7 ↓1.0	7.5 ↓56.5	94.2 ↑0.3	64.0	94.9 ↑0.3	64.0
	SC	91.3	64.0	91.6	64.0	83.5	64.0	83.7	64.0	93.9	64.0	94.6	64.0
	WSC	90.7 ↓0.6	64.0	91.1 ↓0.5	64.0	83.2 ↓0.3	64.0	83.8 ↑0.1	64.0	94.2 ↑0.3	64.0	94.9 ↑0.3	64.0
	Best-of-N	89.6 ↓1.7	64.0	90.4 ↓1.2	64.0	81.9 ↓1.6	64.0	83.2 ↓0.5	64.0	89.9 ↓4.0	64.0	90.8 ↓3.8	64.0
	ASC	90.7 ↓0.6	8.7 ↓55.3	91.1 ↓0.5	9.5 ↓54.5	82.3 ↓1.2	11.8 ↓52.2	82.8 ↓0.9	11.5 ↓52.5	93.3 ↓0.6	11.7 ↓52.3	94.2 ↓0.4	11.4 ↓52.6
	CAWS	91.0 ↓0.3	13.6 ↓50.4	91.4 ↓0.2	13.0 ↓51.0	82.9 ↓0.6	14.4 ↓49.6	83.1 ↓0.6	14.0 ↓50.0	93.9 ↑0.0	14.4 ↓49.6	94.6 ↑0.0	13.9 ↓50.1

Table 2: Comparison of CAWS and baselines under calibrated (DASD) and non-calibrated settings. We report accuracy and average sample count. For each model-dataset block, accuracy values are compared against the SC baseline, with differences noted in green (improvement) or red (decline).

Method	ARC-Challenge		CommonSenseQA	
	ACC	Samples	ACC	Samples
w/o Dilution	67.2	19.2	52.3	26.3
w/o Window	65.0	4.7	49.7	7.9
w/o Margin	67.8	19.0	52.7	27.3
w/o Share	67.6	21.6	52.6	29.2
Full	**67.8**	**18.8**	**52.7**	**26.1**

Table 3: Ablation study of CAWS components, evaluating their impact on accuracy and average sample count.

lier and avoid unnecessary computation on more straightforward inputs.

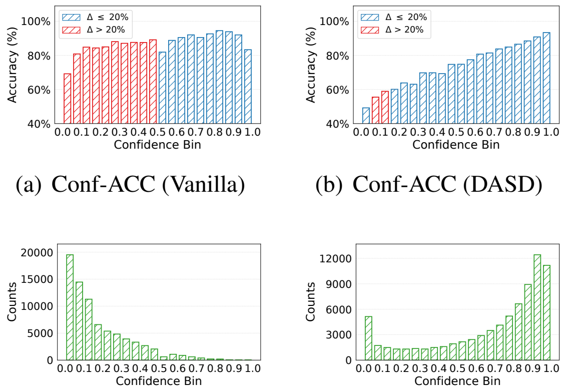
Compared to adaptive baselines such as ASC, CAWS yields more stable improvements, especially under ambiguous inputs. Its convergence-aware stopping mechanism better allocates compute based on observed answer stability, mitigating premature halts or redundant sampling. On DeepSeek + ARC, for instance, CAWS reaches 67.8% accuracy with only 18.8 samples, while ASC uses more steps for similar performance. These results validate that combining DASD and CAWS enables accurate and efficient reasoning. By calibrating internal confidence scores and dynamically adapting sampling, our framework offers a practical solution for scalable LLM inference.

4.5 Ablation Study

We conduct ablations using DeepSeek-R1-Distill-Qwen-1.5B to isolate the effects of both training-time calibration and inference-time control, as shown in Table 3. First, removing DASD and training with unadjusted confidence scores ("w/o Dilution") results in consistent accuracy degradation, particularly on ARC-Challenge (-0.6). This confirms that miscalibrated internal signals undermine adaptive inference and highlights the importance of DASD's semantic redundancy-aware dilution for producing reliable confidence estimates. Next, ablating the convergence criteria individually reveals their complementary roles. Removing the stability check ("w/o Window") significantly reduces accuracy and leads to overly aggressive stopping, while discarding dominance ("w/o Share") or margin gap ("w/o Margin") causes inefficient oversampling. In particular, the absence of margin control increases sample cost on CSQA (+1.2) without any accuracy gain, showing that convergence without discriminative dominance is suboptimal. Together, these results underscore that both DASD and CAWS are critical for achieving robust and efficient inference.

4.6 Confidence Calibration Analysis

We analyze whether DASD improves the alignment between model-predicted confidence scores and actual answer correctness. Figure 2 shows the confidence-accuracy curves. In the vanilla setting, the curves exhibit clear misalignment: predicted confidence does not reliably correspond to empirical accuracy, especially at higher confidence levels. In contrast, models trained with DASD produce more monotonic and well-aligned curves, indicating improved calibration. We also visualize the distribution of predictions across confidence bins. Without calibration, confidence scores are heavily skewed toward the upper end. DASD spreads



(a) Conf-ACC (Vanilla) (b) Conf-ACC (DASD)

(c) Conf-Counts (Vanilla) (d) Conf-Counts (DASD)

Figure 2: Top: Confidence-accuracy alignment curves. Bottom: Sample distribution across confidence bins. We compare models with and without DASD calibration.

the scores more evenly, reducing misleading score concentration and producing more faithful signals.

Figure 3 presents ROC curves comparing models trained with and without DASD to further assess the discriminative utility of confidence scores. We observe consistent AUC improvements: on ARC-Challenge, the AUC increases from 0.65 to 0.71; on CommonsenseQA, from 0.64 to 0.72; and on GSM8K, from 0.69 to 0.73. The largest gains are observed on CommonsenseQA, where confidence-based separation is particularly challenging. These results indicate that DASD enhances the model's ability to distinguish correct from incorrect answers based on internal confidence, improving the reliability of adaptive inference.

To quantitatively assess calibration, we report Expected Calibration Error (ECE) (Guo et al., 2017) across all datasets and model families (Table 4). DASD consistently reduces calibration error compared to the vanilla setting, indicating that the diluted pseudo-labels indeed improve the alignment between predicted confidence and empirical correctness. This improvement is crucial because CAWS relies on confidence signals to decide convergence: better calibrated signals directly translate into more robust stopping behavior and reduced risk of premature or excessive sampling. The results demonstrate that the training-time calibration, though lightweight, substantially enhances the reliability of downstream adaptive inference.

4.7 Parameter Sensitivity Analysis

We analyze how CAWS responds to changes in three key hyperparameters: the margin threshold θ_{margin}, the share threshold θ_{share}, and the stability window size M, using the GSM8K dataset with the LLaMA-3.1-8B-Instruct model. Figure 4 shows their effects on sample usage and accuracy.

Model	ARC-Challenge		CommonSenseQA		GSM8K	
	Vanilla	DASD	Vanilla	DASD	Vanilla	DASD
DeepSeek	12.29	8.86	7.36	5.27	46.66	42.51
Llama	6.70	4.45	15.42	8.09	13.89	9.64
Qwen	54.33	9.08	63.36	14.42	87.17	82.61

Table 4: Expected Calibration Error (ECE) analysis across three datasets and model families. Lower values indicate better alignment between predicted confidence and accuracy. DASD consistently reduces calibration error compared to the vanilla setting, demonstrating its effectiveness in improving confidence reliability.

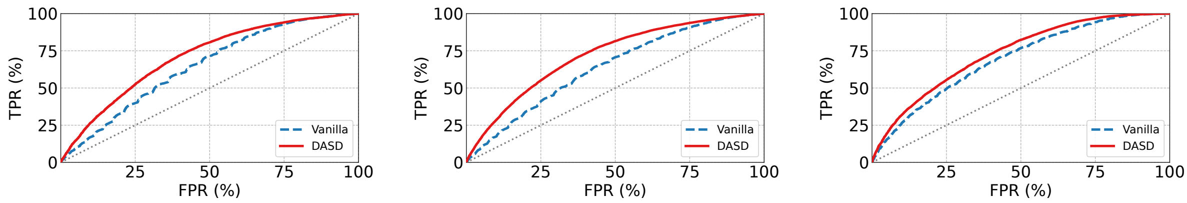
p	ARC-Challenge	CommonSenseQA	GSM8K
0.0	0.65	0.64	0.69
0.1	**0.71**	**0.72**	**0.73**
0.2	0.72	0.69	0.72
0.3	0.69	0.67	0.72

Table 5: Impact of the dilution exponent p on the AUC calibration metric across three datasets.

Increasing the margin threshold improves accuracy to a moderate value but leads to diminishing returns and increased sample cost beyond that point (Figure 4(a)). The vote share threshold exhibits a similar trade-off: values between 0.6 and 0.7 promote answer reliability but may delay convergence on simpler inputs (Figure 4(b)). The window size influences stability: smaller values risk premature termination, while larger ones enhance robustness at a modest computational cost (Figure 4(c)). These trends indicate that CAWS performs best under moderately conservative settings, where the stopping condition balances decision confidence with efficiency. Careful tuning of these parameters can further optimize performance under different task conditions.

4.8 Sensitivity to Dilution Exponent

We analyze the impact of the dilution exponent p on the AUC calibration metric (Table 5). The results reveal a clear non-linear trend: small values (e.g., $p = 0.1$) substantially improve calibration by mitigating redundancy-driven overconfidence, while overly large values (e.g., $p = 0.3$) may excessively penalize clusters and degrade signal quality. The optimal balance is achieved at $p = 0.1$, which consistently yields the highest AUC across all benchmarks. This analysis validates our default

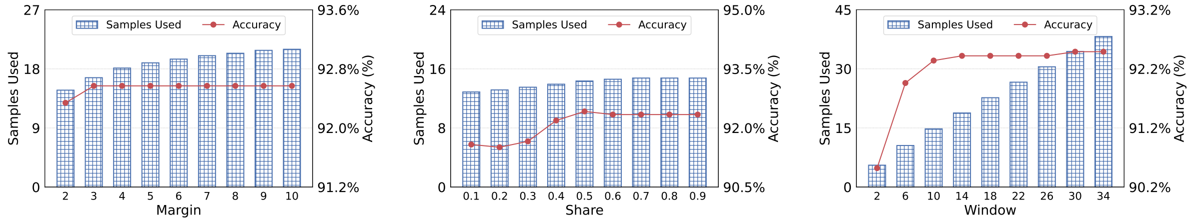


(a) ARC-Challenge

(b) CommonSenseQA

(c) GSM8K

Figure 3: Receiver Operating Characteristic (ROC) curves across three benchmarks. Models trained with DASD consistently achieve higher AUC than the vanilla counterparts, indicating improved confidence scores.



(a) Margin Threshold (θ_{margin})

(b) Vote Share Threshold (θ_{share})

(c) Stability Window Size (M)

Figure 4: Sensitivity analysis of CAWS stopping parameters. We evaluate the effect of varying the margin threshold (θ_{margin}), vote share threshold (θ_{share}), and stability window size (M) on accuracy and average sample usage.

Model	Dataset	Tokens	Overhead
Llama	ARC-Challenge	6512.7	0.18%
	CommonSenseQA	6503.4	0.18%
	GSM8K	6940.8	0.17%
DeepSeek	ARC-Challenge	15142.6	0.07%
	CommonSenseQA	17844.8	0.06%
	GSM8K	14447.8	0.08%
Qwen	ARC-Challenge	5205.7	0.23%
	CommonSenseQA	5009.4	0.24%
	GSM8K	5427.2	0.22%

Table 6: Token-level overhead introduced by confidence querying during inference. Across all datasets and model families, the overhead remains below 0.3%, confirming that the cost of confidence estimation is negligible relative to total decoding.

hyperparameter choice and highlights that moderate dilution is sufficient to stabilize model self-assessments without distorting their discriminative capacity. Together, these findings emphasize that DASD's calibration effect is both principled and robust across datasets.

4.9 Overhead Evaluation

We evaluate the inference-time overhead introduced by querying internal confidence using auxiliary prompts. As shown in Table 6, the additional token usage ranges from 0.06% to 0.24% across datasets and models, indicating that the cost of confidence estimation is negligible relative to total decoding. In contrast, reward-model-based methods typically require a full forward pass for each candidate output, resulting in 100% additional overhead in memory and computation. Our method achieves comparable benefits in answer selection at a fraction of the cost, making it substantially more efficient and deployable in practical settings. Moreover, since the overhead remains stable across model scales, DASD+CAWS can be seamlessly integrated into real-world pipelines without incurring measurable latency, which is particularly valuable for large-scale deployment scenarios where efficiency is critical.

5 Conclusion

This work addresses a central challenge in LLM reasoning: making reliable answer selections under uncertainty using only internal model signals. We propose a unified framework that calibrates self-confidence through Diversity-Aware Self-Signal Dilution (DASD) and leverages these signals via Convergent Adaptive Weighted Sampling (CAWS) for inference-time control. Together, these components enable LLMs to reason more efficiently and reliably, dynamically allocating compute based on internal confidence signals while improving the utility of model self-assessments. Experiments across three reasoning benchmarks and multiple model scales show that our approach achieves substantial efficiency gains while maintaining or even improving accuracy.

Limitations

While our framework enables efficient and self-contained inference without relying on external supervision, several limitations remain. First, the confidence signals are derived through prompt-based querying, which may offer limited calibration fidelity, especially under distributional shifts or adversarial conditions. Second, our experiments focus on single-turn, factoid-style QA tasks where answers are well-defined and easily verifiable. Extending our method to open-ended settings such as multi-turn dialogue or code generation poses new challenges, as aggregating diverse outputs or determining convergence becomes less straightforward when there is no single correct answer. Third, while CAWS performs well on average, it may fail to converge or terminate prematurely on ambiguous or low-signal examples. Understanding and mitigating such failure modes remains an essential direction for improving the robustness of adaptive inference. Furthermore, in safety-critical domains, overly confident convergence on incorrect outputs may lead to hard-to-detect failure cases. Although we observe no such failure in our evaluation domains, future work should explore safeguards to mitigate the amplification of misleading reasoning patterns under uncertainty. This work does not raise any ethical issues.

Acknowledgments

This work is supported by Shenzhen R&D Program under grant NO. KJZD20230923114059020, and in part by Shenzhen Key Laboratory of Software Defined Networking under Grant ZDSYS20140509172959989.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*.

Sudhanshu Agrawal, Wonseok Jeon, and Mingu Lee. 2024. Adaedl: Early draft stopping for speculative decoding of large language models via an entropy-based lower bound on token acceptance probability. *arXiv preprint arXiv:2410.18351*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Umesh Bodhwani, Yuan Ling, Shujing Dong, Yarong Feng, and Hongfei Li. 2025. A calibrated reflection approach for enhancing confidence estimation in llms. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 399–411.

Jianhao Chen, Zishuo Xun, Bocheng Zhou, Han Qi, Qiaosheng Zhang, Yang Chen, Wei Hu, Yuzhong Qu, Wanli Ouyang, and Shuyue Hu. 2025. Do we truly need so many samples? multi-llm repeated sampling efficiently scale test-time compute. *arXiv preprint arXiv:2504.00762*.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023a. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei A Zaharia, and James Y Zou. 2024. Are more llm calls all you need? towards the scaling properties of compound ai systems. *Advances in Neural Information Processing Systems*, 37:45767–45790.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Cheng-Han Chiang and Hung-yi Lee. 2024. Over-reasoning and redundant calculation of large language models. *arXiv preprint arXiv:2401.11467*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Antonio A Ginart, Naveen Kodali, Jason Lee, Caiming Xiong, Silvio Savarese, and John R Emmons. 2025. Lz penalty: An information-theoretic repetition penalty for autoregressive language models. *arXiv preprint arXiv:2504.20131*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

David Herrera-Poyatos, Carlos Peláez-González, Cristina Zuheros, Andrés Herrera-Poyatos, Virilo Tejedor, Francisco Herrera, and Rosana Montes. 2025. An overview of model uncertainty and variability in llm-based sentiment analysis. challenges, mitigation strategies and the role of explainability. *arXiv preprint arXiv:2504.04462*.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Dylan J Foster, and Akshay Krishnamurthy. 2025a. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025b. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*.

James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2024a. Deal: Decoding-time alignment for large language models. *arXiv preprint arXiv:2402.06147*.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo M Ponti, and Ivan Titov. 2024b. Post-hoc reward calibration: A case study on length bias. *arXiv preprint arXiv:2409.17407*.

Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, and 1 others. 2023. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint arXiv:2303.06135*.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Jia Xu, Linjian Mo, and Min Zhang. 2025. Test-time computing: from system-1 thinking to system-2 thinking. *arXiv preprint arXiv:2501.02497*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2024. Taming overconfidence in llms: Reward calibration in rlhf. *arXiv preprint arXiv:2410.09724*.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025a. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Zichong Li, Xinyu Feng, Yuheng Cai, Zixuan Zhang, Tianyi Liu, Chen Liang, Weizhu Chen, Haoyu Wang, and Tuo Zhao. 2025b. Llms can generate a better answer by aggregating their own responses. *arXiv preprint arXiv:2503.04104*.

Qin Liu, Wenxuan Zhou, Nan Xu, James Y Huang, Fei Wang, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2025a. Metascale: Test-time scaling with evolving meta-thoughts. *arXiv preprint arXiv:2503.13447*.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025b. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*.

Ruijie Miao, Yihan Yan, Xinshuo Yao, and Tong Yang. 2024a. An efficient inference framework for early-exit large language models. *arXiv preprint arXiv:2407.20272*.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024b. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. 2023. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. *arXiv preprint arXiv:2505.04881*.

Ali Razghandi, Seyed Mohammad Hadi Hosseini, and Mahdieh Soleymani Baghshah. 2025. Cer: Confidence enhanced reasoning in llms. *arXiv preprint arXiv:2502.14634*.

Mathieu Rita, Florian Strub, Rahma Chaabouni, Paul Michel, Emmanuel Dupoux, and Olivier Pietquin. 2024. Countering reward over-optimization in llm with demonstration-guided reinforcement learning. *arXiv preprint arXiv:2404.19409*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*.

Nishad Singhi, Hritik Bansal, Arian Hosseini, Aditya Grover, Kai-Wei Chang, Marcus Rohrbach, and Anna Rohrbach. 2025. When to solve, when to verify: Compute-optimal problem solving and generative verification for llm reasoning. *arXiv preprint arXiv:2504.01005*.

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

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*.

Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas W Mayer, and Padhraic Smyth. 2025. What large language models know and what people think they know. *Nature Machine Intelligence*, pages 1–11.

Fengfei Sun, Ningke Li, Kailong Wang, and Lorenz Goette. 2025. Large language models are overconfident and amplify human bias. *arXiv preprint arXiv:2505.02151*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *arXiv preprint arXiv:2410.20290*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Zhendong Tan, Xingjun Zhang, Chaoyi Hu, Yancheng Pan, and Shaoxun Wang. 2025. Adaptive rectification sampling for test-time compute scaling. *arXiv preprint arXiv:2504.01317*.

Wenxin Tang, Jingyu Xiao, Wenxuan Jiang, Xi Xiao, Yuhang Wang, Xuxin Tang, Qing Li, Yuehe Ma, Junliang Liu, Shisong Tang, and 1 others. 2025. Slidecoder: Layout-aware rag-enhanced hierarchical slide generation from design. *arXiv preprint arXiv:2506.07964*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*.

Yuxuan Wan, Yi Dong, Jingyu Xiao, Yintong Huo, Wenxuan Wang, and Michael R Lyu. 2024. Mrweb: An exploration of generating multi-page resource-aware web code from ui designs. *arXiv preprint arXiv:2412.15310*.

Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, and 1 others. 2025a. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*.

Tianchun Wang, Zichuan Liu, Yuanzhou Chen, Jonathan Light, Haifeng Chen, Xiang Zhang, and Wei Cheng. 2025b. Diversified sampling improves scaling llm inference. *arXiv preprint arXiv:2502.11027*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xiaobao Wu. 2025. Sailing ai by the stars: A survey of learning from rewards in post-training and test-time scaling of large language models. *arXiv preprint arXiv:2505.02686*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zhiyao Xu, and Michael R Lyu. 2024. Interaction2code: How far are we from automatic interactive webpage generation? *arXiv preprint arXiv:2411.03292*.

Jingyu Xiao, Ming Wang, Man Ho Lam, Yuxuan Wan, Junliang Liu, Yintong Huo, and Michael R Lyu. 2025a. Designbench: A comprehensive benchmark for mllm-based front-end code generation. *arXiv preprint arXiv:2506.06251*.

Jingyu Xiao, Zhongyi Zhang, Yuxuan Wan, Yintong Huo, Yang Liu, and Michael R. Lyu. 2025b. Efficientuicoder: Efficient mllm-based ui code generation via input and output token compression. *Preprint*, arXiv:2509.12159.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, and Xipeng Qiu. 2025. Revisiting the test-time scaling of o1-like models: Do they truly possess test-time scaling capabilities? *arXiv preprint arXiv:2502.12215*.

Kexun Zhang, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. 2024. Scaling llm inference with optimized sample compute allocation. *arXiv preprint arXiv:2410.22480*.

Yinmin Zhong, Zili Zhang, Bingyang Wu, Shengyu Liu, Yukun Chen, Changyi Wan, Hanpeng Hu, Lei Xia, Ranchen Ming, Yibo Zhu, and 1 others. 2024. Rlhfuse: Efficient rlhf training for large language models with inter-and intra-stage fusion. *arXiv preprint arXiv:2409.13221*.

Appendix

A Notation Summary

Table 7 summarizes the key symbols used in our framework, grouped by training-time calibration (Diversity-Aware Self-Signal Dilution, DASD) and inference-time control (Convergent Adaptive Weighted Sampling, CAWS).

Diversity-Aware Self-Signal Dilution	
x	Input question or prompt
y_i	Model-generated reasoning path (sample i)
z_i	Final parsed answer from y_i
$s^{(i)}$	Model-assigned confidence score for y_i
$\hat{R}(y_i)$	Calibrated confidence assigned by DASD
$\mathcal{C}_k$	Cluster of samples sharing answer z_k
D_k	Semantic diversity estimate of cluster $\mathcal{C}_k$
α_k	Soft dilution factor for cluster $\mathcal{C}_k$
S_k^{adj}	Adjusted score for cluster k after dilution
Convergent Adaptive Weighted Sampling	
$w^{(i)}$	Smoothed vote weight for sample $y^{(i)}$
Score(z)	Aggregated vote score for answer z
z_t^*	Leading candidate answer at inference step t
M	Stability window for CAWS stopping
θ_{share}	Vote share threshold for stopping
θ_{margin}	Vote margin threshold for stopping
$N_{\max}$	Maximum number of samples per input

Table 7: Summary of notations used in Diversity-Aware Self-Signal Dilution (DASD) and Convergent Adaptive Weighted Sampling (CAWS).

B Prompt Templates

We provide the exact prompt formats used in our framework for semantic similarity evaluation and confidence querying. All prompts are executed using the same base LLM via in-context decoding.

Semantic Similarity Prompt. This prompt is used to evaluate whether two reasoning paths convey different semantic content. It enables diversity estimation for answer clusters during training (DASD), and follows a format similar to (Jiang et al., 2023; Li et al., 2025b).

> Please rate the **semantic similarity** between the following two reasoning paths on a scale of 1 to 10.
>
> **Text A:** [Candidate A]
> **Text B:** [Candidate B]
>
> Similarity score (1–10):

Confidence Query Prompt. This prompt is used to estimate whether a generated answer appears correct to the LLM. Following prior work on LLM-as-a-judge (Huang et al., 2025b; Singhi et al., 2025), we adopt their same prompt format to query the model's internal judgment without external supervision. The resulting signal supports our unsupervised confidence calibration in DASD.

> **Question:** [Input x]
> **Answer:** [Generated output y]
>
> Is the answer correct? (Yes / No)

C Algorithm Pseudocode

Algorithm 1 outlines the inference procedure of Convergent Adaptive Weighted Sampling (CAWS). The algorithm iteratively samples reasoning paths and maintains a confidence-weighted score distribution over candidate answers. At each step, it evaluates convergence based on the stability of the top-scoring answer within a sliding window of size M, along with two additional criteria: dominance in vote share and margin over alternatives. The algorithm stops early when a sufficiently consistent and confident answer emerges. If no answer meets the stopping criteria within $N_{\max}$ samples, the final output is selected as the answer with the highest accumulated confidence score.

D Efficiency-Accuracy Tradeoff

Table 8 presents a detailed comparison of accuracy under varying sampling budgets on datasets. The results are based on models trained with DASD to ensure calibrated confidence scores. We vary the confidence threshold for ASC and Early Stopping to control the average number of samples. For CAWS, we achieve budget alignment by adjusting its stopping parameters. At each budget level, we report the configuration that yields an average sample count closest to the target.

Across all datasets and models, CAWS consistently achieves stronger or comparable accuracy with significantly fewer samples than both fixed-budget and adaptive baselines. On ARC-Challenge, CAWS reaches 68.2% with DeepSeek and 91.6% with Qwen at a 16-sample budget, outperforming SC and WSC (both at 64 samples) as well as ASC, which achieves only 66.9% (DeepSeek) and 91.4% (Qwen) under the same budget. On CommonsenseQA with Qwen, CAWS achieves 83.4% at 16

Algorithm 1 CAWS

```
Require: Input $x$, model $p_\theta$, max samples $N_{\max}$, stability window $M$, thresholds $\theta_{\text{share}}$, $\theta_{\text{margin}}$
Initialize Score $\leftarrow \{\}$; history buffer $\mathcal{H} \leftarrow []$
for $t = 1$ to $N_{\max}$ do
    Sample $y^{(t)} \sim p_\theta(\cdot \mid x)$
    $z^{(t)} \leftarrow \texttt{extract\_answer}(y^{(t)})$
    $s^{(t)} \leftarrow \texttt{confidence\_query}(x, y^{(t)})$
    $w^{(t)} \leftarrow \sigma(s^{(t)})$
    Score$(z^{(t)})$ += $w^{(t)}$
    Append $z^{(t)}$ to $\mathcal{H}$
    if $t \geq M$ then
        $z^* \leftarrow \arg\max_z$ Score$(z)$
        stable $\leftarrow \mathcal{H}[-M:] = [z^*]^M$
        Compute share via Eq. (11)
        Compute margin via Eq. (12)
        if stable and (share $\geq \theta_{\text{share}}$ or margin $\geq \theta_{\text{margin}}$) then
            return $z^*$
        end if
    end if
end for
return $\arg\max_z$ Score$(z)$
```

samples, matching WSC and outperforming ASC (83.0%) and Early Stopping (83.1%) at similar cost. On GSM8K, CAWS maintains accuracy above 92% with LLaMA and Qwen using only 16 samples, while other methods either require more budget or show accuracy degradation as sampling decreases. These results demonstrate that CAWS achieves a more favorable efficiency-accuracy tradeoff than existing methods, making it a practical and scalable solution for compute-efficient LLM reasoning.

E Dataset Licenses

We report the licenses for all datasets used in our experiments:

- **GSM8K**[2] (Cobbe et al., 2021): Released under the MIT License.
- **CommonsenseQA**[3] (Talmor et al., 2018): Released under the MIT License.
- **ARC-Challenge**[4] (Clark et al., 2018): Released under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license.

All datasets are publicly released and have served as standard benchmarks in reasoning with large language models.

[2] https://huggingface.co/datasets/openai/gsm8k

[3] https://huggingface.co/datasets/tau/commonsense_qa

[4] https://huggingface.co/datasets/allenai/ai2_arc

Table 8: Accuracy (%) on ARC-Challenge, CommonsenseQA, and GSM8K under varying numbers of shots (2^x). For 2^1 to 2^6, each value shows the absolute change (↑ or ↓) relative to SC under the same setting. Equal values are marked as ↑0.0. Within each group, the highest score is **bolded** and the second-highest is underlined.

Model	Method	2^0	2^1	2^2	2^3	2^4	2^5	2^6
ARC-Challenge								
DeepSeek	Early Stopping	59.4	**59.4** ↑0.0	58.3 ↓6.2	58.2 ↓9.3	60.3 ↓7.3	64.3 ↓3.3	67.8 ↑0.2
	SC	59.4	**59.4**	64.5	**67.5**	67.6	67.6	67.6
	WSC	59.4	59.2 ↓0.2	64.3 ↓0.2	67.4 ↓0.1	66.9 ↓0.7	67.7 ↑0.1	67.8 ↑0.2
	Best-of-N	59.4	59.2 ↓0.2	59.2 ↓5.3	61.2 ↓6.3	59.1 ↓8.5	57.8 ↓9.8	56.9 ↓10.7
	ASC	-	**59.4** ↑0.0	**64.7** ↑0.2	64.9 ↓2.6	66.9 ↓0.7	66.7 ↓0.9	66.7 ↓0.9
	CAWS	-	-	-	65.0 ↓2.5	**68.2** ↑0.6	**68.2** ↑0.6	**68.2** ↑0.6
Llama	Early Stopping	81.6	**84.6** ↑3.0	84.8 ↑0.2	85.6 ↓0.8	86.1 ↓1.6	86.1 ↓1.5	**88.7** ↑0.8
	SC	81.6	81.6	84.6	86.4	87.7	87.6	87.9
	WSC	81.6	**84.6** ↑3.0	**85.8** ↑1.2	87.5 ↑1.1	**88.1** ↑0.4	**88.2** ↑0.6	**88.7** ↑0.8
	Best-of-N	81.6	**84.6** ↑3.0	85.5 ↑0.9	86.0 ↓0.4	86.0 ↓1.7	86.0 ↓1.6	86.3 ↓1.6
	ASC	-	81.6 ↑0.0	85.4 ↑0.8	87.6 ↑1.2	87.5 ↓0.2	87.5 ↓0.1	87.5 ↓0.4
	CAWS	-	-	85.4 ↑0.8	**87.7** ↑1.3	88.0 ↑0.3	88.1 ↑0.5	88.1 ↑0.2
Qwen	Early Stopping	89.1	**89.3** ↑0.2	89.4 ↓1.6	89.4 ↓1.6	89.4 ↓1.6	89.4 ↓2.2	91.1 ↓0.5
	SC	89.1	89.1	91.0	91.0	91.0	**91.6**	**91.6**
	WSC	89.1	**89.3** ↑0.2	89.9 ↓1.1	90.4 ↓0.6	90.8 ↓0.2	91.0 ↓0.6	91.1 ↓0.5
	Best-of-N	89.1	**89.3** ↑0.2	89.3 ↓1.7	89.5 ↓1.5	89.3 ↓1.7	89.9 ↓1.7	90.4 ↓1.2
	ASC	-	89.1 ↑0.0	91.3 ↑0.3	91.4 ↑0.4	91.4 ↑0.4	91.4 ↓0.2	91.4 ↓0.2
	CAWS	-	-	**91.5** ↑0.5	**91.6** ↑0.6	**91.6** ↑0.6	**91.6** ↑0.0	**91.6** ↑0.0
CommonSenseQA								
DeepSeek	Early Stopping	43.0	42.7 ↓0.3	43.2 ↓4.9	42.5 ↓7.0	44.2 ↓8.8	43.7 ↓9.0	**53.0** ↑0.3
	SC	43.0	43.0	**48.1**	49.5	**53.0**	52.7	52.7
	WSC	43.0	**44.2** ↑1.2	47.7 ↓0.4	**50.8** ↑1.3	52.7 ↓0.3	52.7 ↑0.0	**53.0** ↑0.3
	Best-of-N	43.0	**44.2** ↑1.2	43.6 ↓4.5	42.4 ↓7.1	43.1 ↓9.9	41.7 ↓11.0	40.9 ↓11.8
	ASC	-	43.0 ↑0.0	43.0 ↓5.1	49.7 ↑0.2	50.0 ↓3.0	51.6 ↓1.1	51.6 ↓1.1
	CAWS	-	-	-	49.7 ↑0.2	51.7 ↓1.3	**52.8** ↑0.1	52.8 ↑0.1
Llama	Early Stopping	72.3	73.7 ↑1.4	75.4 ↓0.6	76.7 ↓0.8	76.7 ↓1.7	76.7 ↓1.8	**79.4** ↑0.1
	SC	72.3	72.3	76.0	77.5	78.4	78.5	79.3
	WSC	72.3	**75.7** ↑3.4	76.6 ↑0.6	**78.4** ↑0.9	78.5 ↑0.1	79.2 ↑0.7	**79.4** ↑0.1
	Best-of-N	72.3	**75.7** ↑3.4	**76.7** ↑0.7	77.6 ↑0.1	77.3 ↓1.1	77.2 ↓1.3	77.7 ↓1.6
	ASC	-	72.3 ↑0.0	76.3 ↑0.3	77.9 ↑0.4	78.8 ↑0.4	78.8 ↑0.3	78.8 ↓0.5
	CAWS	-	-	76.1 ↑0.1	**78.4** ↑0.9	**78.9** ↑0.5	**79.3** ↑0.8	79.3 ↑0.0
Qwen	Early Stopping	80.4	81.2 ↑0.8	81.8 ↑0.0	82.7 ↑0.0	83.1 ↑0.0	**83.6** ↑0.2	**83.8** ↑0.1
	SC	80.4	80.4	81.8	82.7	83.1	83.4	83.7
	WSC	80.4	**82.0** ↑1.6	81.7 ↓0.1	82.4 ↓0.3	**83.5** ↑0.4	83.3 ↓0.1	**83.8** ↑0.1
	Best-of-N	80.4	**82.0** ↑1.6	81.8 ↑0.0	81.9 ↓0.8	82.2 ↓0.9	82.9 ↓0.5	83.2 ↓0.5
	ASC	-	80.4 ↑0.0	81.9 ↑0.1	**83.0** ↑0.3	83.0 ↓0.1	83.0 ↓0.4	83.0 ↓0.7
	CAWS	-	-	**82.1** ↑0.3	82.9 ↑0.2	83.4 ↑0.3	83.4 ↑0.0	83.4 ↓0.3
GSM8K								
DeepSeek	Early Stopping	72.2	72.4 ↑0.2	71.0 ↓10.7	71.3 ↓14.3	71.1 ↓16.5	72.1 ↓15.9	88.4 ↓0.2
	SC	72.2	72.2	**81.7**	**85.6**	87.6	88.0	**88.6**
	WSC	72.2	**72.6** ↑0.4	79.8 ↓1.9	83.8 ↓1.8	86.8 ↓0.8	88.4 ↑0.4	88.4 ↓0.2
	Best-of-N	72.2	**72.6** ↑0.4	72.9 ↓8.8	71.3 ↓14.3	69.6 ↓18.0	71.4 ↓16.6	68.9 ↓19.7
	ASC	-	72.2 ↑0.0	72.2 ↓9.5	72.2 ↓13.4	87.2 ↓0.4	87.3 ↓0.7	87.3 ↓1.3
	CAWS	-	-	-	85.4 ↓0.2	**88.2** ↑0.6	**88.5** ↑0.5	88.5 ↓0.1
Llama	Early Stopping	80.4	**85.2** ↑4.8	86.1 ↓0.6	85.8 ↓4.5	86.4 ↓5.3	88.3 ↓3.7	92.5 ↑0.1
	SC	80.4	80.4	86.7	90.3	91.7	92.0	92.4
	WSC	80.4	84.0 ↑3.6	**88.9** ↑2.2	91.1 ↑0.8	92.1 ↑0.4	92.1 ↑0.1	92.5 ↑0.1
	Best-of-N	80.4	84.0 ↑3.6	86.4 ↓0.3	86.9 ↓3.4	86.8 ↓4.9	86.6 ↓5.4	86.6 ↓5.8
	ASC	-	80.4 ↑0.0	80.4 ↓6.3	90.6 ↑0.3	91.4 ↓0.3	91.4 ↓0.6	91.4 ↓1.0
	CAWS	-	-	-	**91.7** ↑1.4	**92.6** ↑0.9	**92.6** ↑0.6	**92.6** ↑0.2
Qwen	Early Stopping	87.6	87.6 ↑0.0	87.6 ↓5.7	87.6 ↓6.2	91.2 ↓3.3	93.0 ↓1.5	**94.9** ↑0.3
	SC	87.6	87.6	93.3	93.8	94.5	94.5	94.6
	WSC	87.6	**91.2** ↑3.6	93.3 ↑0.0	93.9 ↑0.1	94.6 ↑0.1	**94.8** ↑0.3	**94.9** ↑0.3
	Best-of-N	87.6	**91.2** ↑3.6	91.6 ↓1.7	91.4 ↓2.4	91.6 ↓2.9	91.4 ↓3.1	90.8 ↓3.8
	ASC	-	87.6 ↑0.0	93.8 ↑0.5	94.6 ↑0.8	94.6 ↑0.1	94.6 ↑0.1	94.6 ↑0.0
	CAWS	-	-	**93.9** ↑0.6	**94.7** ↑0.9	**94.7** ↑0.2	94.7 ↑0.2	94.7 ↑0.1