NoisyCoconut: Counterfactual Consensus via Latent Space Reasoning

Anonymous authors
Paper under double-blind review

Abstract

This paper presents NoisyCoconut, a novel inference-time method that enhances large language model (LLM) reliability by manipulating internal representations. Unlike fine-tuning methods that require extensive retraining, NoisyCoconut operates directly on model representations during inference and requires no retraining. Rather than training models to reason in latent space, we inject controlled noise into latent trajectories to generate diverse reasoning paths. Agreement among these paths provides a confidence signal, enabling models to abstain when uncertain. We demonstrate that this approach achieves effective coverage-accuracy tradeoffs across multiple reasoning benchmarks without requiring access to training data or modification of model parameters. This approach provides a practical pathway to improving the reliability of LLM outputs while maintaining compatibility with existing models. Our experiments show that unanimous agreement among noise-perturbed paths reduces error rates from 40–70% to below 15%, enabling models to exceed 95% accuracy on mathematical reasoning tasks through selective abstention.

1 Introduction

Large language models (LLMs) continue to demonstrate remarkable capabilities and are increasingly deployed in high-stakes environments, including healthcare, financial services, and legal practice (Haltaufderheide & Ranisch, 2024). These nondeterministic models generate text autoregressively, producing one token at a time conditioned on the preceding token sequence based on a probability distribution. While this process yields fluent and coherent outputs, it provides no inherent mechanism for the model to signal uncertainty about its own generations. As a result, LLMs can produce seemingly confident but erroneous outputs, a phenomenon known as hallucination (Xu et al., 2025b; Farquhar et al., 2024). Such errors pose particular risks in regulated sectors, where legal and regulatory obligations demand accuracy and auditability.

One approach to mitigating the risk of unreliable outputs in predictive machine learning systems is selective prediction, which allows models to abstain from giving predictions when it is likely to be incorrect (Geifman & El-Yaniv, 2017; El-Yaniv & Wiener, 2010). This method introduces a tradeoff between coverage (the fraction of queries answered) and accuracy (the correctness of answered queries). In classification settings, selective prediction is well-established based on softmax confidence providing a natural abstention signal (Geifman & El-Yaniv, 2017; Guo et al., 2017). Enabling confidence-based abstention for generative LLMs is less straightforward, however, because token-level probabilities often reflect linguistic uncertainty rather than factual correctness (Kuhn et al., 2023). The main problem is how to identify instances where abstention is warranted.

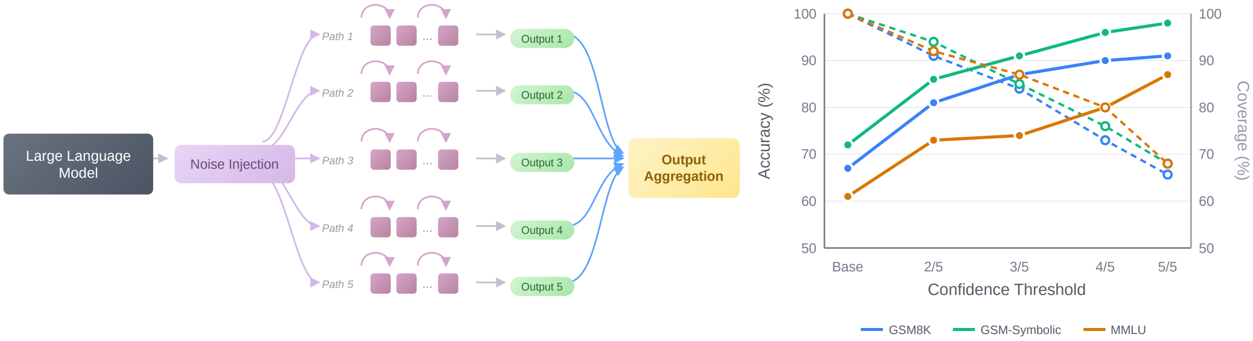


Figure 1: (Left) NoisyCoconut architecture with noise-induced branching for diverse reasoning paths. (Right) Accuracy-coverage tradeoff for Qwen2.5-7B-Instruct across three benchmarks. Solid lines show accuracy, while the dashed lines show coverage.

Several methods have been proposed for estimating the confidence of generative LLMs at inference-time. Verbalized confidence methods prompt models to explicitly rate their certainty, though such self-assessments can be poorly calibrated (Kadavath et al., 2022; Xiong et al., 2024). Self-correction approaches (e.g., Madaan et al., 2023; Shinn et al., 2023) iteratively refine outputs through self-generated feedback, though recent work suggests that intrinsic self-correction struggles for reasoning tasks without external signals (Huang et al., 2024). Sampling-based methods generate multiple outputs and measure their consistency, using majority voting across multiple, generated paths or clustering semantically equivalent responses to compute uncertainty over meanings (e.g., Wang et al., 2023; Kuhn et al., 2023). Thus, when sampled outputs disagree, the system can abstain rather than producing an unreliable answer, enabling coverage–accuracy tradeoffs. However, because these methods rely on standard autoregressive sampling, they are restricted to the diversity present in the surface-level token distribution at the output layer (Holtzman et al., 2020). Consequently, they often fail to explore a sufficiently broad set of reasoning trajectories, as the sampling process is confined by the language space.

A separate line of research has begun exploring methods that operate directly on the continuous hidden states of language models, rather than through token generation (Zhu et al., 2025c; Goyal et al., 2023; Hao et al., 2025). These approaches reflect a growing recognition that the discrete, token-by-token generation process may be suboptimal for complex reasoning tasks and that reasoning in neural networks involves operations in a continuous representational space not readily accessible through token manipulations. However, latent-space methods face the same reliability challenges as their token-space counterparts, as models can produce incorrect outputs with no inherent mechanism for assessing confidence. Moreover, all of these methods require specialized architectures and training to enable latent-space reasoning.

We address this gap by extending agreement-based confidence estimation to latent space. We build on the Continuous Chain-of-Thought (Coconut) framework from Hao et al. (2025), which trains language models to perform iterative computation in hidden states by feeding the last hidden state back as the next input embedding. Our goal differs from Coconut's—rather than training models to reason in latent space, we leverage latent representations to assess confidence in model outputs. Specifically, we introduce NoisyCoconut, illustrated in Figure 1. NoisyCoconut injects controlled noise into latent trajectories to generate counterfactual reasoning paths. This enables selective prediction for latent-space methods, achieving effective coverage–accuracy tradeoffs without additional training.

Contributions. We make these main contributions: (1) We introduce NoisyCoconut, a training-free inference-time method that generates diverse reasoning paths through controlled noise injection in continuous latent space, enabling agreement-based confidence estimation directly on hidden states (Section 3). (2) We demonstrate that path agreement provides a reliable proxy for prediction confidence (Section 4). (3) We characterize the noise-accuracy relationship in latent space (Section 4.2). (4) We show that NoisyCoconut achieves effective coverage-accuracy tradeoffs across three reasoning benchmarks and five language models (Section 4.3).

2 Background and Related Work

Our work connects two research threads: inference-time methods for improving LLM reliability and latent-space computation. We review each area before describing the Coconut framework we build upon.

2.1 Confidence Estimation and Selective Prediction

Selective prediction for LLMs in generative settings presents fundamentally different challenges than selective prediction for classification tasks (Xin et al., 2021). In classification, models select from a fixed set of discrete labels, but autoregressive generation produces outputs token-by-token over many steps, with an effectively unbounded space of possible responses (Ren et al., 2023). This distinction has motivated a substantial body of work developing uncertainty quantification methods tailored for generative models (Si et al., 2023; Varshney et al., 2022; Kadavath et al., 2022; Kuhn et al., 2023). Prior works have examined selective prediction in settings where questions themselves are ambiguous, requiring models to recognize when clarification is needed rather than producing potentially incorrect answers (Cole et al., 2023; Yin et al., 2023). Another line of research employs auxiliary models trained to discriminate between correct and incorrect predictions from a primary QA system (Varshney & Baral, 2023), though such approaches introduce additional training requirements and may not generalize across domains.

These challenges have motivated sampling-based approaches that operationalize selective prediction by generating multiple outputs and using their agreement as a confidence signal for abstention decisions. Self-consistency (Wang et al., 2023) samples N reasoning chains from the model, then selects the most frequent final answer via majority voting. The key insight is that correct answers tend to be reachable via multiple reasoning paths, while incorrect answers arise from idiosyncratic errors unlikely to recur across samples. This approach substantially improves accuracy on arithmetic and commonsense reasoning benchmarks, though it requires generating N complete token sequences per query.

An alternative direction uses training to improve selective prediction. ASPIRE (Chen et al., 2023b) fine-tunes LLMs via parameter-efficient tuning to not only answer questions but also evaluate whether their generated answers are correct, producing explicit confidence scores. While effective, such approaches require task-specific training data and model access, limiting applicability.

Semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024) addresses a limitation of token-level uncertainty measures since different surface forms can express the same meaning. Rather than computing entropy over token sequences, semantic entropy clusters sampled responses by meaning using bidirectional entailment, then computes entropy over these semantic equivalence classes. This provides uncertainty estimates that are invariant to paraphrasing and has been shown to detect hallucinations more reliably than token probabilities alone.

Verbalized confidence methods take a different approach, prompting models to directly express uncertainty. Kadavath et al. (2022) introduced P(True), which asks models to evaluate whether their own sampled answers are correct; the probability assigned to the token “True” serves as a confidence estimate. While effective for well-calibrated models, subsequent work has shown that verbalized confidence can be poorly calibrated, particularly for instruction-tuned models (Xiong et al., 2024; Tian et al., 2023). Probing-based methods instead train classifiers on internal representations to predict correctness (Azaria & Mitchell, 2023), though these require labeled data for each task.

Recent work has extended these approaches in various directions. Confidence-Informed Self-Consistency (Taubenfeld et al., 2025) weights the majority vote by model confidence, reducing the number of samples needed. Kernel Language Entropy (Nikitin et al., 2025) generalizes semantic entropy using kernel methods for finer-grained uncertainty estimates.

2.2 Inference-Time Approaches to Improving LLM Reliability

Token-Space Methods. Beyond confidence estimation, several inference-time methods seek to improve LLM reliability by structuring the generation process itself. Chain-of-Thought (Wei et al., 2022) improves accuracy on complex tasks by eliciting intermediate steps, reducing errors that arise from single-step generation. This approach has been extended through decomposition strategies (Khot et al., 2022; Zhou et al., 2022) and explicit search methods such as Tree-of-Thoughts (Yao et al., 2023), which explores multiple solution paths and enables backtracking.

Self-correction approaches attempt to improve reliability through iterative refinement. Self-Refine (Madaan et al., 2023) prompts models to critique and revise their own outputs, while Reflexion (Shinn et al., 2023) maintains memory of past errors to guide future attempts. However, Huang et al. (2024) demonstrate that intrinsic self-correction, without external feedback, often fails to improve accuracy on reasoning tasks.

These methods operate entirely in the language and token-space, generating explicit text at each step. This limits the model to express all intermediate computation through natural language. As Madaan & Yazdanbakhsh (2022) observe, most tokens in a reasoning chain serve textual coherence rather than substantive computation.

Latent-Space Computation. An alternative approach performs computation directly in the continuous hidden states of language models, bypassing token generation. Studies have shown that intermediate reasoning steps can be recovered from hidden representations (Yang et al., 2024) and that models may employ latent processes distinct from their generated text (Turpin et al., 2024). Zhu et al. (2025c) categorize latent-space methods into activation-based approaches that expand computational depth through repeated layer processing, hidden state-based methods that maintain compressed memory states, and diffusion-based approaches enabling iterative refinement.

Chain of Continuous Thought (Coconut). Prior work has explored augmenting LLMs with special tokens that enable additional hidden-state computation, such as learnable pause tokens (Goyal et al., 2023) and filler tokens (Pfau et al., 2024). However, these approaches require training and still operate through the token interface. Hao et al. (2025) introduced COCONUT (Chain of Continuous Thought), which enables reasoning entirely in continuous latent space. Rather than generating intermediate tokens, COCONUT feeds the model's last hidden state back as the next input embedding. Formally, given an initial hidden state $\mathbf{h}_0$, the model iteratively computes $\mathbf{h}_{t+1} = f_\theta(\mathbf{h}_t)$ where $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ represents a forward pass through the transformer. This formulation enables reasoning without intermediate tokenization, achieving strong performance on tasks requiring search and planning.

Analysis of COCONUT reveals structured latent dynamics: hidden states exhibit exploration phases with rapid movement through representation space, followed by convergence phases as solutions crystallize. The model learns to maintain stable representations of problem constraints while iteratively refining candidate solutions. Several methods have built on this foundation, including theoretical frameworks for understanding reasoning through superposition of computational states (Zhu et al., 2025b) and demonstrations of parallel exploration in continuous representations (Gozeten et al., 2025b).

Our work bridges confidence estimation and latent-space reasoning. We observe that COCONUT's continuous states provide a natural substrate for diversification, and injecting controlled noise into the hidden state trajectory induces branching into alternative reasoning paths. By measuring agreement among these paths, analogous to how self-consistency measures agreement across sampled token sequences, we obtain confidence estimates that enable coverage-accuracy tradeoffs. Unlike COCONUT, which trains models for latent reasoning, NOISYCOCONUT operates at inference time to assess confidence in model outputs, requiring no additional training.

3 NoisyCoconut

The core idea of NOISYCOCONUT is to perturb the hidden states during the continuous hidden state space process, creating multiple reasoning trajectories that explore different regions of the solution space. Agreement among diverse reasoning paths provides stronger evidence for correctness, and allows for accuracy–coverage tradeoffs to improve reliability.

3.1 Noise-Based Path Exploration

The method of NoisyCoconut is simple—we inject noise into the internal process of an LLM and aggregate results across multiple paths with different noise injection. We sample a random noise pattern from a Gaussian distribution and inject it into the last hidden layer of the first forward pass of an LLM to create multiple reasoning paths from a common initial state. Ideally, the noise should be large enough to induce meaningful divergences, yet constrained enough to maintain coherent sequences. We hypothesize that this perturbation creates divergences in the latent space, yielding disparate reasoning chains. If these chains reach the similar conclusions, this increases our confidence that the model's prediction is correct.

Let $\mathcal{M}$ be a pre-trained language model with hidden dimension d. For input query $q \in \mathcal{Q}$, let $\mathbf{h}_0 = \phi(q) \in \mathbb{R}^d$ denote the initial hidden state from the first forward pass. The NoisyCoconut process evolves as:

$$\mathbf{h}_{t+1} = f_\theta(\mathbf{h}_t + \boldsymbol{\eta}_t), \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 \mathbf{I}_d) \tag{1}$$

where $\{\boldsymbol{\eta}_t\}$ are independent and the noise scale decays exponentially:

$$\sigma_t = \sigma_0 e^{-\lambda t}, \quad \sigma_0, \lambda > 0 \tag{2}$$

We adapt the noise based on trajectory properties. Let μ_t denote the exponentially-weighted moving average of the hidden state norm, updated as:

$$\mu_t = \alpha \mu_{t-1} + (1-\alpha)\|\mathbf{h}_t\|_2, \quad \alpha \in (0,1) \tag{3}$$

Thus, the adaptive noise scale is:

$$\sigma_t = \sigma_0 e^{-\lambda t} \cdot \frac{\|\mathbf{h}_t\|_2}{\mu_t} \tag{4}$$

3.2 Path Diversity

To ensure effective exploration of the solution space, we require sequences that are sufficiently disparate. Our goal is to produce perturbations that result in distinct paths rather than minor variations of the same solution strategy.

For K paths $\{\mathbf{h}_{0:T}^{(i)}\}_{i=1}^K$, we define pairwise trajectory diversity as:

$$\mathcal{D}_K = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \frac{1}{T} \sum_{t=0}^{T-1} \|\mathbf{h}_t^{(i)} - \mathbf{h}_t^{(j)}\|_2 \tag{5}$$

The expected diversity scales with noise: $\mathbb{E}[\mathcal{D}_K] = \Omega(\sigma_0 \sqrt{dT})$ under independent path divergence.

3.3 Aggregating Outputs

Each of the K generated paths yields an output, which we denote as $\{y^{(i)}\}_{i=1}^K \in \mathcal{Y}$. To produce the consensus output, we aggregate these generated outputs to either produce a consensus output or abstain (denoted as $\perp$). We choose a majority voting strategy by selecting the output that appears most frequently among the K generated paths, treating each path equally regardless of its generation process or characteristics:

$$\hat{y} = \begin{cases} \hat{y}^\star & \text{if } \exists \hat{y}^\star \in \mathcal{Y} \text{ s.t. } \sum_{i=1}^K \mathbb{I}\{y^{(i)} = \hat{y}^\star\} > \frac{K}{2} \\ \perp & \text{otherwise} \end{cases} \tag{6}$$

While Equation 6 defines the general selection criteria for any K, for our experimental analysis (where $K = 5$), we specifically categorize the consensus patterns into mutually exclusive outcomes to diagnose reliability based on the minimum number of paths that must agree for the system to produce an answer:

- **Unanimous (5/5):** All five paths converge to the exact same answer.
- **Strong Majority (4/5):** Four paths agree on an answer, while one diverges.
- **Moderate Majority (3/5):** Three paths agree on an answer.
- **Minimal Plurality (2/5):** Two paths agree on an answer, while the other three are distinct (e.g., A, A, B, C, D).

These categories allow us to map the latent stability of the model to the correctness of the final output in our experiments, and would provide an accuracy–coverage tradeoff in a deployed system. In cases where no answer appears on more than two paths, the system will always abstain.[1]

4 Evaluation and Results

To understand the coverage–accuracy tradeoffs enabled by NoisyCoconut, we conduct a systematic evaluation across five language models and three benchmarks. Section 4.1 describes our experimental setup, including model selection, benchmarks, and implementation details. We then validate a core assumption of our method by characterizing the noise-accuracy relationship as the noise scale varies (Section 4.2). Section 4.3 presents our main results, analyzing how agreement patterns among diverse reasoning paths correlate with accuracy. Our findings show that path agreement in latent space provides a strong signal for prediction reliability. Unanimous agreement among five noise-perturbed reasoning paths reduces error rates from 40–70% to below 15%, enabling models to achieve over 95% accuracy on mathematical reasoning benchmarks when selectively abstaining on low-confidence predictions.

4.1 Experimental Setup

We selected representative open-source language models of similar parameter scales to evaluate the effectiveness of NoisyCoconut across different architectures and training paradigms. The selection includes a mix of instruction-tuned models, base models, and a specialized distilled reasoning model to assess performance across different model optimization approaches.

Table 1: Language models used in evaluation

Model	**Parameters**	**Type**
Qwen2.5-7B-Instruct	7B	Instruction-tuned
Llama-3.1-8B-Instruct	8B	Instruction-tuned
Mixtral-8B-Instruct-v0.1	8B	Instruction-tuned
gpt-oss-20B	20B	Foundation model
DeepSeek-R1-Distill-Qwen-7B	7B	Distilled reasoning

Table 1 summarizes the models tested. Qwen2.5-7B-Instruct (Qwen Team, 2024) is a recent instruction-tuned model from the Qwen family, known for strong reasoning capabilities despite its moderate size. Llama-3.1-8B-Instruct (Grattafiori et al., 2024) is one of Meta's smaller, instruction-tuned variants in the Llama series, widely used as a foundation for many applications. Mixtral-8B-Instruct-v0.1 (Jiang et al., 2024) provides another instruction-tuned variant, optimized specifically for following complex instructions with enhanced reasoning capabilities. gpt-oss-20B (OpenAI, 2025) is an open-weights model that offers strong

---

[1] Although one can imagine future work that attempts to find the most likely answer even with no explicit agreement or that derives more information from the different paths by considering token probabilities or other factors, we do not explore that in this work, but keep things simple by just counting outcomes.

general-purpose language abilities. Finally, DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) is a specialized distilled reasoning model that combines elements from both DeepSeek and Qwen architectures, where knowledge distillation techniques were specifically employed to enhance reasoning capabilities.

While our NoisyCoconut implementation works directly with standard architectures (Qwen-2.5-7B-Instruct, DeepSeek-R1-Distill-Qwen-7B), we observed that gpt-oss-20B produced degenerate outputs when using the default final-layer hidden state feedback mechanism, regardless of noise level. Thus, we use $\mathbf{h}^{(1)}$ (first layer) instead of $\mathbf{h}^{(L)}$ (final layer) for the continuous sequence. This modification enables functional operation, though it reduces the effective reasoning depth per latent pass. All other models use the standard final-layer configuration.

Benchmarks. To evaluate both mathematical reasoning and broader knowledge capabilities, we selected three established benchmarks. GSM8K (Cobbe et al., 2021) provides a challenging grade-school math word problem benchmark consisting of 1000 linguistically diverse problems requiring multi-step reasoning. GSM-Symbolic (Mirzadeh et al., 2025) represents a variant of GSM8K that replaces natural language descriptions with more symbolic representations of the same underlying problems, testing how well models handle different formulations of equivalent mathematical concepts. The MMLU (Massive Multitask Language Understanding) (Hendrycks et al., 2021) benchmark offers a comprehensive assessment covering 57 subjects across STEM, humanities, social sciences, and more, testing both factual knowledge and reasoning abilities. For each benchmark, we randomly sampled 1000 questions to ensure sufficient statistical power while maintaining computational feasibility.

4.2 Validating Impact of Noise

A fundamental assumption of NoisyCoconut is that injecting noise into hidden states produces controlled and predictable degradation in model performance. To validate this assumption and inform our choice of noise level, we systematically measured accuracy across noise scales $\sigma \in [0, 50]$, where σ represents the ratio of noise norm to hidden state norm ($\|\boldsymbol{\epsilon}\|_2 = \sigma\|\mathbf{h}\|_2$).

Figure 2 summarizes the results, showing a characteristic sigmoid decay pattern across all models, with fits achieving $R^2 \geq 0.94$. Accuracy remains relatively stable at low noise levels before undergoing steep degradation, eventually plateauing near random performance. This consistent pattern across architectures confirms that noise injection produces predictable, exploitable effects.

Based on these results, we selected $\sigma_0 = 0.2$ (marked by the vertical dotted line in Figure 2) as the noise level for our main experiments. This value lies in the early portion of the degradation curve, where accuracy remains within 2–5% of unperturbed performance. This choice introduces sufficient perturbation to generate meaningful diversity in generated paths, while avoiding excessive degradation that would compromise individual path quality. Further analysis of this trade-off is provided in Appendix B.

4.3 Results and Analysis

We implemented NoisyCoconut as described in Section 3, with several key hyperparameters selected to balance performance and computational cost.

For our main experiments, we use five reasoning paths per problem, which our ablation studies (Appendix B) show provides a good efficiency-performance tradeoff. As mentioned earlier, based on our experiments in Section 4.2 we set the noise scale to $\sigma_0 = 0.2$. Noise injections were performed at the first hidden state after the first forward pass. Early injection enables greater exploration of the reasoning space and aligns with our goal of generating diverse reasoning trajectories from the outset. We set a maximum of eight latent thinking steps to allow sufficient reasoning depth, following the same constraint established in the original Coconut work.[2]

For each model and benchmark combination, we first establish a baseline by evaluating the model using standard greedy decoding without inference-time modifications. We then apply Noisy Coconut, generating

---

[2] Note that these component-wise ablations do not capture interaction effects between hyperparameters.

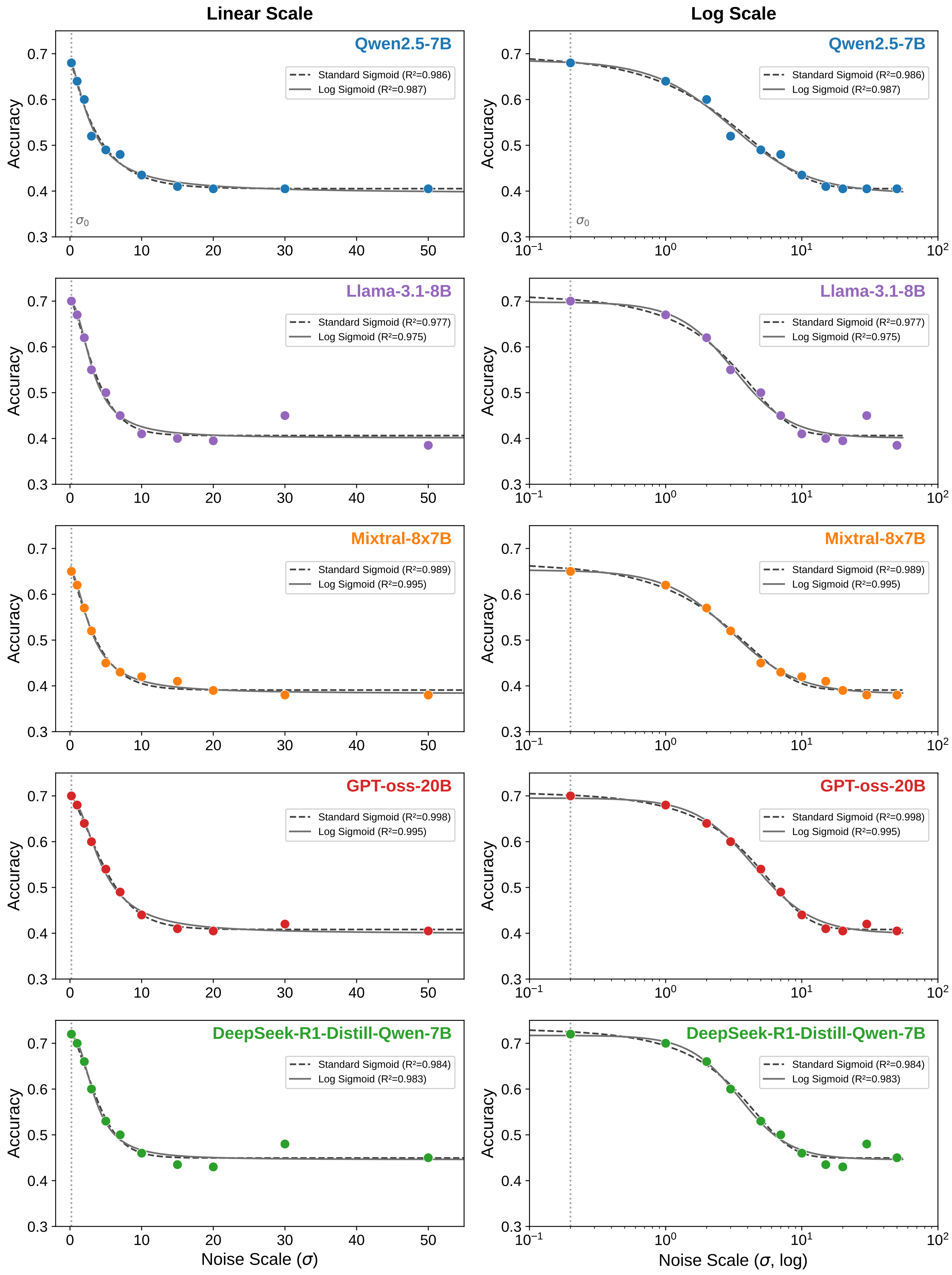
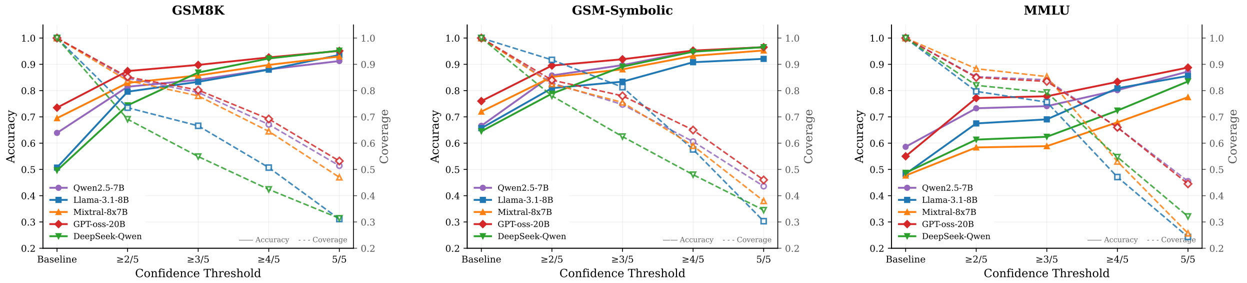


Figure 2: Accuracy degradation vs. noise scale σ on linear (left) and logarithmic (right) scales. The vertical dotted line marks $\sigma_0 = 0.2$, chosen to balance perturbation strength with model performance (see Section B). Sigmoid fits achieve $R^2 \geq 0.94$ for all models.



(a) GSM8K

(b) GSM-Symbolic

(c) MMLU

Figure 3: Coverage-accuracy trade-off across confidence thresholds. Solid lines denote accuracy and dashed lines denote coverage. As the agreement threshold increases from baseline to unanimous, accuracy improves while coverage decreases correspondingly.

five ($K = 5$) distinct reasoning paths for each test example by injecting controlled noise into the hidden states. We analyze the generated paths using the aggregation outcomes defined in Section 3. While we track distinct outcomes such as “Split Votes” for error analysis, our primary performance evaluation focuses on the cumulative confidence threshold. We perform this generation step once per test example, and then analyze the impact of applying the different aggregation functions to this static set of outputs. This allows us to observe how the consensus output $\hat{y}$, which is either a specific answer $\hat{y}^{\star}$ or an abstention $\perp$, changes as a function of the strictness of the agreement threshold.

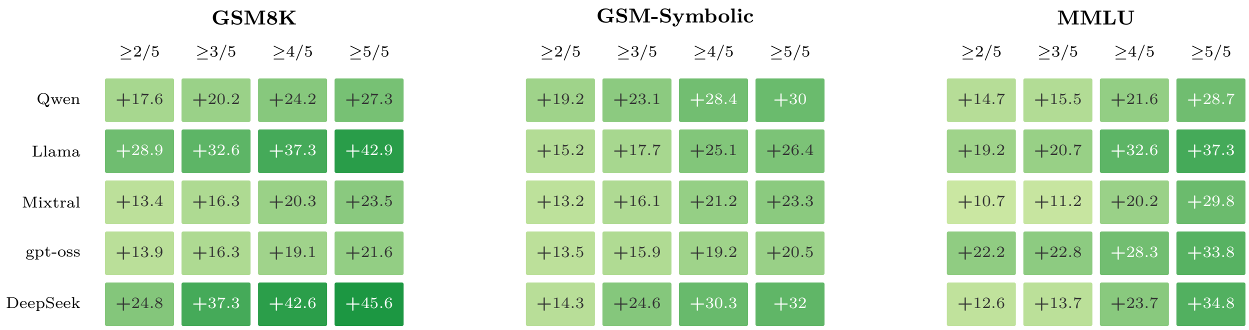
Figure 3 illustrates the fundamental trade-off enabled by our method. We observe a consistent relationship between the agreement threshold and accuracy. As the threshold increases from a plurality ($\geq 2/5$ agreement) through majority requirements ($\geq 3/5$) to unanimous agreement ($\geq 5/5$), the reliability improves significantly.

Across the five language models and three benchmarks, the path agreement confidence measure enabled by NoisyCoconut provides a trade-off between *coverage* (the proportion of questions for which the system provides an answer) and *accuracy* (the correctness of answers output when the model does not abstain).

We observe a consistent relationship between agreement threshold and accuracy, with performance improving as the threshold increases from the baseline (single-path inference) through ≥ 2, ≥ 3, ≥ 4 and unanimous agreement thresholds.

Our results demonstrate that NoisyCoconut offers a mechanism for enhancing predictive accuracy through selective abstention on low-confidence predictions. The magnitude of improvement on mathematical benchmarks is particularly noteworthy. On GSM8K, no baseline model exceeds 75% accuracy, yet at the unanimous (5/5) agreement threshold, all models exceed 90% accuracy, with coverage ranging from 31.1% (Llama-3.1-8B) to 53.2% (gpt-oss-20B). The largest accuracy improvement is observed for DeepSeek-Qwen, which improves from 62.0% accuracy at the baseline to 95.2% at unanimous agreement, while maintaining 31.4% coverage. Similar patterns emerge across all evaluated models, suggesting that path agreement constitutes an effective method for measuring prediction reliability.

Accuracy Improvement Analysis. Figure 4 shows the accuracy gains achieved at each confidence threshold relative to baseline performance. The heatmap reveals consistent improvements across all model–benchmark combinations, with gains ranging from approximately 10 percentage points at the lowest agreement threshold to over 40 percentage points with unanimous agreement for certain models. Models with lower baseline accuracy exhibit the largest absolute improvements, achieving near-parity with stronger models when predictions are restricted to high-agreement instances. This finding suggests that NoisyCoconut can serve as an equalizing mechanism, enabling weaker models to approach the reliability of stronger counterparts on the subset of questions where they exhibit high confidence. The continuous increase in improvement magnitude from further corroborates the effectiveness of agreement level as a confidence signal.



GSM8K

	≥2/5	≥3/5	≥4/5	≥5/5
Qwen	+17.6	+20.2	+24.2	+27.3
Llama	+28.9	+32.6	+37.3	+42.9
Mixtral	+13.4	+16.3	+20.3	+23.5
gpt-oss	+13.9	+16.3	+19.1	+21.6
DeepSeek	+24.8	+37.3	+42.6	+45.6

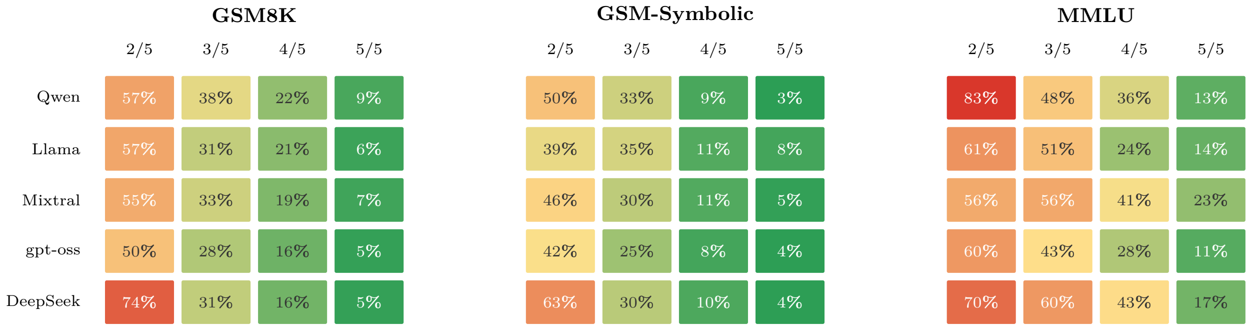
GSM-Symbolic

	≥2/5	≥3/5	≥4/5	≥5/5
Qwen	+19.2	+23.1	+28.4	+30
Llama	+15.2	+17.7	+25.1	+26.4
Mixtral	+13.2	+16.1	+21.2	+23.3
gpt-oss	+13.5	+15.9	+19.2	+20.5
DeepSeek	+14.3	+24.6	+30.3	+32

MMLU

	≥2/5	≥3/5	≥4/5	≥5/5
Qwen	+14.7	+15.5	+21.6	+28.7
Llama	+19.2	+20.7	+32.6	+37.3
Mixtral	+10.7	+11.2	+20.2	+29.8
gpt-oss	+22.2	+22.8	+28.3	+33.8
DeepSeek	+12.6	+13.7	+23.7	+34.8

Figure 4: Cumulative accuracy improvement over baseline (in percentage points) at each confidence threshold. Higher agreement requirements yield larger improvements across all models and benchmarks. Values represent the accuracy gain when the model provides an answer at that confidence threshold.



GSM8K

	2/5	3/5	4/5	5/5
Qwen	57%	38%	22%	9%
Llama	57%	31%	21%	6%
Mixtral	55%	33%	19%	7%
gpt-oss	50%	28%	16%	5%
DeepSeek	74%	31%	16%	5%

GSM-Symbolic

	2/5	3/5	4/5	5/5
Qwen	50%	33%	9%	3%
Llama	39%	35%	11%	8%
Mixtral	46%	30%	11%	5%
gpt-oss	42%	25%	8%	4%
DeepSeek	63%	30%	10%	4%

MMLU

	2/5	3/5	4/5	5/5
Qwen	83%	48%	36%	13%
Llama	61%	51%	24%	14%
Mixtral	56%	56%	41%	23%
gpt-oss	60%	43%	28%	11%
DeepSeek	70%	60%	43%	17%

Figure 5: Error rates at each agreement level across models and benchmarks. Lower agreement levels (2/5, 3/5) consistently exhibit higher error rates (shown in red/orange), while high-agreement predictions (4/5, 5/5) achieve substantially lower error rates (shown in green). This confirms that path disagreement effectively identifies unreliable predictions suitable for abstention.

Error Distribution. Figure 5 presents another perspective through examination of error rates stratified by agreement level. Test examples where there is low path agreement exhibit substantially elevated error rates, typically ranging from 40% to 70%, whereas high-agreement predictions demonstrate markedly lower error rates, generally below 15% for instances of unanimous agreement. These results are remarkably consistent across the five tested models and three benchmarks.

This concentration of errors within low-agreement categories provides direct empirical validation for the assumption underlying NoisyCoconut: path disagreement serves as a reliable indicator of prediction uncertainty, effectively identifying instances where model outputs are unreliable and where abstention or additional verification would be warranted. The consistency of this pattern across diverse models and benchmarks underscores the robustness of agreement-based confidence estimation as a general-purpose mechanism for uncertainty quantification in language model inference.

5 Discussion

By injecting controlled noise into hidden states rather than operating at the token level, NoisyCoconut provides an inference-time method that introduces an accuracy–coverage tradeoff which can be used to enhance LLM reliability without any need for retraining. The consistent relationship between path agreement and accuracy suggests that uncertainty in LLMs may be visible in latent space. When noise perturbations lead to divergent reasoning paths, this likely indicates the model is operating in a region of representational instability where small changes in hidden states lead to substantially different outputs. Conversely, when perturbed paths converge to the same answer, the model appears to be in a stable attractor basin where the solution is robust to perturbation. This interpretation aligns with recent theoretical work analyzing how reasoning emerges through superposition of computational states in continuous representations (Zhu et al., 2025a).

The sigmoid degradation pattern observed in Figure 2, with fits achieving $R^2 \geq 0.94$ across all models, suggests a phase transition in model behavior. At low noise levels, the model's reasoning process is robust enough to absorb perturbations without changing outputs. Beyond a critical threshold, perturbations overwhelm the signal, causing rapid degradation to near-random performance. The consistency of this pattern across architectures, from instruction-tuned models to distilled reasoning models, indicates this may be a fundamental property of how transformer representations encode reasoning processes rather than an artifact of specific training procedures.

Our finding that weaker baseline models exhibit larger absolute improvements under high-agreement filtering merits further investigation. One hypothesis is that weaker models possess latent capabilities that are inconsistently activated during standard inference. NoisyCoconut may function as a capability elicitation mechanism, identifying instances where the model "knows" the answer but requires favorable initialization to reliably surface it. This speculation connects to work on latent knowledge in language models (Christiano et al., 2021) and suggests that agreement-based filtering could complement techniques designed to extract reliable knowledge from uncertain models.

5.1 Limitations

Generalizability. NoisyCoconut requires the ability to access and perturb internal model states, limiting applicability to open-weight models or settings with sufficient API access. The method cannot currently be applied to closed models accessible only through text-based APIs. As latent-space methods gain prominence, API designs that expose intermediate representations could enable broader application of techniques like ours (Gao et al., 2023).

Our method's effectiveness varies across architectures. Most notably, gpt-oss-20B required using first-layer hidden states rather than final-layer states for the continuous feedback loop, as the default configuration produced degenerate outputs. This sensitivity suggests that the structure of hidden representations differs meaningfully across model families, and optimal noise injection strategies may need architecture-specific tuning. Understanding why certain architectures respond differently to latent perturbation remains an open question.

Discrete Responses. Our evaluation focused on mathematical reasoning and knowledge-intensive tasks where the set of responses is small (multiple-choice questions) and agreement is well-defined. Extending NoisyCoconut to open-ended generation tasks where "agreement" requires semantic similarity rather than exact matching poses a difficult, but we think not impenetrable, challenge. Integration with semantic entropy methods (Kuhn et al., 2023; Farquhar et al., 2024) could enable agreement-based confidence for tasks like summarization or translation, where multiple valid outputs exist. Additionally, exploring whether latent perturbation can improve factuality in long-form generation, where hallucination risks compound, represents a practically important direction.

Design space exploration. While we identified $\sigma_0 = 0.2$ as effective across our experiments, the optimal noise scale likely depends on task difficulty, model capacity, and input characteristics. Our current approach uses fixed hyperparameters, but adaptive noise scaling based on input uncertainty or model confidence could improve robustness. The exponentially weighted moving average adaptation in Equation 4 represents a first step, but more sophisticated approaches drawing on adaptive gradient methods (Kingma & Ba, 2015) or learned noise schedules (Ho et al., 2020) may prove beneficial.

Our evaluation was limited to consensus by voting as the aggregation strategy, where outputs are compared via exact match. The vast space of possible aggregation strategies remains largely unexplored. More sophisticated approaches such as learned aggregation functions that consider output semantics, confidence-weighted schemes based on token-level probabilities, or methods that leverage partial agreement structure may yield improved performance. Additionally, our exact-match criterion for agreement is well-suited to mathematical reasoning tasks with unique correct answers but may be overly restrictive for domains where semantically equivalent but lexically distinct outputs are valid. Integrating semantic similarity measures (Kuhn et al., 2023) or embedding-based comparison into the aggregation step could extend NoisyCoconut to open-ended generation tasks where multiple valid phrasings exist. Making the aggregation adaptive would also offer additional opportunities, using additional executions only when necessary to reduce cost, and providing finer grained methods for determining when to abstain, when to invest more compute, and when there is sufficient confidence to produce a reliable output.

Cost. A straightforward implementation of NoisyCoconut requires computational overhead to generating K paths that scales approximately linearly with K. While we believe $K = 5$ is already a reasonable cost for improved reliability in many settings, this may be prohibitively expensive for many applications where inference costs already dominate.

Current NoisyCoconut generates paths independently, missing opportunities for computation sharing. Speculative decoding techniques (Leviathan et al., 2023; Chen et al., 2023a) demonstrate that parallel verification can be substantially cheaper than parallel generation. Analogously, architectures that share early computation across paths while branching only at key decision points could reduce overhead while maintaining diversity. Tree-structured approaches (Yao et al., 2023) provide one template, though adapting these to continuous latent space rather than discrete token space requires further development.

Techniques for early termination when paths show rapid convergence could mitigate this cost. It may also be the case that multiple executions of a smaller model that can be run locally within the NoisyCoconut framework can be used to provide more reliable results for lower cost than would be required for a single execution of an expensive proprietary model.

Controlling the computational budget allocated to reasoning is an active research area. Length Controlled Policy Optimization (LCPO) enables precise control over reasoning length, allowing models to generate outputs adhering to user-specified length constraints (Aggarwal & Madaan, 2025). Similarly, the simple test-time scaling (s1) approach introduces “budget forcing” as a technique to control test-time computation by either terminating the model’s thinking process early or extending it to encourage further reasoning (Muennighoff et al., 2025). Both approaches reveal that models can adapt their reasoning strategies based on available computational resources. Combining these insights with NoisyCoconut suggests an adaptive approach: allocate more reasoning paths to instances showing early signs of disagreement while terminating early when paths rapidly converge. Such selective computation could maintain reliability gains while substantially reducing average-case overhead.

5.2 Future Directions

Integration with Learned Latent Reasoning. One promising direction involves architectures specifically designed to leverage the latent space directly for extended reasoning. Recent work has introduced recurrent latent reasoning architectures that enable scaling test-time computation through iterative processing in latent space (Tan et al., 2025; Orlicki, 2025; Xu et al., 2025a). These approaches employ core recurrent blocks that can be executed multiple times before producing a final output, allowing models to perform more computation without generating additional tokens. Since these architectures explicitly optimize representations for iterative reasoning, they may exhibit even more structured uncertainty geometry that NoisyCoconut could exploit. Investigating whether our agreement signal remains predictive—or becomes even more informative—when applied to such architectures represents a natural extension.

Language Mixing and Code-Switching Phenomena. During our experiments with varying noise thresholds, we observed an unexpected phenomenon: at certain perturbation levels, models exhibited spontaneous language mixing, producing reasoning traces that alternated between English and other languages (e.g., Chinese) despite receiving English-only prompts (Appendix C). Intriguingly, these code-switched outputs sometimes yielded correct answers where monolingual reasoning failed, suggesting that language mixing may activate alternative reasoning pathways or access knowledge encoded differently across linguistic subspaces. This observation aligns with recent systematic studies by Wang et al. (2025), who demonstrate that language mixing in reasoning language models reflects latent processing preferences and that forcing models to reason in specific scripts can notably improve accuracy. The broader phenomenon connects to emerging research on latent chain-of-thought reasoning, which decouples reasoning from explicit language generation (Chen et al., 2025), and work showing that activation-space perturbations can encode complex reasoning patterns (Zhang & Viteri, 2025). However, we did not pursue this direction further, as we lacked a principled mechanism to predict when code-switching would be beneficial or to reliably induce it. Developing methods to detect instances where language mixing improves reasoning—and techniques to controllably trigger such behavior—represents a promising avenue for future work, potentially combining our agreement-based confidence estimation with script-aware decoding strategies.

Theoretical Foundations. The empirical success of agreement-based confidence estimation invites theoretical investigation. Why should path diversity in latent space correlate with correctness? Recent work by Zhu et al. (2025a) provides one lens through their analysis of reasoning via superposition, suggesting that correct solutions may correspond to more stable superposition states. Gozeten et al. (2025a) demonstrate that continuous representations enable parallel exploration of solution paths, which may explain why noise injection effectively diversifies reasoning. Developing a formal framework connecting latent geometry, perturbation stability, and output correctness could guide principled improvements to noise injection strategies and provide guarantees on when agreement-based confidence is reliable.

References

Pranjal Aggarwal and Aman Madaan. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying, 2023. URL `https://arxiv.org/abs/2304.13734`.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Hoffman. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023a.

Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan O Arik, Tomas Pfister, and Somesh Jha. Adaptation with self-evaluation to improve selective prediction in LLMs. In *EMNLP*, 2023b.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025.

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. *Alignment Forum*, 2021.

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. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *EMNLP*, 2023.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Halil Alperen Gozeten, M. Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. *arXiv preprint arXiv:2505.23648*, 2025a.

Halil Alperen Gozeten, M. Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning, 2025b. URL `https://arxiv.org/abs/2505.23648`.

Aaron Grattafiori, … (191 authors elided), and Zhiyu Ma. The Llama 3 Herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Joschka Haltaufderheide and Robert Ranisch. The ethics of ChatGPT in medicine and healthcare: a systematic review on large language models. *npj Digital Medicine*, 7(1), July 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL `https://arxiv.org/abs/2009.03300`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL `https://arxiv.org/abs/1904.09751`.

Jie Huang, Shikun Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models cannot self-correct reasoning yet. 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL `https://arxiv.org/abs/2401.04088`.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. 2022. URL `https://arxiv.org/abs/2207.05221`.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, 2023.

Aman Madaan and Amir Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*, 2022.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=S37hOerQLB`.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models, 2025. URL `https://arxiv.org/abs/2410.05229`.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *EMNLP*, 2025.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *International Conference on Learning Representations*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

José I. Orlicki. Beyond words: A latent memory approach to internal reasoning in LLMs. *arXiv preprint arXiv:2502.21030*, 2025.

Jacob Pfau, William Merrill, and Samuel R Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Salber, Yang Liu, and Balaji Lakshminarayanan. Out-of-distribution detection and selective generation for conditional language models. In *International Conference on Learning Representations*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. Prompting GPT-3 to be reliable. In *International Conference on Learning Representations*, 2023.

Wenhui Tan, Yunlong Li, Zihan Wang, Yuxuan Liu, Zhuoyi Chen, and Xiangyu Xu. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. *arXiv preprint arXiv:2505.16552*, 2025.

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

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 2024.

Neeraj Varshney and Chitta Baral. Post-abstention: Towards reliably re-attempting the abstained instances in QA. *arXiv preprint arXiv:2305.01812*, 2023.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. Stitch: Self-referential transformer for conformal prediction. *arXiv preprint arXiv:2210.01478*, 2022.

Mingyang Wang, Lukas Lange, Heike Adel, Yunpu Ma, Jannik Strötgen, and Hinrich Schuetze. Language mixing in reasoning language models: Patterns, impact, and internal causes. In *EMNLP*, November 2025. URL `https://aclanthology.org/2025.emnlp-main.132/`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. The art of abstention: Selective prediction and error regularization for natural language processing. In *ACL-IJCNLP*, 2021.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, 2024.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *ACL*, pp. 23336–23351, 2025a.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2025b. URL `https://arxiv.org/abs/2401.11817`.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2023.

Fan Yin, Xinyu Shi, Zhengtao Chang, Jinheon Kim, and Hannaneh Hajishirzi. Large language models can self-improve at answering ambiguous questions. In *EMNLP*, 2023.

Jason Zhang and Scott W Viteri. Uncovering latent chain of thought vectors in large language models. In *Workshop on Neural Network Weights as a New Data Modality*, 2025. URL `https://openreview.net/forum?id=ICuIdJzBPm`.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. *arXiv preprint arXiv:2505.12514*, 2025a.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought, 2025b. URL `https://arxiv.org/abs/2505.12514`.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, Zhenhe Wu, Yongchi Zhao, Tianyu Liu, Jian Yang, Wangchunshu Zhou, Chujie Zheng, Chongxuan Li, Yuyin Zhou, Zhoujun Li, Zhaoxiang Zhang, Jiaheng Liu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A survey on latent reasoning, 2025c. URL `https://arxiv.org/abs/2507.06203`.

A Main Results

		Question Count	Correct	No Answer	Incorrect	Accuracy (%)
Qwen2.5-7B	Baseline (No majority)	1000	639	0	261	71.00
	Minimal Majority (2/5)	54	23	0	31	42.59
	Moderate Majority (3/5)	122	76	0	46	62.30
	Other	31	2	0	29	6.45
	Split Vote (2/2)	22	12	0	10	54.55
	Strong Majority (4/5)	157	122	0	35	77.71
	Unanimous (5/5)	514	469	0	45	91.25
Llama-3.1-8B	Baseline (No majority)	1000	507	0	293	63.38
	Minimal Majority (2/5)	69	30	0	39	43.48
	Moderate Majority (3/5)	159	109	0	50	68.55
	Other	31	1	0	30	3.23
	Split Vote (2/2)	34	10	0	24	29.41
	Strong Majority (4/5)	196	155	0	41	79.08
	Unanimous (5/5)	311	291	0	20	93.57
Mixtral-8x7B	Baseline (No majority)	1000	695	0	305	69.50
	Minimal Majority (2/5)	58	26	0	32	44.83
	Moderate Majority (3/5)	135	90	0	45	66.67
	Other	35	3	0	32	8.57
	Split Vote (2/2)	27	14	0	13	51.85
	Strong Majority (4/5)	175	142	0	33	81.14
	Unanimous (5/5)	470	437	0	33	92.98
gpt-oss-20B	Baseline (No majority)	1000	735	0	265	73.50
	Minimal Majority (2/5)	50	25	0	25	50.00
	Moderate Majority (3/5)	110	79	0	31	71.82
	Other	28	4	0	24	14.29
	Split Vote (2/2)	20	11	0	9	55.00
	Strong Majority (4/5)	160	135	0	25	84.38
	Unanimous (5/5)	532	506	0	26	95.11
DeepSeek-Qwen	Baseline (No majority)	1000	496	0	304	62.00
	Minimal Majority (2/5)	142	37	0	105	26.06
	Moderate Majority (3/5)	125	86	0	39	68.80
	Other	66	13	0	53	19.70
	Split Vote (2/2)	43	10	0	33	23.26
	Strong Majority (4/5)	110	92	0	18	83.64
	Unanimous (5/5)	314	299	0	15	95.22

Table 2: GSM8K Results.

		Question Count	Correct	No Answer	Incorrect	Accuracy (%)
Qwen2.5-7B	Baseline (No majority)	1000	666	0	234	74.00
	Minimal Majority (2/5)	82	41	0	41	50.00
	Moderate Majority (3/5)	141	94	0	47	66.67
	Other	40	3	0	37	7.50
	Split Vote (2/2)	31	10	0	21	32.26
	Strong Majority (4/5)	170	155	0	15	91.18
	Unanimous (5/5)	436	421	0	15	96.56
Llama-3.1-8B	Baseline (No majority)	1000	657	0	343	65.70
	Minimal Majority (2/5)	104	64	0	40	61.54
	Moderate Majority (3/5)	237	155	0	82	65.40
	Other	41	10	0	31	24.39
	Split Vote (2/2)	42	13	0	29	30.95
	Strong Majority (4/5)	273	244	0	29	89.38
	Unanimous (5/5)	303	279	0	24	92.08
Mixtral-8x7B	Baseline (No majority)	1000	720	0	280	72.00
	Minimal Majority (2/5)	70	38	0	32	54.29
	Moderate Majority (3/5)	165	115	0	50	69.70
	Other	42	5	0	37	11.90
	Split Vote (2/2)	33	12	0	21	36.36
	Strong Majority (4/5)	210	188	0	22	89.52
	Unanimous (5/5)	380	362	0	18	95.26
gpt-oss-20B	Baseline (No majority)	1000	760	0	240	76.00
	Minimal Majority (2/5)	60	35	0	25	58.33
	Moderate Majority (3/5)	130	98	0	32	75.38
	Other	35	7	0	28	20.00
	Split Vote (2/2)	25	12	0	13	48.00
	Strong Majority (4/5)	190	175	0	15	92.11
	Unanimous (5/5)	460	444	0	16	96.52
DeepSeek-Qwen	Baseline (No majority)	1000	645	0	355	64.50
	Minimal Majority (2/5)	155	58	0	97	37.42
	Moderate Majority (3/5)	145	102	0	43	70.34
	Other	72	18	0	54	25.00
	Split Vote (2/2)	48	14	0	34	29.17
	Strong Majority (4/5)	135	122	0	13	90.37
	Unanimous (5/5)	345	333	0	12	96.52

Table 3: GSM-Symbolic Results.

		Question Count	Correct	No Answer	Incorrect	Accuracy (%)
Qwen2.5-7B	Baseline (No majority)	1000	586	0	314	65.11
	Minimal Majority (2/5)	12	2	0	10	16.67
	Moderate Majority (3/5)	181	94	0	87	51.93
	Split Vote (2/2)	47	10	0	37	21.28
	Strong Majority (4/5)	205	132	0	73	64.39
	Unanimous (5/5)	455	397	0	58	87.25
Llama-3.1-8B	Baseline (No majority)	1000	483	0	417	53.67
	Minimal Majority (2/5)	41	16	0	25	39.02
	Moderate Majority (3/5)	285	141	0	144	49.47
	Other	1	0	0	1	0.00
	Split Vote (2/2)	102	29	0	73	28.43
	Strong Majority (4/5)	228	173	0	55	75.88
	Unanimous (5/5)	243	208	0	35	85.60
Mixtral-8x7B	Baseline (No majority)	900	429	0	471	47.67
	Minimal Majority (2/5)	27	12	0	15	44.44
	Moderate Majority (3/5)	291	128	0	163	43.99
	Split Vote (2/2)	105	27	0	78	25.71
	Strong Majority (4/5)	246	145	0	101	58.94
	Unanimous (5/5)	231	179	0	52	77.49
gpt-oss-20B	Baseline (No majority)	1000	368	0	632	36.80
	Minimal Majority (2/5)	12	4	0	8	33.33
	Moderate Majority (3/5)	180	65	0	115	36.11
	Other	6	2	0	4	33.33
	Split Vote (2/2)	52	19	0	33	36.54
	Strong Majority (4/5)	220	82	0	138	37.27
	Unanimous (5/5)	430	168	0	262	39.07
DeepSeek-Qwen	Baseline (No majority)	1000	487	0	413	54.11
	Minimal Majority (2/5)	27	8	0	19	29.63
	Moderate Majority (3/5)	246	99	0	147	40.24
	Other	4	0	0	4	0.00
	Split Vote (2/2)	76	35	0	41	46.05
	Strong Majority (4/5)	226	128	0	98	56.64
	Unanimous (5/5)	321	268	0	53	83.49

Table 4: MMLU Results

B Ablations

To validate our key hyperparameter choices, we conducted ablation studies on the two parameters that fundamentally define Noisy Coconut: the noise scale (σ_0) and the number of reasoning paths (K). Experiments were conducted on GSM8K using Qwen2.5-7B-Instruct. Table 5 summarizes our findings.

Table 5: Core ablation study on GSM8K (Qwen2.5-7B-Instruct) where number of questions is equal to 1000. We vary noise scale and path count independently. Baseline uses K=1 with no noise injection.

Configuration	**Accuracy (%)**	Δ
Baseline (K=1, σ_0=0)	71.0	—
Noise scale (fixed K=5):		
σ_0=0.0	76.4	+5.4
σ_0=0.2	78.2	+7.2
σ_0=0.5	77.8	+6.8
Path count (fixed σ_0=0.2):		
K=1	72.0	+1.0
K=5 ($\geq$3/5)	78.2	+7.2
K=10 ($\geq$6/10)	80.2	+9.2

Noise scale. We evaluated $\sigma_0 \in \{0, 0.2, 0.5\}$ with K=5 paths. Setting σ_0=0 (i.e., multiple paths without noise injection) yields 76.4% accuracy (+5.4% over baseline), demonstrating that diversity alone provides substantial improvements through majority voting. However, performance peaks at σ_0=0.2 with 78.2% accuracy, indicating that noise injection provides an additional +1.8% gain beyond path diversity alone. Higher noise scales show logarithmic decay, with σ_0=0.5 achieving 77.8% as individual path quality degrades. We select σ_0=0.2 as it achieves the optimal balance between exploration diversity and reasoning coherence.

Number of paths. We evaluated $K \in \{1, 5, 10\}$ with σ_0=0.2. Performance scales sublinearly with K, showing diminishing returns beyond K=5. While K=10 achieves 2.0 points higher accuracy, it requires 2× the computational cost. We select K=5 as a practical tradeoff between performance and efficiency.

C Language Mixing

Input Question

Question ID: 1 **Reference Answer:** 285

Gabriel is popping popcorn for a snack. As the skillet of kernels heats up, the kernels start popping faster. 30 pop in the first 20 seconds of cooking, then 2 times that amount in the next 20 seconds. The kernels increase to 4 times the initial popping rate in the next 20 seconds, but in the final 20 seconds, the popping slows down to half the rate as the past 20 seconds. After Gabriel takes the skillet off the heat, a quarter of the number of kernels that popped in the final 20 seconds of cooking also pop from the residual heat. How many pieces of popcorn does Gabriel have to eat?

Response A: Monolingual (English)

Initial popping rate is 30 kernels in 20 seconds.
In the next 20 seconds, the popping rate is 2 times the initial rate:

$$2 \times 30 = 60 \text{ kernels}$$

In the next 20 seconds, the popping rate is 4 times the initial rate:

$$4 \times 30 = 120 \text{ kernels}$$

In the final 20 seconds, the popping rate is half of the previous 20 seconds:

$$\frac{1}{2} \times 60 = 30 \text{ kernels}$$

← *Error: Should be half of 120, not 60*

After taking the skillet off the heat, a quarter of the final 20 seconds' popping also pops from the residual heat:

$$\frac{1}{4} \times 30 = 7.5 \text{ kernels}$$

Adding all the popped kernels together:

$$30 + 60 + 120 + 30 + 7.5 = 247.5 \text{ kernels}$$

Since we can't have half a kernel, we'll round down to the nearest whole number:

Final Answer: 247 ✗ Incorrect

Response B: Code-Switched (Chinese + English)

总共有几个爆米花？
首先，我们来计算每个阶段爆米花的产生数量。

1. 第一阶段：前20秒，30个爆米花。
2. 第二阶段：接下来的20秒，爆米花数量是第一阶段的2倍，即 $30 \times 2 = 60$ 个。
3. 第三阶段：接下来的20秒，爆米花数量是第一阶段的4倍，即 $30 \times 4 = 120$ 个。
4. 第四阶段：最后20秒，爆米花数量是第三阶段的0.5倍，即 $120 \times 0.5 = 60$ 个。
   Correct: Half of 120

现在，我们计算在最后20秒的爆米花中，由于残余热量，额外爆开的爆米花数量。这是最后20秒爆米花数量的四分之一，即 $60 \times \frac{1}{4} = 15$ 个。
最后，我们将所有阶段产生的爆米花数量加起来：

$$30 + 60 + 120 + 60 + 15 = 285$$

因此，Gabriel有285个爆米花可以吃。所以，答案是：

Final answer: 285 ✓ Correct

Translation of Code-Switched Response

How many pieces of popcorn are there?
First, let's calculate the popping rate for each step:

1. ***First step****: the first 20 seconds, 30 pieces of popcorn.*
2. ***Second step****: In the next 20 seconds, the number of popped kernels is 2 times the amount from the first step,* i.e. $30 \times 2 = 60$ *kernels.*
3. ***Third step****: In the next 20 seconds, the number of popped kernels is 4 times the amount from the first step,* i.e. $30 \times 4 = 120$ *kernels.*
4. ***Fourth step****: In the last 20 seconds, the number of popped kernels is 0.5 times the amount from the third step,* i.e. $120 \times 0.5 = 60$ *kernels.*

Now, we calculate the number of popped kernels in the last 20 seconds' popping from the residual heat. This is a quarter of the final 20 seconds' popping, meaning that $60 \times \frac{1}{4} = 15$ *kernels.*
Finally, we sum up all the numbers of popped kernels from every step:

$$30 + 60 + 120 + 60 + 15 = 285$$

Thus, Gabriel has 285 pieces of popcorn that he can eat. Therefore, the answer is:

Final answer: 285 ✓ Correct

Figure: Comparison of model outputs demonstrating noise-induced code-switching behavior. At certain noise thresholds, the model exhibits language mixing (Chinese-English) which correlates with improved reasoning accuracy on this mathematical task. Yellow highlighting indicates Chinese text segments in the code-switched response.