DisCal: Distribution-Aware Calibration for Mathematical Reasoning Under Character-Level Noisy Inputs

Bo Zhang[1,2,3†], Jiawei Zhang[4], Cong Gao[5], Bingxu Han[4], Minghao Hu[2,3], Jun Zhang[2,3*], Yunbo Cao[2,3], Zhunchen Luo[2,3], Wen Yao[6], Guotong Geng[2,3], Zhong Wang[1*]

[1]PLA Rocket Force University of Engineering [2]Center of Information Research, PLA AMS
[3]Discipline and Technology Research Center for Large Model Intelligence Applications
[4]Shandong University [5]Nankai University [6]Defense Innovation Institute, PLA AMS
mcgrady150318@163.com, dsp863wang@163.com

Abstract

Although large reasoning models (LRMs) exhibit exceptional mathematical reasoning capabilities on clean inputs, their reasoning accuracy drops substantially in the presence of character-level noise such as typographical errors. Critically, their confidence estimates fail to reflect the corresponding decline in reasoning accuracy. While confidence calibration offers a principled solution, existing methods predominantly target clean inputs, leaving noisy scenarios largely unexplored. To address this gap, we propose **DisCal** (**Dis**tribution-aware **Cal**ibration), a confidence calibration framework for character-level noisy inputs. DisCal extracts uncertainty signals from both the empirical answer distribution and the model's predictive distribution, and integrates them via a learned calibrator to produce well-calibrated confidence. Experiments across multiple mathematical reasoning benchmarks demonstrate that DisCal consistently outperforms existing calibration methods under noisy inputs, reducing Expected Calibration Error (ECE) by up to 39.21% and improving Area Under the Receiver Operating Characteristic Curve (AUROC) by up to 31.44%.

1 Introduction

Large Reasoning Models (LRMs) have demonstrated substantial advances in reasoning capabilities (Georgiev et al., 2024; Guo et al., 2025), performing particularly well on mathematical reasoning tasks with clean inputs (Wei et al., 2022; Yu et al., 2025b). However, LRMs in practice frequently encounter character-level noise such as typographical errors (Aliakbarzadeh et al., 2025). Recent studies have demonstrated that noise, even when it does not alter the underlying semantics of a problem, can still significantly impair reasoning performance: real-world noise disproportionately harms mathematical reasoning (Aliakbarzadeh et al., 2025); and small perturbations can break reasoning chains (Yang et al., 2025b; Huang et al., 2025; Hao et al., 2025). While prior work primarily examines whether models produce correct answers under noisy inputs, model confidence, crucial for reliable deployment, remains largely unexplored. A critical question persists: ***How does character-level noise affect model confidence?***

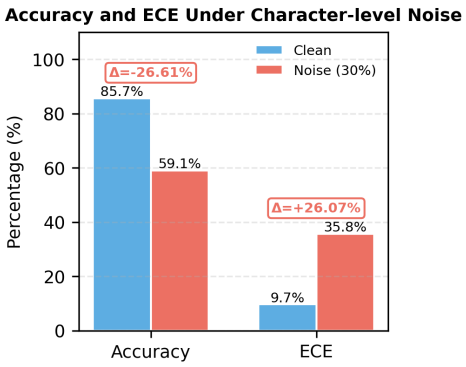

Figure 1: Adding 30% character-level noise to the inputs, DeepSeek-7B on the GSM8K benchmark exhibits a significant confidence-accuracy mismatch: while answer accuracy declines sharply, the ECE increases substantially.

To investigate this question, we introduce controlled character-level noise by randomly selecting 30% of non-numerical words and applying a random perturbation (substitution, insertion, deletion, or swap) to each selected word. For example, the problem "What is 15 divided by 3?" can be perturbed to "What is 15 diided by 3?", preserving numerical values while introducing realistic typographical errors. We quantify model confidence as the geometric mean of token-level likelihoods assigned to the answer tokens. As shown in Figure 1, character-level noise causes answer accuracy to

*Corresponding authors.
†Work performed while an intern at the Center of Information Research, PLA Academy of Military Science.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14484–14507
July 2-7, 2026

drop sharply from 85.7% to 59.1%. Meanwhile, the ECE increases significantly from 9.7% to 35.8%, indicating a widening gap between confidence and accuracy. **Model confidence fails to decrease correspondingly with the decline in reasoning accuracy, resulting in severe overconfidence.** This highlights the urgent need for effective confidence calibration under noisy inputs. However, existing confidence calibration methods, such as token-level indicators (Kadavath et al., 2022), sampling-based consistency measures (Tao et al., 2024; Lyu et al., 2025), and self-evaluation approaches (Zong et al., 2025), are primarily designed for clean inputs, leaving the calibration challenge under noisy inputs largely unexplored.

To fill this gap, we propose DisCal (Distribution-aware Calibration), a confidence calibration framework for character-level noisy inputs. DisCal extracts three uncertainty signals: (i) distributional entropy and (ii) distributional stability, both derived from the empirical answer distribution obtained via multiple samplings, and (iii) intrinsic confidence from the model's predictive distribution. Distributional entropy exhibits heightened sensitivity to noise-induced uncertainty, while distributional stability captures the concentration of the answer distribution. These signals are linearly combined to derive a gating factor that modulates intrinsic confidence via multiplicative scaling. **This mechanism effectively discriminates between correct and incorrect predictions, yielding calibrated scores consistent with actual accuracy.**

In summary, our contributions are as follows:

- We reveal a critical confidence-accuracy mismatch under character-level noisy inputs: while reasoning accuracy declines sharply, model confidence fails to reflect the increased error risk, resulting in severe overconfidence.
- We propose DisCal, a confidence calibration framework for mathematical reasoning in LRMs under character-level noisy inputs.
- We conduct experiments on multiple LRMs across mathematical reasoning benchmarks of varying difficulty under character-level noise. The results demonstrate that DisCal consistently outperforms existing calibration methods.

2 Related Work

We review two closely related research areas: robustness analysis in mathematical reasoning and confidence calibration for large language models (LLMs).

2.1 Robustness Analysis of Mathematical Reasoning

Large language models exhibit pronounced vulnerability to input noise in mathematical reasoning tasks. Previous work shows that even mild semantic or arithmetic-preserving transformations cause sharp drops in accuracy, revealing strong sensitivity to surface-level changes (Li et al., 2024). Models are also easily distracted by task-irrelevant contextual information, indicating fragile semantic grounding during reasoning (Shi et al., 2023). Numerical noise, such as number rescaling or range expansion, substantially increases logical and arithmetic errors, exposing weak numerical invariance (Shrestha et al., 2025). Linguistic variations, including paraphrasing and expression-level changes, further disrupt reasoning consistency despite semantic equivalence (Kirtane et al., 2025). Broader robustness benchmarks confirm significant degradation under structural and contextual noise (Yu et al., 2025a; Shang et al., 2026), while even on clean inputs, systematic failures in multi-step reasoning persist (Zhang et al., 2026).

2.2 Confidence Calibration

Confidence calibration for LLMs is widely studied and can be broadly categorized into white-box and black-box approaches, depending on whether token-level probabilities or logits are required. White-box methods (Gawlikowski et al., 2023) rely on the model's internal predictive distribution. Mean Token Entropy (MTE) (Fadeeva et al., 2023) quantifies uncertainty by computing token-level entropy along the generated sequence, where lower entropy indicates higher certainty. These methods provide interpretable signals but require internal logits, limiting their use to open-weight or locally deployed models. Black-box methods (Lin et al., 2023) infer confidence solely from generated outputs or sampling behavior. Verbalized confidence prompts the model to explicitly articulate its confidence, which is then parsed from text. Self-critique (Zong et al., 2025) encourages models to reassess their own predictions through self-evaluation. Another major line of work lever-

ages output diversity: Self-consistency (Wang et al., 2022) estimates confidence via the agreement rate across multiple stochastic sampling generations. In contrast, semantic entropy (Kuhn et al., 2023) estimates uncertainty based on the semantic dispersion of sampled outputs. These black-box approaches are compatible with closed-source APIs and align with chain-of-thought reasoning.

While prior work examines robustness in terms of accuracy degradation and calibration methods for clean inputs (Zhang et al., 2019, 2023), confidence reliability under noisy mathematical reasoning remains underexplored. We propose DisCal, a calibration framework designed to address overconfidence under noisy input conditions.

3 Methodology

As shown in Figure 2, we introduce DisCal, a distribution-aware confidence calibration framework designed for noisy inputs. DisCal extracts uncertainty signals from both empirical answer distributions and the model's predictive distributions to improve calibration.

3.1 Simulated Noisy Input Scenario

We use a mathematical reasoning dataset $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^{N}$, where Q_i denotes a clean input question and A_i represents its reference answer. To study confidence calibration under noisy inference-time conditions, we apply random character-level perturbations to clean questions that preserve the underlying problem semantics. These perturbations simulate common typographical errors and other benign input noise encountered in real-world deployment.

For each question Q_i, a single perturbation operator is randomly sampled from $\mathcal{P} = \{\text{deletion, insertion, substitution, swap}\}$ and is applied to a subset of words. The proportion of modified words determines the perturbation strength, while numeric values and mathematical symbols are preserved to ensure semantic consistency and answer invariance. For each input question, a single perturbed instance Q_i^p is generated with the reference answer A_i unchanged, reflecting one-shot noisy queries encountered in practical inference settings.

The perturbation strength is fixed to 30% during evaluation to provide a controlled noisy input setting. The same perturbation strategy is used during training to mitigate overfitting to specific noise patterns. Additional implementation details and examples are provided in Appendix A.

3.2 Distribution-Aware Uncertainty Signals

To capture uncertainty under noisy inputs, DisCal extracts signals from two distributional sources: the *empirical answer distribution* induced by stochastic sampling, and the model's *internal predictive distribution* during token generation.

For each noisy input instance Q_i^p, the model is sampled $K = 5$ times to obtain a set of generated answers $\{a_{i,j}^p\}_{j=1}^{K}$, where each $a_{i,j}^p$ is extracted from the final boxed answer in the corresponding model output. These answers induce an empirical distribution that reflects the model's output uncertainty under noisy input. Among the K sampled answers, the first one $a_{i,1}^p$ is treated as the *primary answer*, representing the model's one-shot response under noisy input. All sampled answers are used to estimate the empirical distribution.

Let $\mathcal{A}_i^p$ denote the set of unique reasoning answers for question Q_i. For any answer value $z \in \mathcal{A}_i^p$, its empirical probability is defined as

$$\pi_i(z) = \frac{1}{K} \sum_{j=1}^{K} \mathbb{I}\left(a_{i,j}^p = z\right). \tag{1}$$

This empirical probability reflects the likelihood of each possible answer based on the model's sampled outputs under noisy input conditions.

Uncertainty from Empirical Answer Distribution. From the empirical distribution π_i, we extract two signals that characterize distributional properties:

Distributional Entropy quantifies the dispersion of the empirical answer distribution:

$$H_i = -\sum_{z \in \mathcal{A}_i^p} \pi_i(z) \log \pi_i(z). \tag{2}$$

Higher entropy values indicate greater dispersion in the empirical distribution, reflecting increased uncertainty induced by noisy inputs. This signal exhibits heightened sensitivity in high-uncertainty regions, which typically require stronger calibration interventions.

Distributional Stability measures the concentration of the empirical distribution:

$$S_i = \max_{z \in \mathcal{A}_i^p} \pi_i(z). \tag{3}$$

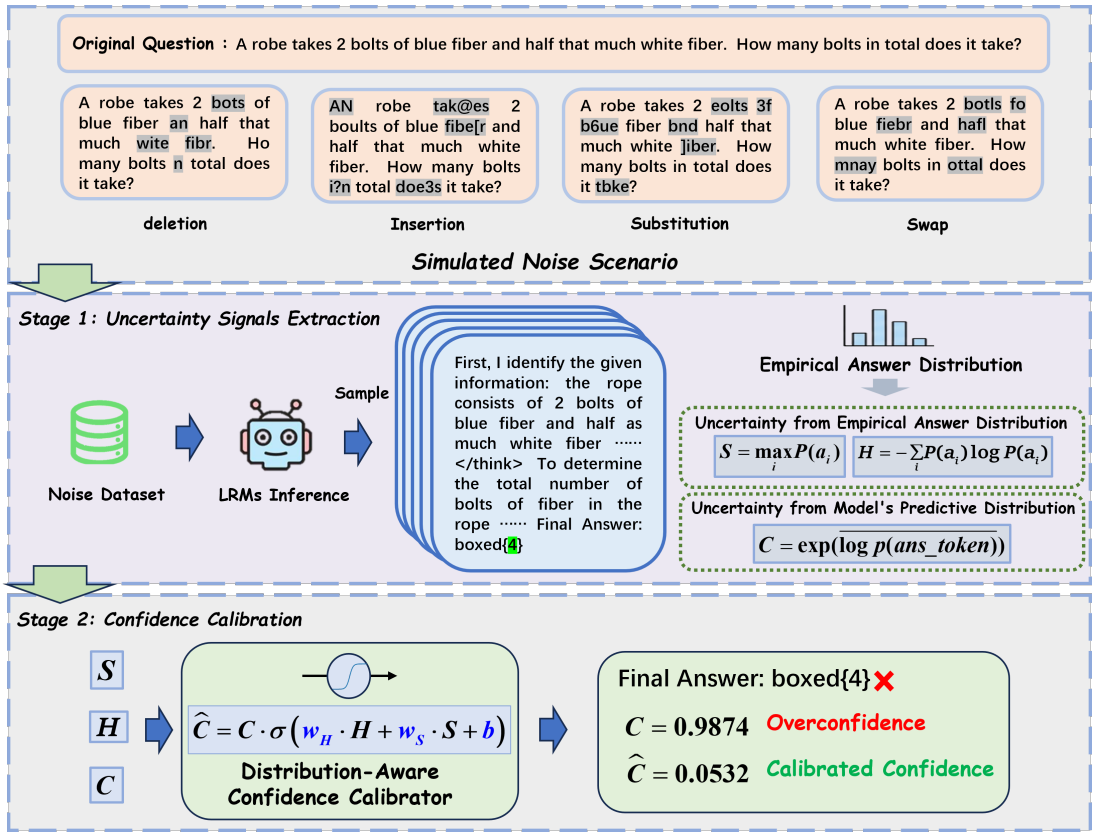

Figure 2: An overview of the proposed DisCal framework for calibrating confidence under noisy input conditions.

Higher stability indicates that the empirical distribution concentrates on a dominant answer, demonstrating consistent output patterns under noisy inputs. Using both entropy and stability is necessary because they quantify different distributional properties, enabling accurate characterization of the full spectrum of uncertainty patterns. **Detailed theoretical analysis and case study are provided in Appendix B.**

Uncertainty from Model's Predictive Distribution. We extract intrinsic confidence from the model's predictive distribution $P(\cdot \mid Q_i^p; \theta)$ at the token level:

$$C_i = \exp\left(\frac{1}{T_i}\sum_{t=1}^{T_i} \log P(a_t \mid Q_i^p, a_{<t}; \theta)\right), \quad (4)$$

where $a_t = a_{i,1}^p[t]$ denotes the t-th token of the final boxed answer, $a_{<t} = a_{i,1}^p[< t]$ represents all preceding tokens, and $T_i = |a_{i,1}^p|$ is the number of tokens in the boxed answer.

3.3 Distribution-Aware Confidence Calibrator

We train a distribution-based calibrator parameterized by a globally shared set of three learnable weights (w_H, w_S, b) to calibrate confidence by modulating the model's intrinsic confidence with distributional signals through multiplicative gating, where w_H and w_S control the sensitivity to distributional entropy and stability respectively, and b provides a baseline gating bias.

Multiplicative Gating Mechanism. Given distributional entropy H_i and stability S_i from the empirical answer distribution, the calibrator computes a gating factor via a linear transformation followed by sigmoid activation:

$$g_i = \sigma(w_H \cdot H_i + w_S \cdot S_i + b), \quad (5)$$

where $\sigma(\cdot)$ denotes the sigmoid function and (w_H, w_S, b) are learnable parameters. The calibrated confidence is obtained by scaling the intrinsic confidence C_i from the model's predictive distribution:

$$\hat{C}_i = C_i \cdot g_i. \quad (6)$$

This formulation enables adaptive confidence calibration. The optimization process inherently drives the learned weights to satisfy $w_H < 0$ and $w_S > 0$, reflecting the intended behavior where higher entropy suppresses confidence, while higher stability preserves it. When the empirical distribution is concentrated (high S_i, low H_i), the positive

contribution from $w_S \cdot S_i$ dominates, causing the gating factor to approach unity and thereby preserving intrinsic confidence. Conversely, when the distribution is dispersed (low S_i, high H_i), the negative contribution from $w_H \cdot H_i$ dominates, attenuating confidence to mitigate overconfidence.

Training Objective. The calibrator is trained on the GSM8K (Cobbe et al., 2021) training split, where perturbed inputs are generated using the strategy described in Section 3.1, and uncertainty signals are extracted via inference with DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025). Training is guided by a composite objective that imposes reliability constraints on calibrated confidence at multiple granularities, where $y_i = \mathbb{I}(a^p_{i,1} = A_i)$ indicates the correctness of the primary answer:

$$\begin{aligned}\mathcal{L} = &\frac{1}{N}\sum_{i=1}^{N}(\hat{C}_i - y_i)^2 \\ &+ \lambda_1 \frac{1}{|\mathcal{R}|}\sum_{(i,j)\in\mathcal{R}} \sigma(\hat{C}_i - \hat{C}_j) \\ &+ \lambda_2 \sum_{m=1}^{M} w_m \left(\bar{\hat{C}}_m - \bar{y}_m\right)^2,\end{aligned} \tag{7}$$

where N denotes the number of training instances and $\mathcal{R} = \{(i,j) \mid y_i = 0,\ y_j = 1\}$ denotes the set of all (incorrect, correct) prediction pairs; the loss encourages incorrect predictions to receive lower calibrated confidence than correct ones. The third term enforces bin-level reliability by partitioning predictions into M confidence bins, where $\bar{\hat{C}}_m$ and $\bar{y}_m$ denote the average calibrated confidence and empirical accuracy within bin m, respectively, and w_m is the proportion of samples assigned to that bin. The three terms respectively enforce instance-level confidence alignment, pairwise ordering consistency between correct and incorrect predictions, and bin-level reliability constraints. The calibrator learns only three parameters (w_H, w_S, b), enabling efficient optimization and robust generalization. Implementation details are provided in Appendix C.

4 Experiments

4.1 Experiment Settings

Datasets. We evaluate the proposed calibration method on three mathematical reasoning benchmark datasets with increasing difficulty: Easy, Medium, and Hard. **SVAMP** (Patel et al., 2021) (*Easy*) consists of elementary arithmetic problems and is used to examine calibration behavior under mild noisy input conditions. **GSM8K** (Cobbe et al., 2021) (*Medium*) contains multi-step arithmetic reasoning problems; we use its training split to train the DisCal calibrator and evaluate calibration performance on the test set. **AIME** (AI Mathematical Olympiad, 2025) (*Hard*) comprises competition-level problems with complex structures and long reasoning chains, serving to assess calibration robustness under highly challenging reasoning scenarios.

Baselines. We compare DisCal against a diverse set of baseline approaches for estimating model confidence, spanning token-level, entropy-based, semantic, and reasoning-based signals. ***Token Log-Probability*** **(Token)** computes confidence as the average log-probability of generated tokens. ***Mean Token Entropy*** **(MTE)** (Fadeeva et al., 2023) quantifies predictive uncertainty by averaging token-level entropy across the output sequence. ***Semantic Entropy*** **(SE)** (Kuhn et al., 2023) captures semantic variability by measuring meaning dispersion across multiple sampled answers. ***Self-consistency*** **(SC)** (Lyu et al., 2025) derives confidence from the majority-vote ratio among sampled reasoning trajectories. ***Self-critique*** (Zong et al., 2025) elicits a self-assessed confidence score through natural-language critique of the model's own answer. ***DisCal-base*** applies the gating function with heuristically initialized parameters ($w_H = -1$, $w_S = 1$, $b = 0$) without training, serving as an untrained variant for ablation. Implementation details are provided in Appendix D.

Models. We evaluate DisCal on six LLMs, including DeepSeek-R1-Distill-Qwen-7B/14B/32B (abbreviated as DeepSeek-7B/14B/32B) (Guo et al., 2025), Skywork-OR1-7B-Preview (Skywork-7B) (He et al., 2025), GLM-Z1-9B-0414 (GLM-Z1-9B) (Zeng et al., 2024), and Qwen2.5-32B-Instruct (Yang et al., 2025a).

Metrics. The following evaluation metrics are used for the calibration evaluation:

Area Under the Receiver Operating Characteristic Curve (AUROC) measures how well the calibrated confidence aligns with prediction correctness. In mathematical reasoning tasks, a higher AUROC indicates that the model, after calibration,

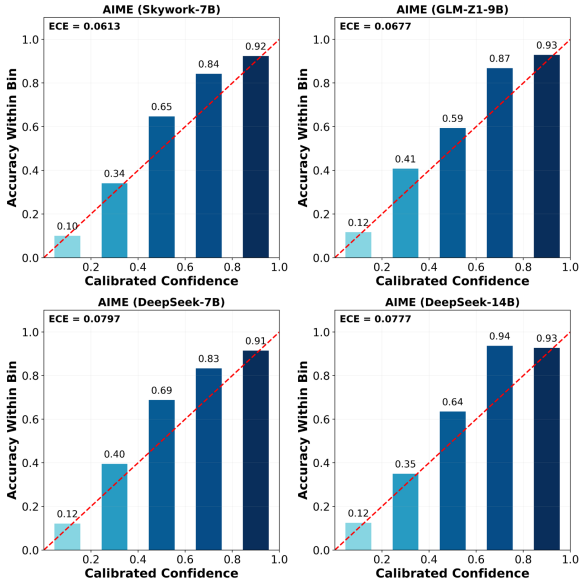

Figure 3: Reliability diagrams of four LRMs on the AIME dataset. Each plot compares predicted confidence against empirical accuracy across bins, demonstrating well-calibrated confidence estimates.

assigns higher confidence to correct answers and lower confidence to incorrect ones, reflecting the performance of the calibration method in improving the consistency between confidence and prediction accuracy. ***Expected Calibration Error (ECE)*** measures the difference between predicted confidence and empirical accuracy. The predictions are divided into 10 confidence bins, and ECE is computed as the weighted average of the absolute differences between the average confidence and the average accuracy in each bin. Lower ECE indicates better calibration.

4.2 Experimental Analysis and Findings

To evaluate the effectiveness of DisCal, we answer the following questions.

Q1: How effective is DisCal at calibrating confidence under noisy inputs?

As shown in Table 1, DisCal effectively calibrates confidence under noisy input across four LRMs and three datasets, demonstrating robust generalization across different model architectures and levels of reasoning difficulty. The results indicate that, under noisy inputs, sampling-based baselines (e.g., SC, SE) generally outperform token-level calibration strategies (Token, MTE). However, DisCal consistently attains the lowest ECE and the highest AUROC in most settings. For example, on the challenging AIME dataset, DisCal reduces the ECE of DeepSeek-7B by 39.21% compared to token-based confidence, and improves AUROC by 20.53%. This leads to a substantially tighter alignment between confidence scores and actual answer accuracy, significantly enhancing calibration reliability and discriminative capability under noisy input conditions.

As shown in the reliability diagrams (Figure 3), DisCal exhibits a well-calibrated structure on AIME, the most challenging dataset, under noisy input conditions simulated via character-level perturbations. Across all four LRMs, the confidence–accuracy curves closely follow the ideal diagonal, indicating that DisCal enables confidence scores to faithfully reflect prediction reliability. For instance, in DeepSeek-7B, the accuracy in the lowest confidence bin (0.0–0.2) ranges from 0.10 to 0.12; in the medium-confidence bins (0.4–0.6), accuracy increases to 0.35–0.69; and in the highest-confidence bin (0.8–1.0), accuracy stabilizes at 0.91–0.94. These patterns demonstrate a clear monotonic alignment between confidence and correctness. **Overall, DisCal produces stable and noise-resilient confidence estimates across diverse model architectures and challenging reasoning conditions, confirming the reliability of its calibration performance under noisy inputs.** Moreover, we extend our evaluation to large-scale models (Appendix E) and analyze calibration behavior across different noise levels and task difficulties (Appendix F), further validating the effectiveness and generalization of DisCal.

Q2: How do different uncertainty signals influence the calibration performance of DisCal?

We conduct an ablation study to assess the individual contribution of each uncertainty feature to DisCal's calibration performance. As shown in Table 2, the three signals contribute hierarchically. **Distributional Entropy** (H) is the most critical component. Removing it substantially increases ECE by 0.14–0.24 across datasets, indicating that entropy is essential for mitigating the strong overconfidence induced by noisy inputs. **Distributional Stability** (S) primarily refines fine-grained calibration. Its removal consistently degrades ECE by 0.09–0.13, while leaving AUROC largely unaffected, suggesting that stability improves confidence alignment without significantly impacting discriminative ranking. **Intrinsic Confidence** (C) provides additional gains, but DisCal remains effective without it. Notably, the $H + S$ configuration achieves reasonable calibration, highlighting DisCal's applicability in black-box settings where

Dataset	Method	Skywork-7B		GLM-Z1-9B		DeepSeek-7B		DeepSeek-14B	
		ECE ↓	AUROC ↑	ECE ↓	AUROC ↑	ECE ↓	AUROC ↑	ECE ↓	AUROC ↑
SVAMP (Easy)	Token	0.2465	0.6624	0.1628	0.4695	0.2667	0.6440	0.1991	0.6466
	SE	0.2390	0.5636	0.2289	0.5485	0.2264	0.5740	0.2221	0.5579
	MTE	0.1017	0.6382	0.0981	0.6073	0.1130	0.6539	0.1174	0.6533
	SC	0.0975	0.7621	0.1077	**0.6995**	0.0880	0.7583	0.1033	0.7643
	Self-Critique	0.1691	0.6522	0.0973	0.7003	0.2365	0.6091	0.2010	0.6421
	DisCal-base	0.1284	0.8104	0.1797	0.6420	0.1323	**0.8302**	0.1563	0.8046
	DisCal	**0.0515**	**0.8125**	**0.0737**	0.6803	**0.0524**	0.8247	**0.0644**	**0.8177**
GSM8K (Medium)	Token	0.3372	0.6044	0.2298	0.5187	0.3578	0.5769	0.2746	0.5876
	SE	0.3525	0.5722	0.3301	0.5457	0.3680	0.5706	0.3028	0.5626
	MTE	0.2441	0.6650	0.0648	0.6767	0.2488	0.6926	0.1971	0.6265
	SC	0.1494	0.7946	0.1328	**0.7170**	0.1412	0.8122	0.1538	0.7825
	Self-Critique	0.2872	0.6519	0.1594	0.7156	0.3163	0.6125	0.2790	0.6708
	DisCal-base	0.1206	0.8227	0.1059	0.7051	0.1245	0.8433	0.1276	0.8255
	DisCal	**0.0592**	**0.8240**	**0.0713**	0.7119	**0.0492**	**0.8486**	**0.1081**	**0.8310**
AIME (Hard)	Token	0.4450	0.6901	0.4393	0.5844	0.4718	0.6851	0.4559	0.6177
	SE	0.4704	0.6197	0.4214	0.6301	0.4843	0.6151	0.4929	0.6243
	MTE	0.2196	0.7262	0.2480	0.6491	0.2528	0.6924	0.3491	0.6787
	SC	0.3424	0.6822	0.3142	0.7049	0.2819	0.7439	0.2367	0.8058
	Self-Critique	0.1616	0.5999	0.2357	0.8175	0.2526	0.6785	0.4697	0.6185
	DisCal-base	0.1130	0.8994	0.1169	0.8866	0.1083	0.8883	0.1170	0.8796
	DisCal	**0.0613**	**0.9059**	**0.0679**	**0.8988**	**0.0797**	**0.8904**	**0.0800**	**0.8944**

Table 1: A comparison of confidence calibration performance under 30% perturbation intensity, evaluated using ECE and AUROC. The table presents the mean values, with the best results highlighted in **bold**. For a complete presentation of the results, including standard deviations, refer to Appendix I.

Dataset	Method	ECE ↓	AUROC ↑
GSM8K	DisCal	**0.0592**	**0.8240**
	w/o H	0.1947	0.7582
	w/o S	0.1861	0.8244
	w/o C	0.1169	0.8155
AIME	DisCal	**0.0613**	**0.9059**
	w/o H	0.3029	0.8260
	w/o S	0.1539	0.9071
	w/o C	0.1094	0.8979

Table 2: Ablation study of DisCal components on Skywork-7B. H: Distributional Entropy, S: Distributional Stability, C: Intrinsic Confidence.

token-level probabilities may be inaccessible.

The full DisCal configuration, incorporating all three signals, achieves optimal performance, with ECE values of 0.0592 and 0.0613, and AUROC scores of 0.8240 and 0.9059 on GSM8K and AIME, respectively. These results demonstrate that each uncertainty metric contributes meaningfully, and their integration yields the most reliable confidence calibration under noisy input conditions.

Q3: Is the distributional entropy sensitive to noise intensity?

As shown in Figure 4, the entropy distribution shifts significantly as noise strength increases from 10% to 30%. On GSM8K, mean entropy rises from 0.2438 to 0.4183; on AIME, from 0.7469 to 0.8336. A key observation is that entropy exhibits *region-dependent sensitivity* to noise variations:

(1) In the low-entropy region ($H \approx 0$), distributions under different noise levels largely overlap. These samples have concentrated answer distributions and require minimal calibration.

(2) In the high-entropy region ($H > 0.5$ for GSM8K; $H > 1.0$ for AIME), clear separation emerges, with higher noise producing elevated density. These are precisely the cases where predictions are more likely to be incorrect, and confidence suppression is most needed.

In summary, entropy is highly sensitive to noise, especially in uncertain regions where calibration is most critical.

Q4: Does DisCal capture prediction correctness through its uncertainty signals? Figure 5 shows the distribution of the uncertainty-conditioned gate g values for correct and incorrect

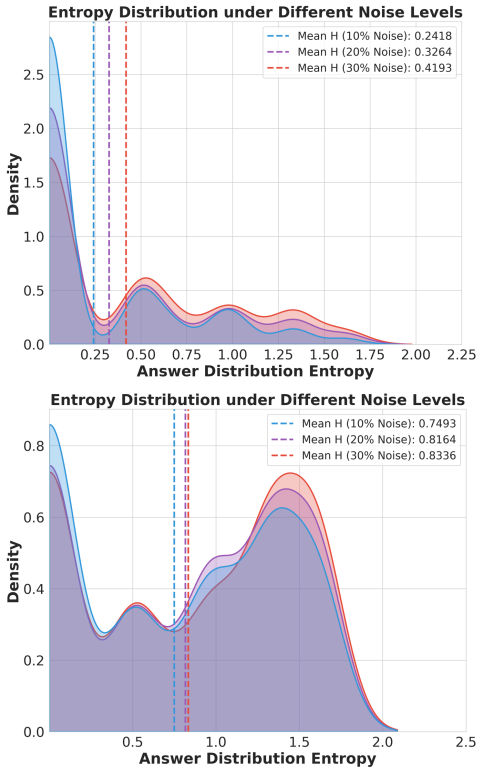

Figure 4: Entropy distribution under 10%, 20%, and 30% noise on GSM8K (top) and AIME (bottom). Higher noise shifts entropy rightward, especially in the high-entropy region.

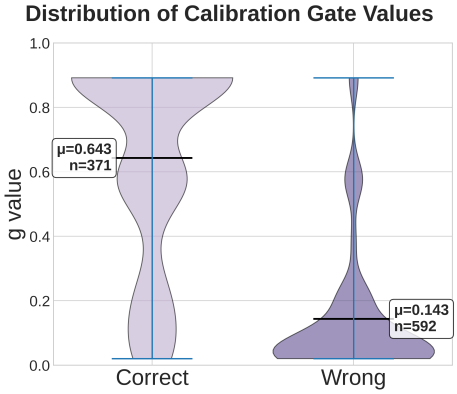

Figure 5: Distribution of the uncertainty-conditioned gate g learned by DisCal for correct and incorrect predictions on AIME, evaluated on DeepSeek-7B.

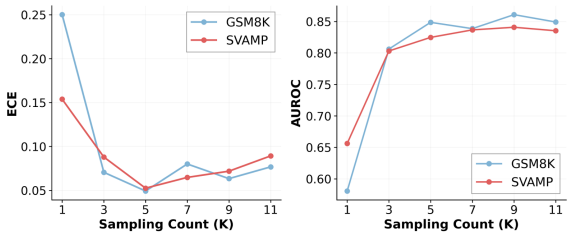

Figure 6: Effect of sampling count K under noisy input conditions on calibration and confidence discrimination.

predictions on the AIME benchmark, evaluated on DeepSeek-7B. The width of the plot indicates the density of predictions at each g value: wider sections represent higher frequency, while narrower sections indicate lower frequency. Correct predictions exhibit higher g values (mean = 0.643), indicating greater confidence and stability, while incorrect predictions exhibit lower g values (mean = 0.143), reflecting reduced confidence and instability. The minimal overlap between the two distributions demonstrates that g effectively distinguishes correct from incorrect predictions. Since calibrated confidence is computed as $\hat{C}_i = C_i \cdot g_i$, higher g values preserve confidence for correct predictions, while lower g values attenuate confidence for incorrect ones, thereby aligning confidence with actual accuracy. **DisCal effectively discriminates prediction correctness via its uncertainty signals, enabling well-calibrated confidence estimates.**

Q5: How does the sampling count K influence confidence calibration performance?

As shown in Figure 6, the impact of the sampling count K on calibration performance is illustrated. The most significant gains occur when increasing K from 1 to 5: ECE drops sharply, and AUROC rises consistently on both GSM8K and SVAMP on the validation set, indicating that only a small number of samples is sufficient for exposing uncertainty in the model's reasoning. Beyond $K = 5$, improvements plateau and occasionally decline, suggesting diminishing returns and the introduction of noise from lower-quality samples. **Based on this observation, we set $K = 5$ as the default configuration, as it achieves near-optimal calibration performance while maintaining low computational overhead.**

Q6: Does DisCal generalize to real-world noisy inputs? To provide an illustrative case study of DisCal beyond simulated noise, Table 3 presents a representative case drawn from real-world user-generated mathematical queries containing authentic typographical errors.

As shown in Table 3, despite containing multiple typographical errors (*"numbre"*, *"htan"*, *"eh"*, *"nubmers"*), the noisy input preserves the underlying mathematical semantics, yet triggers visible reasoning instability: two out of five independent sampling runs produce an incorrect answer of 100%, while the remaining three correctly compute 25%.

Field	Content
Clean Question	Jerry is rolling a six-sided die. How much more likely is it (expressed as a percentage) that he rolls a number greater than 3 than that he rolls two even numbers in a row?
Noisy Input	Jerry is rolling a six-sided die. How much more likely is it (expressed as a percentage) that he rolls a numbre greater than 3 htan that eh rolls two even nubmers in a row?
Gold Answer	25%
Primary Prediction	25% ✓
Sampled Answers	[25%, 25%, 100%, 100%, 25%]

Table 3: A real-world noisy input case study. Typographical errors are highlighted in red.

Although the primary prediction happens to be correct, the model assigns near-perfect intrinsic confidence of 0.9995, failing to reflect the latent uncertainty induced by the noisy input. DisCal detects this instability through elevated distributional entropy (0.6730) and reduced stability (0.60), suppressing the calibrated confidence from 0.9995 to 0.3852. This substantial reduction accurately reflects the true uncertainty: a model that produces the correct answer in only 60% of independent samples should not be assigned near-certain confidence. This case highlights a failure mode that token-level confidence measures fundamentally cannot capture: a model may appear highly confident while its reasoning is in fact fragile.

5 Conclusion

We identify a critical overconfidence problem in LRMs under character-level noise: models maintain high confidence despite sharp accuracy drops. This confidence-accuracy mismatch reveals an urgent need for calibration methods designed for noisy inputs. To address this challenge, we propose DisCal, a distribution-aware calibration framework that integrates uncertainty signals from both empirical answer distributions and model predictive distributions. DisCal learns only three parameters to calibrate confidence, enabling efficient training and robust generalization. Extensive experiments across multiple mathematical reasoning benchmarks demonstrate that DisCal substantially improves calibration quality, consistently outperforming existing methods under noisy input conditions.

Limitations

This work focuses on confidence calibration under character-level input noise (e.g., typographical errors), which is prevalent in real-world user inputs and can expose reasoning instability while preserving the core problem semantics. Although we include a representative real-world case in Section 4.2, our main experiments are conducted on simulated noise rather than large-scale authentic user-generated data, which we leave for future work.

Acknowledgements

We sincerely thank the Discipline and Technology Research Center for Large Model Intelligence Applications for their invaluable support throughout this research. The center provided essential computational resources, research infrastructure, and an intellectually stimulating environment that made this work possible.

References

AI Mathematical Olympiad. 2025. AIMO validation AIME dataset. https://huggingface.co/datasets/AI-MO/aimo-validation-aime. Licensed under Apache 2.0.

Amirhossein Aliakbarzadeh, Lucie Flek, and Akbar Karimi. 2025. Exploring robustness of multilingual llms on real-world noisy data. *arXiv preprint arXiv:2501.08322*.

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

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, and 1 others. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589.

Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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*.

Yuren Hao, Xiang Wan, and Chengxiang Zhai. 2025. An investigation of robustness of llms in mathematical reasoning: Benchmarking with mathematically-equivalent transformation of advanced mathematical problems. *arXiv preprint arXiv:2508.08833*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, and 1 others. 2025. Mathperturb: Benchmarking llms' math reasoning abilities against hard perturbations. *arXiv preprint arXiv:2502.06453*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Neeraja Kirtane, Yuvraj Khanna, and Peter Relan. 2025. Mathrobust-lv: Evaluation of large language models' robustness to linguistic variations in mathematical reasoning. *arXiv preprint arXiv:2510.06430*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. GSM-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984, Bangkok, Thailand. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2025. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19260–19268.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Yuhu Shang, Xiang Cheng, Yimeng Ren, Huijia Wu, Xuexiong Luo, Kangkang Lu, Jian Zhao, and Zhaofeng He. 2026. From chaos to cure: A prefix heuristics guided model-agnostic adaptive detoxification framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32902–32910.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Safal Shrestha, Minwu Kim, and Keith Ross. 2025. Mathematical reasoning in large language models: Assessing logical and arithmetic errors across wide numerical ranges. *arXiv preprint arXiv:2502.08680*.

Linwei Tao, Haolan Guo, Minjing Dong, and Chang Xu. 2024. Consistency calibration: Improving uncertainty calibration via consistency among perturbed neighbors. *arXiv preprint arXiv:2410.12295*.

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*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025a. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Yuli Yang, Hiroaki Yamada, and Takenobu Tokunaga. 2025b. Evaluating robustness of llms to numerical variations in mathematical reasoning. In *The Sixth*

Workshop on Insights from Negative Results in NLP, pages 171–180.

Tong Yu, Yongcheng Jing, Xikun Zhang, Wentao Jiang, Wenjie Wu, Yingjie Wang, Wenbin Hu, Bo Du, and Dacheng Tao. 2025a. Benchmarking reasoning robustness in large language models. *arXiv preprint arXiv:2503.04550*.

Yiyao Yu, Yuxiang Zhang, Dongdong Zhang, Xiao Liang, Hengyuan Zhang, Xingxing Zhang, Mahmoud Khademi, Hany Hassan Awadalla, Junjie Wang, Yujiu Yang, and 1 others. 2025b. Chain-of-reasoning: Towards unified mathematical reasoning in large language models via a multi-paradigm perspective. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24914–24937.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, and 36 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Bo Zhang, Cong Gao, Linkang Yang, Bingxu Han, Minghao Hu, Zhunchen Luo, Guotong Geng, Xiaoying Bai, Jun Zhang, Wen Yao, and 1 others. 2026. Global-local confidence fusion for hallucination detection in mathematical reasoning task. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34620–34628.

Jun Zhang, Wen Yao, Xiaoqian Chen, and Ling Feng. 2023. Transferable post-hoc calibration on pretrained transformers in noisy text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13940–13948.

Jun Zhang, Weien Zhou, Xianqi Chen, Wen Yao, and Lu Cao. 2019. Multisource selective transfer framework in multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(3):424–438.

Qing Zong, Jiayu Liu, Tianshi Zheng, Chunyang Li, Baixuan Xu, Haochen Shi, Weiqi Wang, Zhaowei Wang, Chunkit Chan, and Yangqiu Song. 2025. Critical: Can critique help llm uncertainty or confidence calibration? *arXiv preprint arXiv:2510.24505*.

Appendix

A Perturbation Design

To simulate realistic inference-time noise in a controlled manner, we implement a character-level perturbation module, where perturbations serve as a concrete mechanism for instantiating non-adversarial input noise, covering four operations: deletion, insertion, substitution, and swap. The module randomly selects a proportion of words and applies a single-character modification to each selected word, while a number-protection mechanism is incorporated to preserve the mathematical semantics of the original problem.

Specifically, deletion removes a random character from a word, insertion adds a noise character (e.g., letters or symbols) at a random position, substitution replaces a character with a visually similar or noisy alternative, and swap exchanges two adjacent characters to mimic realistic input noise such as keyboard slips and typographical errors.

To control perturbation strength and enable systematic analysis of confidence behavior under noisy inputs, each selected word undergoes only one character-level operation, and the perturbation ratio is set to 10%, 20%, or 30%, meaning that the corresponding proportion of words is randomly chosen for perturbation. Examples are shown in Table 4.

Type	Description	Example
Original	—	James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?
Deletion	characters removed while preserving meaning	**Jams dcides** to run 3 **sprnts** 3 times a week. He runs 60 meters **eac** sprint. How many total **meter** does he run a week?
Insertion	random inserted characters, simulating keyboard	James **debcides** to **krun** 3 sprints 3 times a week. He runs 60 meters **eacdh +sprint**. How**$** many **tohtal** meters does he run a week?
Substitution	characters replaced with visually similar or incorrect ones	**Ja3es** decides to run 3 sprints 3 times a week. **?e** runs 60 **mnters** each **s9rint**. **Qow** many total meters does he run **)** week?
Swap	adjacent characters swapped	**Jamse** decides to run 3 sprints 3 **timse** a week. **eH rusn** 60 meters **eahc** sprint. **Hwo** many total meters does he run a week?

Table 4: Representative examples of the four character-level perturbation types. Modified words are highlighted in bold color.

B Uncertainty Signal Analysis

This section provides a detailed analysis of the three uncertainty signals used in DisCal: Distributional Entropy (H), Distributional Stability (S), and Intrinsic Confidence (C). We clarify their distinct distributional sources and functional roles in confidence calibration under noisy inputs.

B.1 Distributional Sources of Uncertainty Signals

DisCal extracts uncertainty from two distributional sources:

Entropy H: Distributional Dispersion. Entropy quantifies the dispersion of the empirical answer distribution induced by K stochastic samples (Eq. 2). Higher entropy indicates greater dispersion in the empirical distribution, reflecting increased uncertainty induced by noisy inputs. This signal exhibits heightened sensitivity in high-uncertainty regions, which typically require stronger calibration interventions.

Stability S: Distributional Concentration. Stability measures the concentration of the empirical answer distribution (Eq. 3). Higher stability indicates that the empirical distribution concentrates on a dominant answer, demonstrating consistent output patterns under noisy inputs.

Intrinsic Confidence C: Internal Distributional Certainty. Unlike H and S, which characterize the empirical answer distribution at the answer level, intrinsic confidence is extracted from the model's predictive distribution at the token level (Eq. 4), where a_t denotes the t-th token of the boxed answer $a^p_{i,1}$, $a_{<t}$ represents all preceding tokens, and $T_i = |a^p_{i,1}|$ is the number of tokens in the boxed answer. C_i represents the geometric mean of token-level likelihoods from the predictive distribution $P(\cdot \mid Q^p_i; \theta)$, capturing the model's internal distributional certainty. This signal provides information about the model's internal belief that is not reflected in the empirical answer distribution.

B.2 Case Study: Empirical Distribution under Reasoning Instability

To illustrate how the empirical answer distribution reflects reasoning instability, we examine a representative case in Table 6. The example shows how divergent reasoning trajectories manifest as a dispersed empirical distribution.

From Reasoning Instability to Distributional Dispersion. When reasoning processes are stable, independent stochastic samples tend to produce a concentrated empirical distribution with a dominant answer. Conversely, unstable reasoning leads to a dispersed empirical distribution with multiple competing answers. This motivates the use of distributional properties (entropy and stability) as model-agnostic proxies for reasoning reliability.

Calibration in Action. In Table 6, the model assigns high intrinsic confidence from its predictive distribution ($C = 0.95$), while the empirical answer distribution is highly dispersed, producing four different answers (\$65,000, \$-10,000, \$70,000, \$195,000). This dispersion is captured by high entropy ($H = 1.33$) and low stability ($S = 0.40$) of the empirical distribution. Conditioned on these distributional signals, DisCal suppresses confidence to 0.06, effectively correcting

Signal	Distributional Source	Level	Quantifies
H	Empirical answer distribution	Answer-level	Distributional dispersion
S	Empirical answer distribution	Answer-level	Distributional concentration
C	Model's predictive distribution	Token-level	Internal distributional certainty

Table 5: Distributional sources and functional roles of the three uncertainty signals in DisCal.

the mismatch between predictive and empirical distributions.

B.3 Information-Theoretic Analysis: Non-Redundancy of Distributional Signals

Definition 1 (Statistical Redundancy)**.** *Two statistics $T_1(\pi)$ and $T_2(\pi)$ are **redundant** if and only if there exists a bijective function f such that $T_1 = f(T_2)$, i.e., knowing one uniquely determines the other. Conversely, T_1 and T_2 are **non-redundant** if neither uniquely determines the other.*

Complete Enumeration under $K = 5$. When $K = 5$, the empirical answer distribution π is constrained to exactly 7 distinct patterns, corresponding to the integer partitions of 5. Table 7 provides the complete enumeration:

Theorem 1 (Non-Redundancy of H and S)**.** *When $K = 5$, entropy H and stability S are non-redundant distributional statistics: S does not uniquely determine H.*

We identify counterexamples from Table 7:

Case 1 ($S = 0.6$):

$$\pi_3 = (0.6, 0.4) \Rightarrow S = 0.6, \quad H = 0.67 \tag{8}$$

$$\pi_4 = (0.6, 0.2, 0.2) \Rightarrow S = 0.6, \quad H = 0.95 \tag{9}$$

Case 2 ($S = 0.4$):

$$\pi_5 = (0.4, 0.4, 0.2) \Rightarrow S = 0.4, \quad H = 1.05 \tag{10}$$

$$\pi_6 = (0.4, 0.2, 0.2, 0.2) \Rightarrow S = 0.4, \quad H = 1.33 \tag{11}$$

In both cases, $S(\pi_i) = S(\pi_j)$ but $H(\pi_i) \neq H(\pi_j)$. Therefore, the mapping $S \mapsto H$ is not injective, and H carries information about the empirical distribution π that S alone cannot capture.

Theorem 2 (Positive Conditional Mutual Information)**.** *Let Π be a random variable over the 7 partitions. For any non-degenerate prior $P(\Pi)$:*

$$I(\Pi; H \mid S) > 0 \tag{12}$$

Consider the conditional distribution $P(\Pi \mid S)$:

- $S = 0.6$: $\Pi \in \{\pi_3, \pi_4\}$ $\Rightarrow H(\Pi \mid S = 0.6) > 0$
- $S = 0.4$: $\Pi \in \{\pi_5, \pi_6\}$ $\Rightarrow H(\Pi \mid S = 0.4) > 0$
- $S \in \{1.0, 0.8, 0.2\}$: Π uniquely determined $\Rightarrow H(\Pi \mid S) = 0$

For any prior with $P(\pi_3), P(\pi_4), P(\pi_5), P(\pi_6) > 0$:

$$H(\Pi \mid S) = \sum_s P(S = s) \cdot H(\Pi \mid S = s) > 0 \tag{13}$$

Given both S and H, the partition Π is uniquely determined, so $H(\Pi \mid S, H) = 0$. Therefore:

$$\begin{aligned} I(\Pi; H \mid S) &= H(\Pi \mid S) - H(\Pi \mid S, H) \\ &= H(\Pi \mid S) > 0 \end{aligned} \tag{14}$$

Implications for Prediction Correctness. Let $Y \in \{0, 1\}$ denote prediction correctness. By the chain rule:

$$I(Y; S, H) = I(Y; S) + I(Y; H \mid S) \tag{15}$$

Since H provides information about the empirical distribution Π beyond S (Theorem 2), and Y depends on Π, we have $I(Y; H \mid S) > 0$ unless Y is fully determined by S alone. The ablation study (Table 2 in the main text) empirically validates this: removing H degrades both ECE and AUROC, while removing S primarily degrades ECE, confirming that both distributional signals carry non-redundant information about prediction correctness.

B.4 Why Both Distributional Signals Are Necessary

While Theorems 1 and 2 establish that H and S are information-theoretically non-redundant, this section demonstrates through concrete examples why using either signal alone leads to systematic miscalibration, making both necessary for effective uncertainty quantification.

Perturbed (Noisy) Question	Sampled Reasoning Paths
Jsh decides to try flipping a house. He buys a house for $80,000 and spends $50,000 on repairs. This increases the value of the house by 150%. How much profit did he make?	**Sample 1 (Prediction: $65,000)** Total investment is $80,000 + $50,000 = $130,000. The house value increases by 150% of the total investment: $1.5 \times 130{,}000 = 195{,}000$. Profit is $195{,}000 - 130{,}000 = 65{,}000$.
	Sample 2 (Prediction: $-10,000) The value increases by 150% of the original purchase price: $1.5 \times 80{,}000 = 120{,}000$. Comparing with the total cost 130,000, this results in a loss of $10,000.
	Sample 3 (Prediction: $70,000) The repair increases the value by 150% of the original price: $1.5 \times 80{,}000 = 120{,}000$. The new value is $80{,}000 + 120{,}000 = 200{,}000$. Subtracting total cost gives a profit of $70,000.
	Sample 4 (Prediction: $70,000) The house value becomes 250% of the original price after a 150% increase, resulting in $200,000. Subtracting the total investment of $130,000 yields a profit of $70,000.
	Sample 5 (Prediction: $195,000) The value increase is computed as 150% of the total investment: $1.5 \times 130{,}000 = 195{,}000$. Adding this to the investment gives $325,000, leading to a reported profit of $195,000.

Table 6: Multiple sampled reasoning trajectories for the same noisy input. Divergent reasoning leads to a dispersed empirical answer distribution with high entropy ($H = 1.33$) and low stability ($S = 0.40$). DisCal detects the mismatch between the high predictive distribution confidence ($C = 0.95$) and low empirical distribution stability, suppressing confidence to $\hat{C} = 0.06$.

ID	Partition	Distribution π	S	H (nats)
1	$[5]$	(1.0)	1.0	0
2	$[4, 1]$	$(0.8, 0.2)$	0.8	0.50
3	$[3, 2]$	$(0.6, 0.4)$	0.6	0.67
4	$[3, 1, 1]$	$(0.6, 0.2, 0.2)$	0.6	0.95
5	$[2, 2, 1]$	$(0.4, 0.4, 0.2)$	0.4	1.05
6	$[2, 1, 1, 1]$	$(0.4, 0.2, 0.2, 0.2)$	0.4	1.33
7	$[1, 1, 1, 1, 1]$	$(0.2, 0.2, 0.2, 0.2, 0.2)$	0.2	1.61

Table 7: Complete enumeration of all possible empirical answer distributions under $K = 5$.

Limitation of Using Stability Alone. Consider two distributions with identical stability but vastly different uncertainty characteristics:

Distribution A: $\pi_3 = (0.6, 0.4)$

- Interpretation: 3 out of 5 samples predict answer A, 2 predict answer B
- Metrics: $S = 0.6$, $H = 0.67$
- Uncertainty pattern: Two competing alternatives in clear competition

Distribution B: $\pi_4 = (0.6, 0.2, 0.2)$

- Interpretation: 3 out of 5 samples predict answer A, 1 predicts B, 1 predicts C
- Metrics: $S = 0.6$, $H = 0.95$
- Uncertainty pattern: Multiple competing alternatives with higher dispersion

Both distributions exhibit identical convergence strength ($S = 0.6$), yet their entropy differs by 42%. A calibrator relying solely on stability would treat these identically, failing to recognize that Distribution B, which contains three distinct answers, indicates more severe reasoning instability requiring stronger confidence suppression. The entropy difference ($H = 0.95$ vs 0.67) captures this critical distinction: while both have 60% convergence to a dominant answer, Distribution B's remaining probability mass is more dispersed across multiple alternatives.

Similarly, for $S = 0.4$:

Distribution C: $\pi_5 = (0.4, 0.4, 0.2)$ with $H = 1.05$

Distribution D: $\pi_6 = (0.4, 0.2, 0.2, 0.2)$ with $H = 1.33$

Both show weak convergence ($S = 0.4$), but Dis-

tribution C represents a clear bimodal competition (two equally strong alternatives), while Distribution D indicates more chaotic reasoning with four distinct answers. Using stability alone cannot distinguish these fundamentally different uncertainty patterns.

Limitation of Using Entropy Alone. Conversely, relying solely on entropy fails to capture the strength of convergence. Consider:

Distribution E: $\pi_2 = (0.8, 0.2)$

- Metrics: $S = 0.8$, $H = 0.50$
- Interpretation: Strong convergence (80%) with a single outlier

Distribution F: $\pi_3 = (0.6, 0.4)$

- Metrics: $S = 0.6$, $H = 0.67$
- Interpretation: Moderate convergence (60%) with substantial competition

While both have relatively low entropy, Distribution E demonstrates much stronger convergence ($S = 0.8$ vs 0.6). A calibrator using only entropy might apply similar suppression to both, incorrectly penalizing Distribution E despite its 80% agreement on a dominant answer. The stability difference reveals that Distribution E's prediction is more reliable, warranting less aggressive confidence adjustment.

Joint Necessity. These examples demonstrate that entropy and stability quantify fundamentally different aspects of distributional uncertainty:

- **Entropy** measures global dispersion across all answers, capturing whether the probability mass is concentrated or spread among multiple alternatives.
- **Stability** measures the mass of the dominant answer, capturing convergence strength regardless of how the remaining probability is distributed.

A prediction can exhibit strong convergence to a dominant answer yet still have high distributional uncertainty from competing alternatives (e.g., π_4 with $S = 0.6, H = 0.95$), or conversely, show weak convergence despite relatively low overall dispersion. Only by integrating both signals can DisCal accurately characterize the full spectrum of uncertainty patterns arising from noisy inputs, enabling precise calibration decisions that reflect both convergence strength and distributional shape.

B.5 Role of Intrinsic Confidence

Signals H and S are derived from the empirical answer distribution across K samples, capturing *behavioral* uncertainty at the answer level. In contrast, C is extracted from the model's predictive distribution $P(\cdot \mid Q; \theta)$ at the token level, capturing *internal distributional certainty*.

This distributional distinction is critical for detecting scenarios where the empirical distribution is concentrated but the predictive distribution assigns low probability. In such cases:

- $S_i = 1.0$ (concentrated empirical distribution)
- $H_i = 0$ (no distributional dispersion)

Both empirical distributional signals suggest that high confidence should be preserved. However, if the predictive distribution assigns a low likelihood to the generated tokens, which indicates low internal distributional certainty despite consistent outputs, then C_i will be low. This enables DisCal to integrate evidence from both distributional sources: the empirical answer distribution and the model's predictive distribution.

C Training Details

C.1 Training Setup

The DisCal calibrator is trained on the GSM8K training split. For each instance, we generate a single perturbed input Q_i^p following the perturbation strategy described in Section 3.1. The `DeepSeek-R1-Distill-Qwen-7B` model is sampled $K = 5$ times on the same perturbed input under a fixed stochastic decoding strategy.

From the resulting prediction distribution, we compute three key metrics:

- **Distributional entropy** H_i – measuring noise-induced distributional uncertainty.
- **Distributional stability** S_i – measuring consistency across samples.
- **Intrinsic confidence** C_i – defined as the length-normalized likelihood of the boxed final answer.

Ground-truth correctness labels $y_i \in \{0, 1\}$ are obtained from the official GSM8K solutions.

C.2 Calibration Function

DisCal employs a multiplicative gating mechanism to modulate the intrinsic confidence based on un-

certainty signals:

$$\hat{C}_i = C_i \cdot \sigma(w_H \cdot H_i + w_S \cdot S_i + b)\,, \quad (16)$$

where $w_H, w_S \in \mathbb{R}$ and $b \in \mathbb{R}$ are trainable scalar parameters, and $\sigma(\cdot)$ denotes the sigmoid function. The gate function

$$g_i = \sigma(w_H \cdot H_i + w_S \cdot S_i + b) \quad (17)$$

serves as a learned weighting factor that adjusts confidence based on the model's uncertainty characteristics. Critically, the base model parameters remain frozen during training, ensuring we only learn the calibration mapping.

C.3 Learned Parameters and Interpretation

Parameter Semantics. The negative sign of w_H confirms that increased distributional entropy (indicating reasoning instability) should suppress confidence. The positive sign of w_S indicates that higher answer consistency should preserve confidence. The ratio $|w_H|/|w_S| \approx 2.57$ quantifies the relative importance of entropy versus stability, suggesting entropy is approximately 2.6× more influential in calibration decisions.

Cross-Setting Generalization. We train the calibrator *only once* on GSM8K with 30% perturbation, yet the same parameters achieve strong calibration across:

- Different datasets: SVAMP, GSM8K, AIME
- Different noise levels: 10%, 20%, 30%
- Different noise types: deletion, insertion, substitution, swap

This generalization suggests that the learned parameters capture *universal* uncertainty patterns rather than dataset-specific artifacts. We attribute this to: (1) the normalized nature of H, S, and C; (2) the multiplicative gating mechanism that operates on relative scales; and (3) the minimal parameter count (only 3) that prevents overfitting.

Parameter Stability. Across 5 random seeds, the learned parameters exhibit low variance: $w_H = -3.11 \pm 0.08$, $w_S = 1.21 \pm 0.05$, $b = 0.90 \pm 0.03$, confirming training stability.

C.4 Training Objective

The training objective enforces three reliability constraints at different granularities:

$$\mathcal{L} = \mathcal{L}_{\text{inst}}(\hat{C}, y) + \lambda_1 \mathcal{L}_{\text{rank}}(\hat{C}, y) + \lambda_2 \mathcal{L}_{\text{bin}}(\hat{C}, y), \quad (18)$$

where each component targets a different aspect of calibration quality.

Instance-Level Loss. $\mathcal{L}_{\text{inst}}$ is implemented as mean squared error:

$$\mathcal{L}_{\text{inst}}(\hat{C}, y) = \frac{1}{N} \sum_{i=1}^{N} (\hat{C}_i - y_i)^2, \quad (19)$$

providing direct instance-level supervision. Due to the multiplicative structure in Eq. 16, the gradient flows through the gate:

$$\frac{\partial \mathcal{L}_{\text{inst}}}{\partial w_H} = \frac{2}{N} \sum_{i=1}^{N} (\hat{C}_i - y_i) \cdot C_i \cdot g_i(1 - g_i) \cdot H_i, \quad (20)$$

where the term $g_i(1 - g_i)$ discourages extreme gate values (near 0 or 1), ensuring the gate acts as a smooth modulator rather than a hard threshold.

Ranking Loss. $\mathcal{L}_{\text{rank}}$ enforces correct ordering between incorrect ($y_i = 0$) and correct ($y_j = 1$) predictions:

$$\mathcal{L}_{rank}(\hat{C}, y) = \frac{1}{|\mathcal{R}|} \sum_{(i,j) \in \mathcal{R}} \sigma(\hat{C}_i - \hat{C}_j), \quad (21)$$

where $\mathcal{R} = \{(i, j) : y_i = 0, y_j = 1\}$ is the set of all incorrect-correct pairs. This provides smooth gradients that penalize confidence inversions, improving discrimination between correct and incorrect predictions.

Binning Loss. $\mathcal{L}_{\text{bin}}$ minimizes the discrepancy between average confidence and empirical accuracy within M confidence bins:

$$\mathcal{L}_{\text{bin}}(\hat{C}, y) = \sum_{m=1}^{M} w_m \left[\left(\frac{1}{n_m} \sum_{i \in B_m} \hat{C}_i \right) - \frac{1}{n_m} \sum_{i \in B_m} y_i \right]^2 \quad (22)$$

where B_m denotes the m-th bin, $n_m = |B_m|$ is the number of samples in that bin, and $w_m = n_m/N$ weights each bin by its population. We use $M = 10$ equally-spaced bins. This term provides the dominant gradient signal for bin-level calibration quality (ECE).

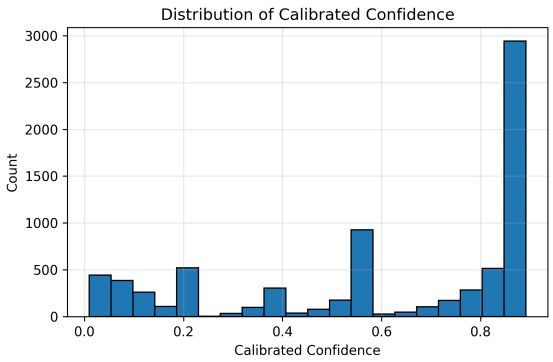

Figure 7: Distribution of calibrated confidence on GSM8K under character-level noise perturbations. The bimodal distribution shows strong concentration at high and low confidence extremes, with a continuous spectrum across intermediate ranges. This pattern demonstrates well-calibrated continuous confidence that accurately reflects prediction reliability, as evidenced by strong calibration metrics.

C.5 Calibrated Confidence Distribution

Figure 7 shows the distribution of calibrated confidence on the GSM8K dataset under character-level noise perturbations. The distribution exhibits a distinctive bimodal pattern:

- **High-confidence region**: A substantial portion of samples concentrates in the highest confidence interval, with correspondingly high empirical accuracy, indicating reliable high-confidence predictions.
- **Low-confidence region**: A notable fraction of samples fall in the lowest confidence range, with proportionally low accuracy, demonstrating the model's ability to identify uncertain predictions.
- **Intermediate region**: The remaining samples are distributed across middle confidence ranges, forming a continuous confidence spectrum.

This concentration at confidence extremes reflects actual prediction reliability, as validated by strong alignment between confidence and accuracy, along with excellent overall calibration metrics.

C.6 Optimization Details

We set hyperparameters $\lambda_1 = 2$ and $\lambda_2 = 5$ based on validation performance. Grid search over $\lambda_2 \in \{1, 5, 10\}$ revealed that moderate values provide the best balance between calibration accuracy and discrimination capability.

Optimization is performed using the Adam optimizer with learning rate $\eta = 0.01$. We employ full-batch gradient updates for 300 epochs with logit clipping to $[-20, 20]$ for numerical stability. All loss components decrease cooperatively during training, with $\mathcal{L}_{\text{bin}}$ typically converging fastest. The lightweight nature of the calibrator enables training to be completed within seconds on a single GPU.

Calibration performance is evaluated using the Expected Calibration Error (ECE) and Area Under the ROC Curve (AUROC), computed on held-out test sets both before and after applying DisCal.

D Calibration Methods Implementation

This section provides a detailed implementation of the various confidence calibration methods discussed in the paper. Each algorithm below describes the steps for each method, from model prediction to confidence calibration and evaluation.

D.1 Distribution-Aware Calibration (Our Method)

This method adjusts model confidence using uncertainty signals, including answer distributional entropy, distributional stability, and intrinsic confidence, to enhance calibration under noisy inputs. For a detailed breakdown, refer to Algorithm 1.

D.2 Implementation of Semantic Entropy

This method leverages the entropy of $K = 5$ sampled model predictions to quantify uncertainty, calibrating confidence according to the variability in the predicted answers. See Algorithm 2 for the full implementation.

D.3 Implementation of Self-Consistency

Confidence is calibrated by evaluating the consistency of multiple predictions sampled $K = 5$ times, with higher consistency indicating greater reliability and confidence. For the full algorithm, refer to Algorithm 3.

D.4 Implementation of Mean Token Entropy

Token-level entropy is calculated for each prediction, and confidence is derived from the average entropy. See Algorithm 4 for the detailed steps.

D.5 Implementation of Self-Critique

In this method, the model performs self-critique on its predictions, adjusting its confidence based on the revised evaluation of its own reasoning. The

prompt is provided in Table 8. Refer to Algorithm 5 for the detailed algorithm.

Self-Critique Prompt Template

You are an AI assistant that evaluates your own reasoning.

Question: `{question}`

Your initial answer: `{answer}`

Now:
1. Re-examine the reasoning.
2. Identify potential issues.
3. Decide if your answer is likely correct.

Respond exactly in this format:

Final Answer: `{answer}`
Updated Confidence: <a number between 0-100>

Table 8: Prompt template for Self-Critique baseline. The model first generates an initial answer, then receives this prompt to perform self-evaluation. Confidence scores are normalized to [0,1].

E Large-Scale Model Results

As shown in Table 9, we extend our evaluation to large-parameter reasoning models and instruction-tuned models to further validate the effectiveness of DisCal. Specifically, we report calibration performance on the challenging AIME benchmark, using ECE and AUROC as evaluation metrics. All results are reported relative to intrinsic confidence C, highlighting the absolute calibration gains brought by DisCal.

Model	Method	ECE ↓	AUROC ↑
Qwen2.5-32B-it	Token	0.7711	0.6664
	DisCal-base	0.1759	0.8612
	DisCal	**0.0633**	**0.8629**
DeepSeek-32B	Token	0.4226	0.6711
	DisCal-base	0.1299	0.8970
	DisCal	**0.0662**	**0.9024**

Table 9: Calibration performance on AIME for large-scale instruction-tuned models.

Model	ECE ↓	AUROC ↑
DeepSeek-7B	0.3973 → **0.0831**	0.5683 → **0.8115**
DeepSeek-14B	0.3428 → **0.1123**	0.5856 → **0.8257**
GLM-Z1-9B	0.2775 → **0.0798**	0.5293 → **0.7278**

Table 10: Calibration performance under mixed noise types.

F Confidence Adjustment Analysis

We analyze whether DisCal provides effective calibration across different task difficulties and noise levels. We evaluate on three datasets of increasing difficulty: SVAMP (Easy), GSM8K (Medium), and AIME (Hard). For each dataset, we apply character-level perturbations at three noise strengths: 10%, 20%, and 30%.

As shown in Figure 8, ΔC denotes the average confidence reduction after calibration. The results demonstrate that **DisCal achieves effective calibration across all noise levels and task difficulties**:

(1) Consistent calibration across noise levels. For each dataset, DisCal successfully adjusts confidence at all three perturbation strengths (10%, 20%, 30%), with adjustment magnitude generally increasing with noise strength on Easy and Medium tasks. This confirms that DisCal is not tuned to a specific noise level but generalizes across varying perturbation intensities.

(2) Adaptive adjustment to task difficulty. Harder tasks receive stronger adjustments: AIME (hard) exhibits $\Delta C = 0.48$–0.54, GSM8K (medium) shows $\Delta C = 0.22$–0.36, and SVAMP (easy) requires only $\Delta C = 0.19$–0.27. This adaptive behavior indicates that DisCal applies calibration proportional to the model's inherent overconfidence on each task.

(3) Generalization across model scales. The trends are consistent across both Skywork-7B and DeepSeek-14B, demonstrating that the learned calibrator transfers well to different model architectures and scales.

G Mixed Noise Calibration

The mixed setting represents a more rigorous and realistic noisy input condition, where four different types of character-level errors simultaneously appear in the same input question. Specifically, each selected word is randomly perturbed by one of four operators (deletion, insertion, substitution, or swap), simulating various noise sources such as typographical errors and keyboard slips. By fixing the noise level at 30% (i.e., perturbing 30% of the words through character-level operations), we create a challenging scenario that places higher demands on the model's reasoning stability and confidence reliability.

We evaluate this mixed noisy input condition using DeepSeek-7B, DeepSeek-14B, and GLM-

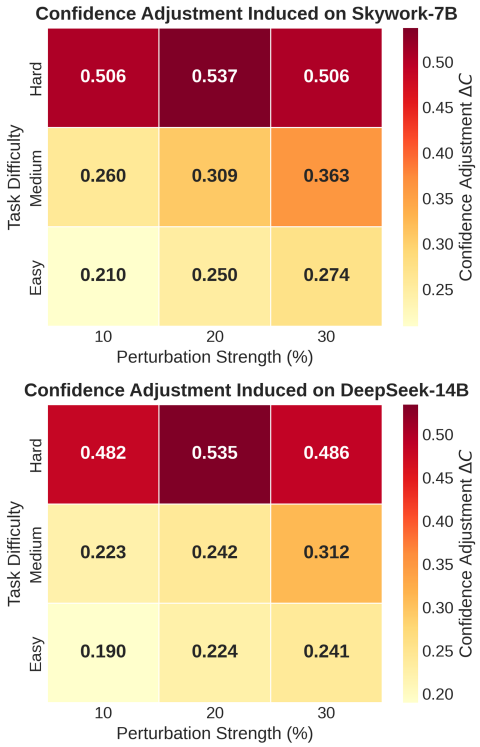

Figure 8: Confidence adjustment ΔC induced by DisCal across three datasets (SVAMP/Easy, GSM8K/Medium, AIME/Hard) and three noise levels (10%, 20%, 30%), evaluated on Skywork-7B (top) and DeepSeek-14B (bottom).

Z1-9B. DisCal is applied as a confidence calibrator to refine the models' *intrinsic confidence estimates* produced during inference, without modifying the underlying model parameters or predictions.

As summarized in Table 10, baseline confidence estimates across all models exhibit substantial miscalibration under mixed noise, with ECE values ranging from 0.28 to 0.40 and AUROC scores close to 0.5, indicating that confidence scores become largely non-informative for discriminating between correct and incorrect predictions. In contrast, DisCal consistently restores confidence quality across all evaluated models, achieving a 65%–80% reduction in ECE and an AUROC improvement of 0.20–0.24. These results demonstrate that DisCal remains robust and effective even under severe, heterogeneous noisy input conditions, highlighting its ability to handle complex and co-occurring character-level noise in realistic reasoning scenarios.

H Noise Subtype Results

As shown in Table 11, we further analyze the calibration behavior of DisCal across different noise subtypes induced by character-level perturbations. Specifically, we construct four independent subtype datasets corresponding to *deletion*, *insertion*, *substitution*, and *swap*, each simulating a distinct form of character-level input noise under a fixed noise level of 30%. This analysis aims to examine whether the calibration improvements provided by DisCal remain consistent across heterogeneous noise patterns and varying task difficulties.

I Baseline Experiment Results

Table 12 provides the complete baseline experiment results with mean and standard deviation over 5 trials, corresponding to Table 1 in the main text.

Dataset	Subtype	ECE ↓	AUROC ↑
AIME	deletion	0.2753 → **0.1035**	0.7017 → **0.9145**
	insertion	0.4628 → **0.0661**	0.6628 → **0.8441**
	substitution	0.5239 → **0.0616**	0.6908 → **0.8628**
	swap	0.2199 → **0.1626**	0.6428 → **0.8971**
GSM8K	deletion	0.1805 → **0.1178**	0.5953 → **0.8874**
	insertion	0.4899 → **0.1337**	0.6936 → **0.7988**
	substitution	0.5566 → **0.0958**	0.6611 → **0.9093**
	swap	0.1491 → **0.1257**	0.5941 → **0.8726**
SVAMP	deletion	0.1384 → **0.1063**	0.6708 → **0.8840**
	insertion	0.3160 → **0.0727**	0.6527 → **0.7656**
	substitution	0.3834 → **0.1040**	0.6521 → **0.8971**
	swap	0.1161 → **0.1144**	0.6616 → **0.9014**

Table 11: Subtype-wise calibration results under 30% perturbations. For each metric, values are reported as *intrinsic confidence* → *DisCal-calibrated confidence*, indicating performance before and after applying DisCal.

Algorithm 1 DisCal: Distribution-aware Confidence Calibration

Require: Dataset $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^{N}$, Model $\mathcal{M}$, Calibrator $\varphi = (w_H, w_S, b)$, $K = 5$

Ensure: Calibrated confidence $\{\hat{C}_i\}_{i=1}^{N}$ and metrics

for $(Q_i, A_i) \in \mathcal{D}$ **do**

  Phase 1: Multi-Sample Prediction

  Sample K responses: $\{a_{i,j}^p\}_{j=1}^{K} \sim \mathcal{M}(Q_i^p)$

  $\mathcal{A}_i^p \leftarrow \text{Unique}(\{a_{i,j}^p\})$ ▷ Distinct answers

  Phase 2: Uncertainty Signal Computation

  Compute answer distribution: $\pi_i(z) = \frac{1}{K}\sum_{j=1}^{K} \mathbb{I}(a_{i,j}^p = z), \quad \forall z \in \mathcal{A}_i^p$

  Compute Entropy: $H_i \leftarrow -\sum_{z \in \mathcal{A}_i^p} \pi_i(z) \log \pi_i(z)$

  Compute Stability: $S_i \leftarrow \max_{z \in \mathcal{A}_i^p} \pi_i(z)$

  Compute Intrinsic Confidence: $C_i \leftarrow \exp\Big(\frac{1}{T_i}\sum_{t=1}^{T_i} \log P(a_t \mid Q_i^p, a_{<t}; \theta)\Big)$ where $a_t = a_{i,1}^p[t]$, $a_{<t} = a_{i,1}^p[< t]$, and $T_i = |a_{i,1}^p|$

  Phase 3: Confidence Calibration

  Compute Gating factor: $g_i \leftarrow \sigma(w_H \cdot H_i + w_S \cdot S_i + b)$

  Calibrate confidence: $\hat{C}_i \leftarrow C_i \cdot g_i$

  Compute Correctness: $y_i \leftarrow \mathbb{I}(a_{i,1}^p = A_i)$

end for

Compute AUROC:

AUROC $\leftarrow f_{\text{AUC}}(\{\hat{C}_i\}, \{y_i\})$

Compute ECE:

ECE $\leftarrow f_{\text{ECE}}(\{\hat{C}_i\}, \{y_i\})$ **return** AUROC, ECE

Algorithm 2 Semantic Entropy Baseline

Input: Dataset $\mathcal{D}$, LLM $\mathcal{M}$, Embedding Model $\mathcal{E}$, Sampling count K, Distance threshold $\tau = 0.5$

Output: Accuracy, Expected Calibration Error (ECE), Area Under ROC (AUC)

Initialize result set $\mathcal{R} \leftarrow \emptyset$

for all Batch $\{q_i, a_{gold,i}\} \in \mathcal{D}$ **do**

  Generate K independent trajectories for each q_i: $\{s_{i,1}, \ldots, s_{i,K}\} \sim \mathcal{M}(Q_i^p)$

  Extract numerical answers: $\mathcal{A}_i = \{f(s_{i,j})\}_{j=1}^{K}$ ▷ $f(\cdot)$ extracts boxed or terminal digits

  Semantic Clustering:

  Compute normalized embeddings: $\mathbf{v}_{i,j} = \mathcal{E}(\text{str}(a_{i,j}))$

  Perform Agglomerative Clustering on $\{\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,K}\}$ using:

    Metric: Cosine Similarity

    Linkage: Average

    Constraint: $d(\mathbf{v}_a, \mathbf{v}_b) > \tau$ to form new clusters

  Entropy Calculation:

  Count occurrences in each cluster C_m: $c_m = |\{a \in \mathcal{A}_i : \text{label}(a) = m\}|$

  Estimate discrete probability: $p_m = \frac{c_m}{K}$

  Calculate Semantic Entropy: $H_{sem} = -\sum_m p_m \log p_m$

  Derive Confidence Score: $\mathcal{C}_i = e^{-H_{sem}}$

  Evaluation:

  Determine correctness: $y_i = \mathbb{I}(|a_{i,1} - a_{gold,i}| < \epsilon)$

  Add $(\mathcal{C}_i, y_i)$ to $\mathcal{R}$

end for

Compute AUROC:

AUC $= \text{ROC}(\{y_i\}, \{\mathcal{C}_i\})$

Compute ECE:

ECE $= \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|$

return AUC, ECE

Algorithm 3 Self-Consistency Baseline

Require: Dataset $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^{N}$, Model $\mathcal{M}$, Sampling count K
Ensure: Accuracy, ECE, AUROC

for $(Q_i, A_i) \in \mathcal{D}$ **do**
  Multi-Sample Generation:
  Sample K responses: $\{r_{i,j}\}_{j=1}^{K} \sim \mathcal{M}(Q_i^p)$
  Extract answers: $\{a_{i,j}\}_{j=1}^{K}$ from responses
  Voting & Confidence:
  Filter valid answers: $\mathcal{A}_i = \{a \in \{a_{i,j}\} : a \neq \text{None}\}$
  Count votes: $n(z) = |\{a \in \mathcal{A}_i : a = z\}|$ for each unique z
  Prediction:
  $\hat{a}_i \leftarrow \arg\max_z n(z)$ ▷ Most voted answer
  Confidence:
  $C_i \leftarrow \frac{\max_z n(z)}{|\mathcal{A}_i|}$ ▷ Vote ratio
  Compute Correctness: $y_i \leftarrow \mathbb{I}(\hat{a}_i = A_i)$
end for
Compute AUROC:
AUROC $\leftarrow f_{\text{AUC}}(\{C_i\}, \{y_i\})$
Compute ECE:
ECE $\leftarrow f_{\text{ECE}}(\{C_i\}, \{y_i\})$
return AUROC, ECE

Algorithm 4 Mean Token Entropy Baseline

Require: Dataset $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^{N}$, Model $\mathcal{M}$
Ensure: ECE, AUROC

Initialize result set $\mathcal{R} \leftarrow \emptyset$
for each $(Q_i, A_i) \in \mathcal{D}$ **do**
  Generation with Token Probabilities:
  Generate response $r_i \sim \mathcal{M}(Q_i^p)$ with token-level logprobs
  Let $\{p_t\}_{t=1}^{T}$ denote the softmax distributions at each position
  Token-level Entropy Calculation:
  for $t = 1$ to T **do**
    $H_t \leftarrow -\sum_v p_t(v) \log p_t(v)$ ▷ Entropy at position t
  end for
  Mean Token Entropy:
  $\text{MTE}_i \leftarrow \frac{1}{T} \sum_{t=1}^{T} H_t$
  Confidence Score:
  $\mathcal{C}_i \leftarrow \exp(-\text{MTE}_i)$
  Correctness:
  Extract predicted answer $\hat{a}_i$ from r_i
  $y_i \leftarrow \mathbb{I}[\hat{a}_i = A_i]$
  Add $(\mathcal{C}_i, y_i)$ to $\mathcal{R}$
end for
Compute Metrics:
AUROC $\leftarrow f_{\text{AUC}}(\{\mathcal{C}_i\}, \{y_i\})$
ECE $\leftarrow f_{\text{ECE}}(\{\mathcal{C}_i\}, \{y_i\})$
return AUROC, ECE

Algorithm 5 Self-Critique Baseline

Input: Dataset $\mathcal{D}$, LLM $\mathcal{M}$, Embedding Model $\mathcal{E}$, Sampling count K
Output: Accuracy, Expected Calibration Error (ECE), Area Under ROC (AUC)
Initialize empty result set $\mathcal{R}$
for each batch $\{q_i, a_{gold,i}\}$ in $\mathcal{D}$ **do**
    Generate K independent responses for each question q_i
    Extract numerical answers $\mathcal{A}_i$ from the responses
    Phase 1: Initial Answering
    For each response, extract the initial answer from the model output
    Phase 2: Self-Critique
    For each initial answer, generate self-critique prompt and feed into model
    Extract confidence score from the critique response
    Evaluation:
    Determine correctness y_i based on the gold answer
    Add $(\mathcal{C}_i, y_i)$ to $\mathcal{R}$
end for
Compute AUROC:
Compute AUC: AUC $\leftarrow f_{\text{AUC}}(\{C_i\}, \{y_i\})$
Compute ECE:
Compute ECE: ECE $\leftarrow f_{\text{ECE}}(\{C_i\}, \{y_i\})$
return AUC, ECE

Dataset	Method	Skywork-7B		GLM-Z1-9B		DeepSeek-7B		DeepSeek-14B	
		ECE ↓	AUROC ↑	ECE ↓	AUROC ↑	ECE ↓	AUROC ↑	ECE ↓	AUROC ↑
SVAMP	Token	0.2465±0.0104	0.6624±0.0063	0.1628±0.0017	0.4695±0.0225	0.2667±0.0025	0.6440±0.0117	0.1991±0.0109	0.6466±0.0148
	SE	0.2390±0.0116	0.5636±0.0050	0.2289±0.0461	0.5485±0.0188	0.2264±0.0402	0.5740±0.0285	0.2221±0.0031	0.5579±0.0090
	MTE	0.1017±0.0054	0.6382±0.0063	0.0981±0.0433	0.6073±0.0262	0.1130±0.0009	0.6539±0.0134	0.1174±0.0423	0.6533±0.0127
	SC	0.0975±0.0060	0.7621±0.0184	0.1077±0.0033	**0.6995±0.0554**	0.0880±0.0031	0.7583±0.0188	0.1033±0.0029	0.7643±0.0067
	Self-Critique	0.1691±0.0425	0.6522±0.0120	0.0973±0.0121	0.7003±0.0179	0.2365±0.0055	0.6091±0.0136	0.2010±0.0123	0.6421±0.0665
	DisCal-base	0.1284±0.0035	0.8104±0.0157	0.1797±0.0014	0.6420±0.0216	0.1323±0.0022	**0.8302±0.0044**	0.1563±0.0041	0.8046±0.0082
	DisCal	**0.0515±0.0096**	**0.8125±0.0160**	**0.0737±0.0075**	0.6803±0.0237	**0.0524±0.0041**	0.8247±0.0055	**0.0644±0.0143**	**0.8177±0.0102**
GSM8K	Token	0.3372±0.0081	0.6044±0.0091	0.2298±0.0057	0.5187±0.0071	0.3578±0.0070	0.5769±0.0091	0.2746±0.0067	0.5876±0.0108
	SE	0.3525±0.0040	0.5722±0.0050	0.3301±0.0780	0.5457±0.0046	0.3680±0.0050	0.5706±0.0177	0.3028±0.0027	0.5626±0.0052
	MTE	0.2441±0.0066	0.6650±0.0109	0.0648±0.0230	0.6767±0.0390	0.2488±0.0084	0.6926±0.0218	0.1971±0.0117	0.6265±0.0113
	SC	0.1494±0.0051	0.7946±0.0004	0.1328±0.0066	**0.7170±0.0467**	0.1412±0.0068	0.8122±0.0043	0.1538±0.0056	0.7825±0.0055
	Self-Critique	0.2872±0.0006	0.6519±0.0067	0.1594±0.0022	0.7156±0.0204	0.3163±0.0045	0.6125±0.0079	0.2790±0.0107	0.6708±0.0654
	DisCal-base	0.1206±0.0065	0.8227±0.0081	0.1059±0.0018	0.7051±0.0119	0.1245±0.0074	0.8433±0.0070	0.1276±0.0034	0.8255±0.0053
	DisCal	**0.0592±0.0112**	**0.8240±0.0087**	**0.0713±0.0070**	0.7119±0.0114	**0.0492±0.0041**	**0.8486±0.0069**	**0.1081±0.0348**	**0.8310±0.0039**
AIME	Token	0.4450±0.0015	0.6901±0.0089	0.4393±0.0112	0.5844±0.0069	0.4718±0.0066	0.6851±0.0293	0.4559±0.0252	0.6177±0.0368
	SE	0.4704±0.0076	0.6197±0.0057	0.4214±0.0030	0.6301±0.0033	0.4843±0.0038	0.6151±0.0119	0.4929±0.0893	0.6243±0.0154
	MTE	0.2196±0.0101	0.7262±0.0095	0.2480±0.0157	0.6491±0.1254	0.2528±0.0060	0.6924±0.0231	0.3491±0.2061	0.6787±0.0269
	SC	0.3424±0.0060	0.6822±0.0075	0.3142±0.0038	0.7049±0.0146	0.2819±0.0495	0.7439±0.0864	0.2367±0.0508	0.8058±0.0443
	Self-Critique	0.1616±0.0042	0.5999±0.0047	0.2357±0.0069	0.8175±0.0050	0.2526±0.0027	0.6785±0.0054	0.4697±0.1795	0.6185±0.1029
	DisCal-base	0.1130±0.0072	0.8994±0.0035	0.1169±0.0050	0.8866±0.0071	0.1083±0.0030	0.8883±0.0081	0.1170±0.0053	0.8796±0.0059
	DisCal	**0.0613±0.0058**	**0.9059±0.0019**	**0.0679±0.0141**	**0.8988±0.0063**	**0.0797±0.0048**	**0.8904±0.0037**	**0.0800±0.0039**	**0.8944±0.0007**

Table 12: Confidence calibration performance comparison under identical perturbation settings, evaluated using ECE (lower is better) and AUROC (higher is better). Best results are shown in **bold**, with DisCal highlighted.