Argument Graph Uncertainty: Quantifying Uncertainty from the Logical Structure of Reasoning Chains

Simon Baur
Fraunhofer Heinrich-Hertz-Institut
10587 Berlin, Germany
simon.baur@hhi.fraunhofer.de

Sumire Honda
Fraunhofer Heinrich-Hertz-Institut
10587 Berlin, Germany
sumire.honda@hhi.fraunhofer.de

Wojciech Samek
Fraunhofer Heinrich-Hertz-Institut
10587 Berlin, Germany
wojciech.samek@hhi.fraunhofer.de

Jackie Ma
Fraunhofer Heinrich-Hertz-Institut
10587 Berlin, Germany
jackie.ma@hhi.fraunhofer.de

Abstract

Reliable uncertainty estimation is essential for deploying large language models in high-stakes reasoning tasks, yet most existing approaches rely on expensive sampling strategies or shallow token-level signals that ignore the internal logical structure of model outputs. We introduce Argument Graph Uncertainty, a post-hoc framework for multiple choice question tasks, that estimates model confidence directly from the logical structure of a single chain-of-thought reasoning trace. Our method uses either a lightweight instruction-tuned model or a simple deterministic rule to segment the reasoning chain into propositional units, then applies a NLI model to infer relational edges and construct two complementary argument graphs over these segments. The resulting edge scores are aggregated into a Dirichlet posterior over answer options, whose concentration yields calibrated uncertainty estimates reflecting both local step-to-step coherence and global logical consistency across the reasoning trace. Evaluated on GPQA, MMLU-Pro and TruthfulQA, our method surpasses token-based and consistency-based uncertainty baselines across multiple reasoning models. We further demonstrate through extensive combination analyses that argument graph uncertainty captures an orthogonal signal to existing approaches, making it a complementary component in uncertainty ensembles. Crucially, our framework requires no additional calls to the evaluated model and relies entirely on small, efficient auxiliary models, producing fully interpretable, per-node uncertainty attributions at low computational cost.

1 Introduction

The landscape of artificial intelligence has been fundamentally reshaped by the emergence of reasoning models [22; 15]. Unlike traditional LLMs that rely on immediate token prediction, these models engage in deliberate, multi-step computation before committing to an answer. As these systems move from scientific performance benchmarks to high-stakes deployments and autonomous agentic loops, the need for additional metrics, in particular, robust uncertainty quantification (UQ) becomes critical. A safe deployment in such domains (e.g. medical diagnosis or legal decisions) hinges not only on a model's accuracy, but on its ability to signal when its own internal reasoning process has failed to converge on a stable or logically consistent conclusion [1; 47]. Existing UQ methods for language models fall into two camps, and both are ill-suited to reasoning models. **Consistency-based methods**

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

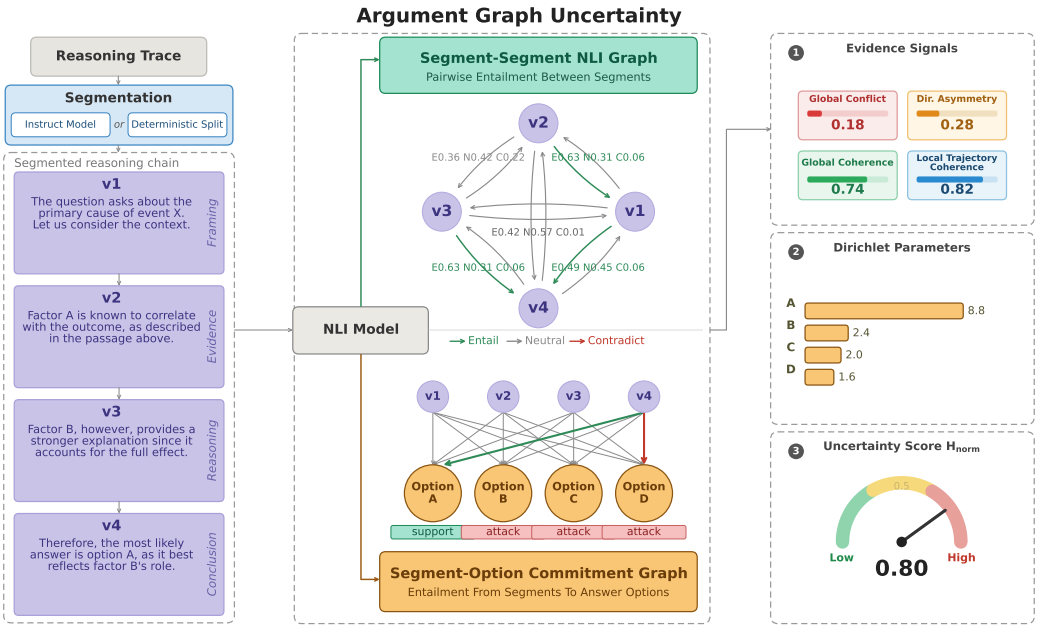

Figure 1: **Argument Graph Uncertainty.** A reasoning trace is segmented into propositional nodes $\{v_1, \ldots, v_4\}$ and passed through an NLI model to construct two complementary graphs: a *Segment–Segment Graph* capturing pairwise entailment, conflict, and asymmetry across all nodes, and a *Segment–Option Commitment Graph* mapping each node to answer options. Graph-derived evidence signals are aggregated into Dirichlet concentration parameters $\{\alpha_o\}$, from which a normalized predictive entropy H_{norm} serves as the final uncertainty score.

sample multiple outputs and measure their agreement [32; 27]. For reasoning models generating thousands of tokens per trace, this multiplies inference cost by an order of magnitude, making the approach impractical in many deployment settings. Moreover, by looking only at the final-answer agreement, these methods discard all information in the reasoning trace itself. **Token-level entropy methods** use token log-probabilities as a confidence proxy [12; 16]. These are computationally less expensive, but the signal degrades precisely where it is needed most [52], as the deeper context mechanically inflates token confidence regardless of the reasoning quality. Simpler heuristics like counting trace length [9], backtracking phrases [28], or peak-entropy tokens [48] seemed to offer an easy and inexpensive way to catch uncertainty signals from reasoning traces, however recent work uncovered major shortcomings of these approaches: Trace length is confounded by problem difficulty, RL-induced length bias [50; 37], overthinking [18], and the apparent signal in overthinking tokens has been traced to training artifacts rather than genuine uncertainty [6]. Token-entropy also cannot separate semantic uncertainty (genuine ambiguity about the answer) from irrelevant lexical variation. Figure 2 gives an intuition of noisy and incoherent behavior token uncertainty can exhibit when confronted with complex tasks. Finally, token- and consistency-based approaches are further compounded by entropy collapse [7; 33; 23]: RL training sharpens output distributions as models exploit high-reward trajectories, making token-level signals increasingly unstable and suppressing the diversity that consistency methods depend on. The core issue is that all of these approaches ignore what actually determines whether a reasoning chain should be trusted: its *logical structure*. What premises are introduced? Do intermediate steps support or contradict one another? Does the conclusion follow from well-grounded inferential chains? None of this is captured by surface statistics or consistency aggregations. Current research in UQ lacks a framework capable of assessing these internal dynamics. To fill this gap, we propose *Argument Graph Uncertainty* (AGU), a post-hoc UQ framework that estimates model confidence from the logical structure of a single chain-of-thought trace for Multiple-Choice-Question-Answering (MCQA). AGU segments the trace into propositional units via a small instruction-tuned model. It then employs a lightweight natural language inference (NLI) model to infer the support and attack relations between these units, constructing two complementary argument graphs that capture global consistency, local trajectory coherence,

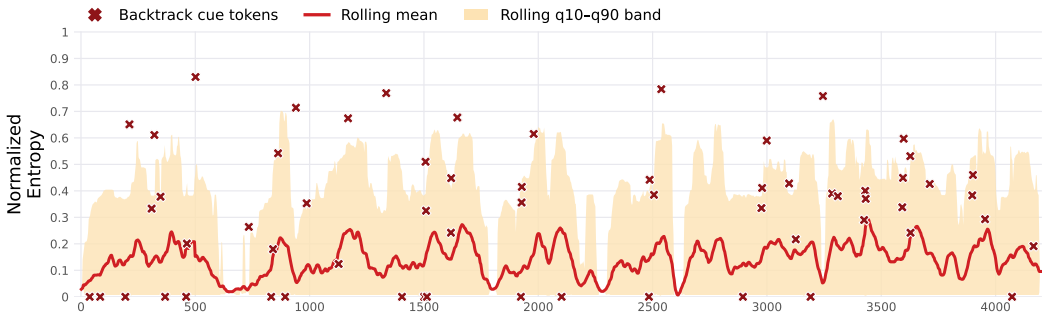

Figure 2: **Token-level entropy across reasoning trace of a single GPQA example.** Normalized per-token entropy (rolling mean, q10–q90 band) and backtracking cue tokens (×) across a single reasoning trace. The vast majority of tokens are generated with high confidence, with intermittent spikes, causing the rolling mean to settle roughly between $\approx 0.1 - \approx 0.2$. Backtracking cues exhibit strongly varying entropy, and do not align with spikes or plateaus. Reasoning traces of correct and incorrect predictions yield similar token entropy trajectories, making it unclear which tokens to select or how to aggregate them into a meaningful uncertainty signal.

and node-to-option evidence. By mapping these structural signals into a Dirichlet posterior, AGU provides a calibrated uncertainty estimate that remains robust even when traditional probabilistic signals fail. Figure 1 provides a graphical overview of individual steps of our proposed method. We show that AGU consistently outperforms token-based and consistency-based baselines across multiple reasoning models for the GPQA, MMLU-PRO and TruthfulQA benchmarks [42; 44; 31]. We further empirically show that AGU captures an orthogonal signal, making it a natural complement to existing methods. Furthermore, it adds only small computational overhead compared to consistency based methods, since it relies entirely on small auxiliary models. Our contributions can be summarized as follows:

- **A Novel Reasoning Trace-Based Framework for UQ:** We propose a new method for uncertainty quantification specifically designed for reasoning models. By exploiting the logical trajectories of reasoning traces, our approach captures structural uncertainty signals that are absent in established methods.
- **Outperforming established Methods:** We demonstrate that our method surpasses both token- and consistency-based uncertainty quantification techniques.
- **Orthogonality of Graph-based UQ signals :** Through detailed empirical combination analysis we show that graph-derived uncertainty serves as an orthogonal signal to consistency based approaches. This allows our method to be integrated with already existing UQ methods, and provide a stronger measure of uncertainty than either single method would on its own.

2 Related Work

Uncertainty Quantification for LLMs Xia et al. [46] categorized existing UQ approaches on LLMs into four: *verbalizing methods*, *latent information methods*, *consistency-based methods*, and *semantic clustering methods*. Verbalizing methods prompt a model to output uncertainty in natural language with the prediction [30]. Although the approach is straightforward and intuitive, instruction models' poor performance in expressing their uncertainty has been observed [49]. Latent information methods use probability distributions on each generated token, which we refer to as *token-level entropy*. They typically estimate uncertainty using specific measures, such as averaged or maximum of negative log-likelihood [35] or entropy of the response distribution in single or multiple responses [41]. Consistency-based methods measure consistency or variance over multiple responses based on similarity [5]. Several metrics are proposed to estimate similarity, from lexical-based metrics such as BLEU [39; 21] to more sophisticated approaches capturing the semantic level, such as SentenceBERT or NLI-based methods [14; 45]. Semantic clustering methods also consider multiple responses, however instead of measuring similarity scores, they cluster the response by their semantic equivalence and calculate the probability over the clusters to estimate uncertainty [27]. They employ

NLI classifier or general LLMs to determine the entailment relationships [27; 13]. Our approach relates to semantic clustering methods, however it differs in applying clustering methods in reasoning traces in terms of the logical relevance to the categorical answer options using argument graphs.

Uncertainty Quantification for Reasoning Models Harsha Tanneru et al. [17] proposed *Chain of Thought Uncertainty*, which measures the agreement between two Chain-of-Thought (CoT) explanations by employing an NLI classifier [19]. They found a strong correlation between confidence in CoT explanations and the correctness of the final responses, highlighting the importance of incorporating CoT into UQ. On the other hand, Agarwal et al. [2] observed surprisingly weak UQ performance especially on large reasoning models, leaving a critical gap in reliable uncertainty estimation for reasoning-intensive tasks. Zhang and Zhang [51] distilled critical information by extracting keywords and assessed their degree of importance to the final answer. Ballon et al. [3] investigated how accuracy evolves over the reasoning trajectory by partitioning CoT traces into deciles and truncating them at each cutoff point. On GPQA Diamond [42] and MMLU-PRO [44], which require multi-step reasoning, they observed a progressive score increase across successive deciles that aligned semantically with reasoning progress, rather than merely reflecting context length or structural form. However, for long-horizon reasoning chains, the issue of inflated uncertainty scores must be addressed [26]; to this end, Mao and Venkat [36] proposed a recurrent propagation mechanism that yields more accurate uncertainty estimates by integrating historical confidence information across steps. Da et al. [8] introduced a method for incorporating structural information from reasoning chains by applying a graph edit distance-based measure.

Abstract Argumentation Frameworks. Our approach is conceptually grounded in the work of Dung [11], which introduced the *Abstract Argumentation Framework*. Dung's framework treats arguments as atomic nodes in a directed graph where edges represent "attack" relations. The "acceptability" of a set of arguments is then determined by graph-theoretic properties such as conflict-freedom and admissibility. While Dung's work focused on formal logic programming and n-person games, it provides the mathematical foundation for identifying consistent subsets of reasoning steps. We adapt this intuition to reasoning chains of LLMs, moving from binary attack relations to probabilistic NLI-based scores that quantify the degree of conflict and support within a generated trace.

NLI Modelling Natural Language Inference is the task of determining whether a hypothesis sentence is entailed, contradicted, or neutral, given a premise [13; 4]. Given the relevance to the logical support and oppositions between propositions, NLI therefore provides a mechanism to infer relational edges, which we employ in our approach. We use DeBERTa [20] as pretrained backbone to infer entailment, neutrality or contradictions that serve as edges in the argument graphs we construct. DeBERTA further enables computationally efficient modeling of dependency between premise-hypothesis pairs. DeBERTa model remains the strongest encoder-based model on NLI benchmarks despite the relatively low computational cost [29].

Evidential Deep Learning and Dirichlet Posteriors. To map our graph-based signals onto a formal uncertainty signal, we adopt a Evidential Deep Learning (EDL) framework. Following Sensoy et al. [43], we treat the parameters of a Dirichlet distribution as evidence for the predicted class probabilities. While EDL has been extensively utilized to quantify epistemic uncertainty and detect out-of-distribution (OOD) samples in computer vision [34] and subjective logic for NLP classification [24], its application to reasoning structures remains unexplored. AGU therefore represents a departure from classical EDL: rather than deriving evidence from raw output logits, we compute it from the structural integrity and logical coherence of the model's argumentation graph.

Multiple-Choice Question Answering Multiple-choice question answering (MCQA) is a natural evaluation paradigm for calibrated uncertainty, as discrete selection over a fixed candidate set mirrors real-world decision problems including agentic tool selection [40], retrieval-augmented re-ranking, and structured clinical triage [25]. The closed-form answer space affords unambiguous accuracy measurement, providing a clean signal for evaluating calibration quality [38; 10; 53]. We adopt GPQA Diamond [42] as our general benchmark. Its difficulty ensures that memorization alone cannot explain performance, as evidenced by the limited accuracy achieved by state-of-the-art systems [42]. We further include MMLU-PRO and TruthfulQA to asses generalizability across broad domains [44; 31].

3 Method

AGU is a post-hoc uncertainty estimation framework that treats a model's chain-of-thought as an *argument graph*. By doing so, it exploits the internal logical structure of a single generated reasoning trace. It decomposes the reasoning chain into discrete propositional segments, constructs two complementary relational graphs over those segments using a NLI scorer, and aggregates the resulting signals into a Dirichlet posterior over answer options whose concentration is then used to infer entropy that serves as uncertainty estimate.

3.1 Reasoning Segmentation

Given a model-generated chain-of-thought R for a multiple-choice question with K answer options $\mathcal{A} = \{a_1, \ldots, a_K\}$, our goal is to arrive at an uncertainty score $U_{\mathrm{AGU}}(R)$. To this end, we partition R into n textual segments $\mathcal{V} = \{v_1, \ldots, v_n\}$ using a light-weight instruct tuned LLM, where each v_i corresponds to a semantically coherent reasoning step. Throughout, $i, j \in \{1, \ldots, n\}$ index segments and $k \in \{1, \ldots, K\}$ indexes answer options.

3.2 NLI Edge Scoring

All relational edges are then scored by a pretrained NLI model f_{NLI}. For a premise p and a hypothesis h, the model returns a probability simplex over three classes:

$$f_{\mathrm{NLI}}(p, h) = \big(p_E(p, h),\ p_N(p, h),\ p_C(p, h)\big), \qquad p_E + p_N + p_C = 1, \tag{1}$$

where E, N, C denote entailment, neutral, and contradiction, respectively. The first argument is always the premise and the second the hypothesis. The NLI model is applied to two types of input pairs: *segment–segment* pairs (v_i, v_j), with segment v_i as premise and segment v_j as hypothesis (Section 3.4), and *segment–option* pairs (v_i, h_k), with segment v_i as premise and the hypothesis h_k = *"The answer is Option X: [option text]"* constructed from answer option a_k as hypothesis (Section 3.3), for which we write $p_E(v_i, h_k)$, $p_N(v_i, h_k)$, $p_C(v_i, h_k)$ analogously.

3.3 Node–Option Graph

For each segment $v_i \in \mathcal{V}$ and each answer option $a_k \in \mathcal{A}$, we define the **contradiction-adjusted support** of node v_i for option a_k as:

$$e_{i,k} = \max\big(0,\ p_E(v_i, h_k) - p_C(v_i, h_k)\big). \tag{2}$$

This quantity is strictly positive only when a node's entailment of an option outweighs its contradiction of that option, i.e. it counts as evidence for an option only if it supports that option more than it contradicts it. Rectifying at zero means a step arguing against an option contributes nothing, rather than cancelling evidence contributed by other steps.

3.4 All-to-All Segment Graph

For every ordered pair (v_i, v_j) with $i \neq j$, we score v_i as the premise and v_j as the hypothesis, yielding $p_E(i, j), p_N(i, j), p_C(i, j)$. This *segment-to-segment* graph captures global logical consistency across the full reasoning chain. We extract two scalar properties from the segment-to-segment graph. We define **global conflict** as the mean pairwise contradiction across all directed edges:

$$\mathrm{Conflict} = \frac{1}{n(n-1)} \sum_{i \neq j} p_C(v_i, v_j). \tag{3}$$

Global conflict measures how much the trace argues against itself, averaged over all scored segment pairs. High conflict indicates that reasoning steps are mutually inconsistent. We define **directional asymmetry** as the mean absolute difference in entailment scores between opposing directions:

$$\mathrm{Asymmetry} = \frac{2}{n(n-1)} \sum_{i<j} \big| p_E(v_i, v_j) - p_E(v_j, v_i) \big|. \tag{4}$$

Directional asymmetry measures whether the entailment between two steps survives swapping premise and hypothesis. High asymmetry indicates that the support between nodes depends on their

order, i.e. that a step follows from another without the reverse holding. Scoring all $n(n-1)$ ordered pairs is quadratic in the number of segments. Alternatively, we can restrict the segment-to-segment graph to pairs within a window of width w along the reasoning chain, $|i-j| \leq w/2$, which reduces the number of scored pairs to $\approx n\,w$. Each segment is thus compared only with its $w/2$ preceding and $w/2$ following segments, in both directions, so the graph captures local rather than global consistency. Conflict, asymmetry and the global coherence (see Section 3.5) are then averaged over the scored pairs instead of all pairs; the remaining steps are unchanged.

3.5 Node Reliability Weights

Each node v_i receives a reliability weight combining both global and local graph signals. The raw coherence (**global coherence**) of node v_i over the all-to-all graph is its mean net entailment across all other nodes:

$$\mathrm{coh}_i^{\mathrm{raw}} = \frac{1}{n-1}\sum_{j\neq i}\big(p_E(v_i,v_j) - p_C(v_i,v_j)\big) \in [-1,1], \quad \text{normalized as} \quad \mathrm{coh}_i = \tfrac{1}{2}(\mathrm{coh}_i^{\mathrm{raw}}+1). \tag{5}$$

We define **local trajectory coherence** as follows: let $\mathcal{T}(i)$ denote the set of adjacent edges incident to v_i, i.e. $(v_{i-1} \to v_i)$ and $(v_i \to v_{i+1})$, each scored with the earlier segment as premise and the later one as hypothesis; the first and last segment have only one such edge. The raw local coherence is:

$$\mathrm{traj}_i^{\mathrm{raw}} = \frac{1}{|\mathcal{T}(i)|}\sum_{j\in\{i-1,\,i\}}\big(p_E(v_j,v_{j+1}) - p_C(v_j,v_{j+1})\big) \in [-1,1], \tag{6}$$

normalized as $\mathrm{traj}_i = \frac{1}{2}(\mathrm{traj}_i^{\mathrm{raw}}+1)$. We then compute a **final node weight** as a weighted sum of global and local coherence:

$$r_i = \lambda\,\mathrm{coh}_i + (1-\lambda)\,\mathrm{traj}_i, \tag{7}$$

where $\lambda \in [0,1]$ is a hyperparameter. Equations (5)–(7) weight each step by how well it fits the trace as a whole (global coherence, Eq. (5)) and its immediate neighbours (local coherence, Eq. (6)): a step is trusted to the extent that it is consistent with both, with λ setting the balance (Eq. (7)).

3.6 Dirichlet Evidence Aggregation

We then aggregate node-level evidence into a per-option signal:

$$E_k = \sum_{i=1}^{n} r_i\,e_{i,k}, \qquad k = 1,\ldots,K. \tag{8}$$

To translate this evidence into a distribution over options, we adopt a Dirichlet model [43]. The effective evidence scale is penalized by the global graph irregularities identified above:

$$\kappa_{\mathrm{eff}} = \kappa\exp(-\gamma_c\,\mathrm{Conflict} - \gamma_a\,\mathrm{Asymmetry}), \tag{9}$$

where κ, γ_c, $\gamma_a > 0$ are hyperparameters. Chains with high internal conflict or directional instability thus receive a globally attenuated evidence scale, reflecting reduced trustworthiness of the reasoning trace as a whole. We can then go on to compute Dirichlet parameters as $\alpha_k = \alpha_0 + \kappa_{\mathrm{eff}}\,E_k,\ k = 1,\ldots,K$, where $\alpha_0 > 0$ is a symmetric prior. The predictive mean over options can be derived as:

$$\hat{p}_k = \frac{\alpha_k}{S}, \qquad S = \sum_{k=1}^{K}\alpha_k. \tag{10}$$

3.7 Uncertainty Quantification

We compute uncertainty estimates from the Dirichlet posterior as Shannon entropy over the predictive mean, normalized to $[0,1]$:

$$U_{\mathrm{AGU}}(R) = H_{\mathrm{norm}} = \frac{-\sum_{k=1}^{K}\hat{p}_k\log\hat{p}_k}{\log K}. \tag{11}$$

High entropy indicates that probability mass is spread broadly across options, while low entropy reflects a concentrated, confident prediction.

4 Experimental Setting and Results

Experimental Setting We evaluate our framework on the aforementioned benchmarks, using nine publicly available reasoning models: DeepSeek-R1-Llama-70B, DeepSeek-R1-Qwen-14B, DeepSeek-R1-Llama-8B, GPT-OSS-120B, GPT-OSS-20B, Nemotron-Nano-30B, Qwen3-30B-Thinking, Qwen3-4B-Thinking, and Phi-4-Mini-Reasoning. All models are run in FP16 precision with sampling temperature 0.7 and top-p 0.9. We report average metrics over n=5 sampled answers and reasoning traces per question. We use Qwen-8B for deriving segments $\mathcal{V}$ from reasoning traces if not stated otherwise. We attach the prompt used for segmentation in Appendix D. For detailed description of how scores and metrics are defined see Appendix A. All experiments reported in Table 1 are conducted with hyperparameters $\alpha_0 = 1.0$, $\kappa = 12.0$, $\gamma_{\text{conflict}} = 1.2$, $\gamma_{\text{asymm}} = 1.0$ and $\lambda = 0.5$. Appendix C.1 offers a thorough analysis on hyperparameter influences. While our chosen default set of parameters is sufficient to achieve good performance, some models could benefit from hyperparameter tuning, as shown in moderate increases of PRR and ρ.

Uncertainty Calibration Table 1 reports uncertainty calibration across all nine reasoning models for GPQA diamond. U_{AGU} achieves the highest PRR in eight out of nine models, demonstrating consistent gains over both U_{Cons} and U_{Token} across model families and scales. The improvements are particularly pronounced for smaller models such as Phi-4 Mini Reasoning and Qwen3 4B Thinking, where token-level uncertainty proves unreliable and consistency-based methods underperform, suggesting that argument-graph structure captures uncertainty signals that are otherwise lost. The single exception is DeepSeek-R1-D.-Llama-70B, where U_{Cons} attains the highest PRR (0.46 ± 0.09). Token uncertainty U_{Token} performs the least reliably overall, with negative PRR values for GPT-OSS 120B and DeepSeek-R1-D.-Llama-70B indicating that token probabilities are miscalibrated to the point of inverting the rejection ordering, in line with the critical stance we took in Section 1. Appendix B.1 provides calibration results for MMLU-Pro and TruthfulQA.

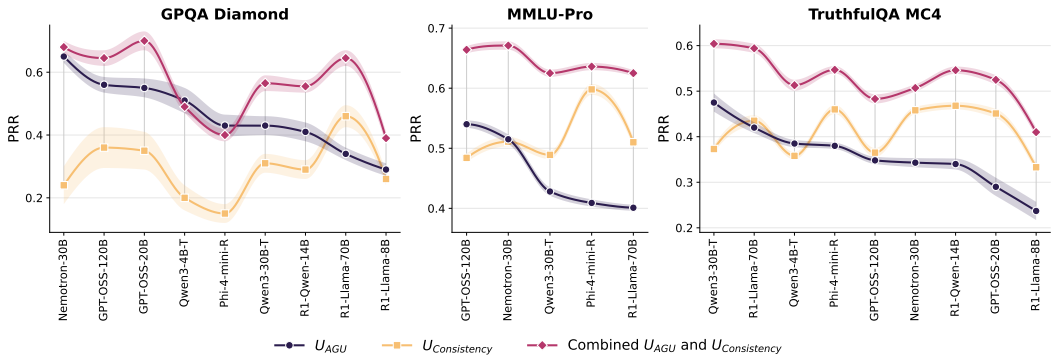

Figure 3: **PRR across all models for U_{AGU}, U_{Cons}, and their equal-weight combination.** The combined score consistently dominates both individual methods.

Table 1: **Calibration performance across nine reasoning models on GPQA Diamond.** We compare Argument Graph Uncertainty $U_{\text{AGU}}(R)$ against consistency-based uncertainty U_{Cons} and token probability uncertainty U_{Token} using Prediction Rejection Rate (PRR ↑) and Spearman correlation (ρ ↓). Bold indicates the best method per model. $U_{\text{AGU}}(R)$ achieves the highest PRR in eight out of nine models.

	Argument Graph Uncertainty $U_{\text{AGU}}(R)$		**U_{Cons}**		**U_{Token}**	
Model	**PRR ↑**	ρ ↓	**PRR ↑**	ρ ↓	**PRR ↑**	ρ ↓
Nemotron-3 Nano 30B	**0.65 ± 0.07**	**-0.37 ± 0.06**	0.24 ± 0.14	-0.24 ± 0.08	0.41 ± 0.15	-0.32 ± 0.08
GPT-OSS 120B	**0.56 ± 0.09**	-0.28 ± 0.06	0.36 ± 0.15	**-0.37 ± 0.08**	-0.27 ± 0.19	0.02 ± 0.08
GPT-OSS 20B	**0.55 ± 0.10**	**-0.37 ± 0.07**	0.35 ± 0.14	-0.36 ± 0.09	0.14 ± 0.18	-0.20 ± 0.09
Qwen3 4B Thinking	**0.51 ± 0.07**	**-0.39 ± 0.06**	0.20 ± 0.09	-0.21 ± 0.07	0.27 ± 0.08	-0.21 ± 0.07
Qwen3 30B Thinking	**0.43 ± 0.09**	-0.27 ± 0.06	0.31 ± 0.11	**-0.32 ± 0.07**	0.10 ± 0.02	-0.10 ± 0.02
Phi-4 Mini Reasoning	**0.43 ± 0.07**	**-0.35 ± 0.06**	0.15 ± 0.08	-0.13 ± 0.07	0.06 ± 0.08	-0.05 ± 0.07
DeepSeek-R1-D.-Qwen-14B	**0.41 ± 0.08**	**-0.32 ± 0.06**	0.29 ± 0.09	-0.28 ± 0.07	0.10 ± 0.10	-0.14 ± 0.07
DeepSeek-R1-D.-Llama-70B	0.34 ± 0.04	-0.24 ± 0.03	**0.46 ± 0.09**	**-0.45 ± 0.06**	-0.39 ± 0.07	0.28 ± 0.06
DeepSeek-R1-D.-Llama-8B	**0.29 ± 0.08**	**-0.27 ± 0.06**	0.26 ± 0.08	-0.22 ± 0.07	-0.13 ± 0.08	0.07 ± 0.07

Figure 3 shows the effect of combining U_{AGU} and U_{Cons} via equal-weight blending. The combined score achieves the highest PRR in every model, and the gains are substantial: for Nemotron-3 Nano 30B the combined score reaches 0.68, compared to 0.65 for U_{AGU} alone and 0.24 for U_{Cons} alone, and for DeepSeek-R1-D.-Llama-70B, the one model where U_{Cons} outperforms U_{AGU} individually, the combination still improves over both. This consistent pattern strongly suggests that the two signals are largely orthogonal: argument-graph structure captures uncertainty arising from internal reasoning conflicts, while consistency-based methods reflect variability across independent samples, and these failure modes do not overlap. Appendix B.6 provides an extensive analysis of alternative combination strategies, offering the insight that basic equal blending is enough to utilize the potential of combining the two uncertainty signals. Furthermore, through visual inspection of distributions of uncertainty score we can get a sense of how well and why they might behave well or ill-calibrated: figure 4 provides an overview over token distributions of different methods for different models. Token uncertainty collapses to near-zero across all models, rendering it uninformative. Consistency scores are strongly bimodal: predictions are either unanimous or maximally uncertain, thereby offering little resolution in between. Argument graph uncertainty, by contrast, spreads broadly across $[0, 1]$, providing a diverse and well-calibrated uncertainty signal.

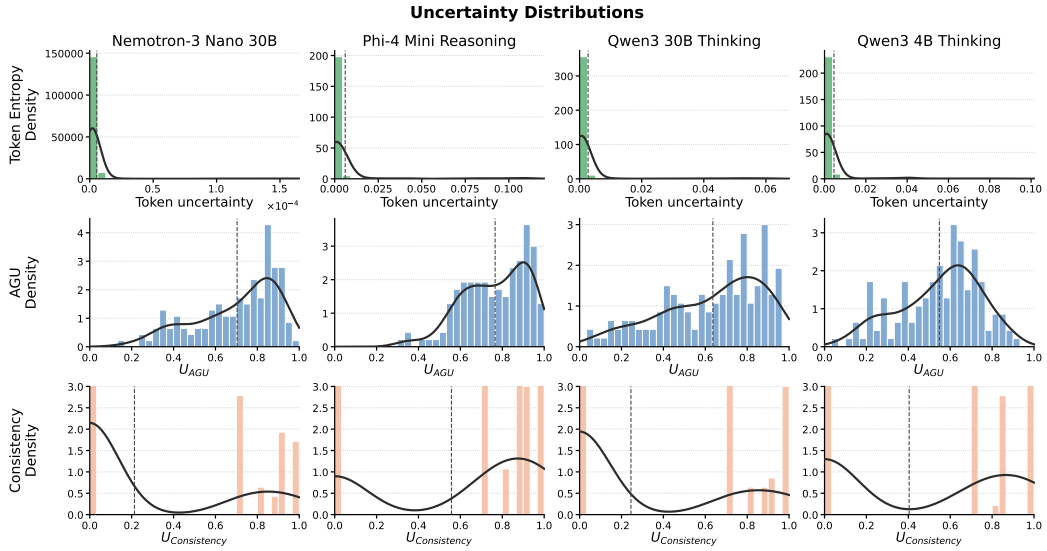

Figure 4: **Uncertainty score distribution examples.** Token entropy (**top**) collapses to near-zero across all models, rendering it uninformative. U_{Cons} (**bottom**) is strongly bimodal (models either agree unanimously or disagree maximally) with little resolution in between. U_{AGU} (**middle**) spreads broadly across $[0, 1]$, capturing diverse uncertainty signals and filling precisely the range that consistency cannot resolve.

Segmentation ablation. We ablate the Qwen3-8B segmenter used in the main experiments against two alternatives: a deterministic split of the trace at paragraph boundaries, which requires no model call, and a larger segmenter, GPT-OSS-120B (Table 2). The deterministic split loses no calibration on average and gains it for several models, with mean PRR 0.47 against 0.46, and improves significantly on GPT-OSS 20B. The larger segmenter yields the highest mean PRR at 0.50 and, notably, cuts the standard deviation from 0.08 to 0.03 on average, so the segmentation stage is the main source of run-to-run variance in U_{AGU}. Since the paragraph split removes the segmentation cost entirely (Eq. 34) at no loss in mean PRR, we consider it the default choice when compute is a concern.

Robustness to NLI errors. Since U_{AGU} is computed from the judgments of a small NLI scorer, we test how errors at the edge level propagate to the uncertainty estimate. We perturb the scorer's outputs on all nine GPQA Diamond models and recompute U_{AGU}, either flipping the predicted class of a random fraction ε of edges or adding Gaussian noise of standard deviation σ to the class log-probabilities (Fig. 5). Degradation is graceful in both cases. With 10% of edges mislabelled, 78% of the clean PRR is retained, and half remains when 40% of all edges carry a wrong class. Noise on

Table 2: **Segmentation ablation on GPQA Diamond.** PRR of U_{AGU} across different segmenters, mean $\pm$ std over seeds. * marks cells that differ significantly from the Qwen3-8B row (Welch's t-test, $p < 0.05$).

Segmenter	Nemotron 30B	GPT-OSS 120B	GPT-OSS 20B	Qwen3 4B-T	Qwen3 30B-T	Phi-4 mini-R	R1-Qwen 14B	R1-Llama 70B	R1-Llama 8B	Mean
Qwen3-8B	$0.65 \pm .07$	$0.56 \pm .09$	$0.55 \pm .10$	$0.51 \pm .07$	$0.43 \pm .09$	$0.43 \pm .07$	$0.41 \pm .08$	$0.34 \pm .04$	$0.29 \pm .08$	0.46
GPT-OSS-120B	$0.67 \pm .07$	$0.68 \pm .03^{*}$	$0.76 \pm .03^{*}$	$0.39 \pm .06^{*}$	$0.47 \pm .02$	$0.32 \pm .03^{*}$	$0.42 \pm .01$	$0.46 \pm .02^{*}$	$0.31 \pm .02$	0.50
Paragraph split	$0.71 \pm .05$	$0.62 \pm .03$	$0.76 \pm .01^{*}$	$0.35 \pm .01^{*}$	$0.46 \pm .01$	$0.29 \pm .03^{*}$	$0.37 \pm .04$	$0.36 \pm .04$	$0.29 \pm .03$	0.47

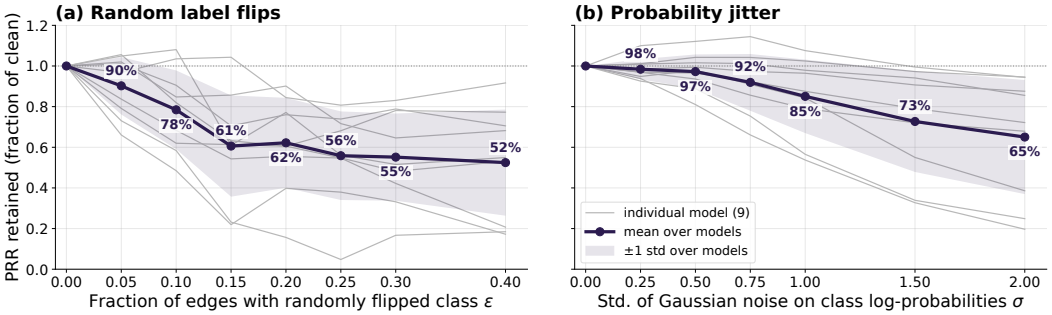

Figure 5: **Robustness of U_{AGU} to NLI errors.** PRR retained on GPQA Diamond when (a) a fraction ε of edge classes is flipped at random and (b) Gaussian noise σ is added to the class log-probabilities. Grey: individual models; bold: mean; band: ± 1 std.

the probabilities costs almost nothing until it exceeds the typical margin between classes, with 97% retained at σ=0.5 and 85% at σ=1. Because the Dirichlet evidence aggregates over many edges, no single misclassification moves the per-option estimate much, and U_{AGU} depends more on the scorer's ordinal judgment rather than on per-edge accuracy or calibration. Appendix B.3 gives the full details. Appendix B.2 provides additional analysis on distribution of edge classification by the NLI-scorer.

All-to-all vs. windowed segment-to-segment graph. As mentioned in 3.4 we can either use the full segment-to-segment (all-to-all) graph, or a windowed subset. We provide an ablation of the all-to-all vs. the windowed graph in Appendix B.4 showing that PRR matches to within 0.01 for a window of $w = 10$ while the cost of the NLI stage reduces from quadratic to linear in the number of segments. Therefore we suggest using the windowed approached as default.

AGU vs. trace length. Since longer traces contain more segments, we verify that U_{AGU} is not simply a proxy for trace length. On GPQA Diamond, the segment count used directly as an uncertainty score attains a mean PRR of 0.33, compared to 0.45 for U_{AGU}, which is the stronger signal on seven of nine models. The two are moderately correlated (mean Spearman 0.43); after rank-regressing the segment count out of U_{AGU}, a PRR of 0.33 remains, indicating that the majority of the graph signal is independent of trace length. See AppendixB.5 for details.

Computational Cost AGU adds no calls to the evaluated model; its overhead runs on an optional segmenter with P_{seg} parameters and an NLI encoder with P_{NLI} parameters. For a trace of L tokens split into segments of average length ℓ, with K answer options and window w,

$$C_{\mathrm{AGU}} = \underbrace{\begin{cases} P_{\mathrm{seg}}\, L \in \mathcal{O}(P_{\mathrm{seg}}\, L) & \text{LLM segmenter} \\ 0 & \text{paragraph split} \end{cases}}_{\text{segmentation}} + \underbrace{\begin{cases} 2(w{+}K)\, P_{\mathrm{NLI}}\, L \in \mathcal{O}(P_{\mathrm{NLI}}\, L) & \text{window } w \\ 2\, P_{\mathrm{NLI}}\, L^2/\ell \in \mathcal{O}(P_{\mathrm{NLI}}\, L^2/\ell) & \text{all-to-all} \end{cases}}_{\text{NLI scoring}} \tag{12}$$

A more detailed explanation and derivation are given in Appendix C. Fig. 6 depicts these costs graphically alongside the consistency baseline and reports measured wall-clock times on GPQA Diamond.

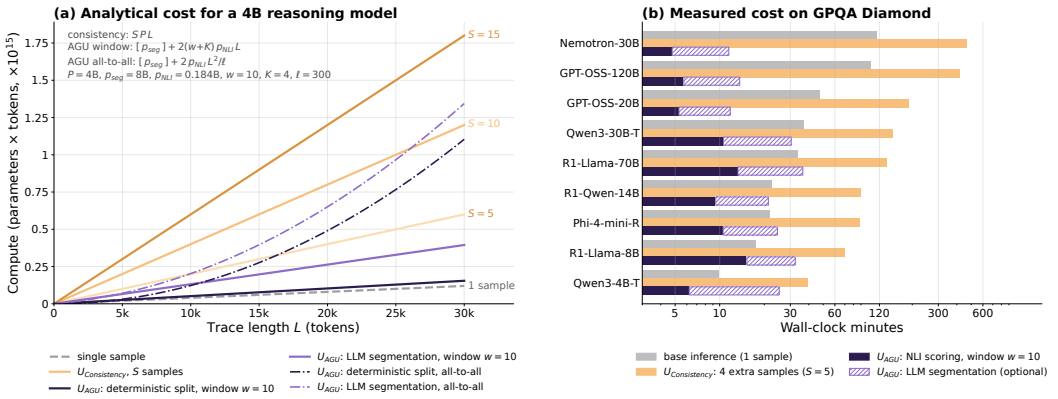

Figure 6: **Compute cost comparison.** (a) Analytical cost against trace length for the consistency and AGU configurations. (b) Measured wall-clock time per model on GPQA Diamond.

5 Discussion

We introduced Argument Graph Uncertainty (AGU), a post-hoc uncertainty quantification framework that estimates model uncertainty from the logical structure of a single chain-of-thought reasoning trace. AGU outperforms both token-level and consistency-based baselines across eight of nine reasoning models on GPQA Diamond, while remaining computationally cheap. Beyond its standalone performance, we find that U_{AGU} captures an orthogonal uncertainty signal to U_{Cons}: a simple equal-weight blend uniformly improves over either method alone on GPQA Diamond, MMLU-Pro and TruthfulQA, even where U_{Cons} is the stronger single signal. This suggests that internal logical coherence and cross-sample output variance probe complementary failure modes. Our ablations show that the framework is robust to its auxiliary components: a deterministic paragraph split matches the LLM segmenter in mean PRR at zero segmentation cost, the windowed segment graph matches the all-to-all graph while reducing the NLI stage from quadratic to linear in the number of segments. We further show that U_{AGU} degrades gracefully under NLI errors, suggesting a robust signal chain. Several directions remain open for future work. First, extending AGU beyond the MCQA setting, for instance to open-ended generation, requires rethinking the node-to-option commitment graph, which currently relies on a fixed answer set. Second, while AGU tolerates edge-level error, the quality of its estimates remains upstream-dependent on the NLI scorer; a scorer trained on reasoning text rather than general-domain NLI could yield further calibration gains.

6 Limitation

AGU depends on the quality of both the segmentation model and the NLI scorer. Although we show that a deterministic paragraph split suffices and that the aggregate tolerates substantial edge-level error, NLI models trained on general-domain corpora can misclassify relations in specialised scientific language, in particular between short numeric fragments, and our robustness analysis only covers synthetic perturbations of a single scorer. Our evaluation is limited to the MCQA setting on three benchmarks; whether AGU generalises to open-ended generation or other reasoning domains remains an open question. Finally, the windowed graph keeps the NLI stage linear in trace length, but very long traces still incur a segmentation cost that scales with the trace when an LLM segmenter is used.

Societal Impact. Reliable uncertainty quantification is a prerequisite for safe deployment of reasoning models in high-stakes domains. By providing interpretable, per-node uncertainty attributions at low computational cost, AGU could lower the barrier to auditing model reasoning in such settings. We note, however, that improved uncertainty estimates could also increase trust in model outputs in contexts where human oversight should be maintained. While we principally see the impact of research in UQ as societally beneficial, we want to emphasize that with the rising deployment of agentic systems, well-calibrated uncertainty estimates should complement rather than replace human oversight.

Acknowledgements

This work was supported by the Senate of Berlin and the European Commision's Digital Europe Programme (DIGITAL) as grant TEF-Health (101100700).

References

[1] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

[2] R. Agarwal, M. Wenzel, S. Baur, J. Zimmer, G. Harvey, and J. Ma. Made: A living benchmark for multi-label text classification with uncertainty quantification of medical device adverse events, 2026. URL `https://arxiv.org/abs/2604.15203`.

[3] M. Ballon, B. Verbeken, V. Ginis, and A. Algaba. Probing the trajectories of reasoning traces in large language models, 2026. URL `https://arxiv.org/abs/2601.23163`.

[4] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In L. Màrquez, C. Callison-Burch, and J. Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL `https://aclanthology.org/D15-1075/`.

[5] J. Cole, M. Zhang, D. Gillick, J. Eisenschlos, B. Dhingra, and J. Eisenstein. Selectively answering ambiguous questions. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.35. URL `https://aclanthology.org/2023.emnlp-main.35/`.

[6] J. Cuesta-Ramirez, S. Beaussant, and M. Mounsif. Large reasoning models are not thinking straight: on the unreliability of thinking trajectories. *arXiv preprint arXiv:2507.00711*, 2025.

[7] G. Cui, Y. Zhang, J. Chen, L. Yuan, Z. Wang, Y. Zuo, H. Li, Y. Fan, H. Chen, W. Chen, Z. Liu, H. Peng, L. Bai, W. Ouyang, Y. Cheng, B. Zhou, and N. Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL `https://arxiv.org/abs/2505.22617`.

[8] L. Da, X. Liu, J. Dai, L. Cheng, Y. Wang, and H. Wei. Understanding the uncertainty of llm explanations: A perspective based on reasoning topology, 2025. URL `https://arxiv.org/abs/2502.17026`.

[9] S. Devic, C. Peale, A. Bradley, S. Williamson, P. Nakkiran, and A. Gollakota. Trace length is a simple uncertainty signal in reasoning models. *arXiv preprint arXiv:2510.10409*, 2025.

[10] J. Duan, J. Diffenderfer, S. Madireddy, T. Chen, B. Kailkhura, and K. Xu. Uprop: Investigating the uncertainty propagation of llms in multi-step agentic decision-making. *arXiv preprint arXiv:2506.17419*, 2025.

[11] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[12] E. Fadeeva, A. Rubashevskii, A. Shelmanov, S. Petrakov, H. Li, H. Mubarak, E. Tsymbalov, G. Kuzmin, A. Panchenko, T. Baldwin, et al. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, 2024.

[13] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[14] X. Gao, J. Zhang, L. Mouatadid, and K. Das. SPUQ: Perturbation-based uncertainty quantification for large language models. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.143. URL `https://aclanthology.org/2024.eacl-long.143/`.

[15] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[16] N. Gupta, H. Narasimhan, W. Jitkrittum, A. S. Rawat, A. K. Menon, and S. Kumar. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*, 2024.

[17] S. Harsha Tanneru, C. Agarwal, and H. Lakkaraju. Quantifying uncertainty in natural language explanations of large language models. In S. Dasgupta, S. Mandt, and Y. Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 1072–1080. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/harsha-tanneru24a.html.

[18] M. Hassid, G. Synnaeve, Y. Adi, and R. Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

[19] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

[20] P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL https://arxiv.org/abs/2006.03654.

[21] Y. Huang, J. Song, Z. Wang, S. Zhao, H. Chen, F. Juefei-Xu, and L. Ma. look before you leap: an exploratory study of uncertainty analysis for large language models. *IEEE Transactions on Software Engineering*, 51(2):413–429, Feb. 2025. ISSN 2326-3881. doi: 10.1109/tse.2024.3519464. URL http://dx.doi.org/10.1109/TSE.2024.3519464.

[22] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[23] R. Jin, P. Gao, Y. Ren, Z. Han, T. Zhang, W. Huang, W. Liu, J. Luan, and D. Xiong. Revisiting entropy in reinforcement learning for large reasoning models. *arXiv preprint arXiv:2511.05993*, 2025.

[24] A. Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016.

[25] Z. Kanjee, B. Crowe, and A. Rodman. Accuracy of a generative artificial intelligence model in a complex diagnostic challenge. *Jama*, 330(1):78–80, 2023.

[26] T. Korbak, M. Balesni, E. Barnes, Y. Bengio, J. Benton, J. Bloom, M. Chen, A. Cooney, A. Dafoe, A. Dragan, S. Emmons, O. Evans, D. Farhi, R. Greenblatt, D. Hendrycks, M. Hobbhahn, E. Hubinger, G. Irving, E. Jenner, D. Kokotajlo, V. Krakovna, S. Legg, D. Lindner, D. Luan, A. Mądry, J. Michael, N. Nanda, D. Orr, J. Pachocki, E. Perez, M. Phuong, F. Roger, J. Saxe, B. Shlegeris, M. Soto, E. Steinberger, J. Wang, W. Zaremba, B. Baker, R. Shah, and V. Mikulik. Chain of thought monitorability: A new and fragile opportunity for ai safety, 2025. URL https://arxiv.org/abs/2507.11473.

[27] L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

[28] A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[29] M. Laurer, W. van Atteveldt, A. Casas, and K. Welbers. Building efficient universal classifiers with natural language inference, 2024. URL https://arxiv.org/abs/2312.17543.

[30] S. Lin, J. Hilton, and O. Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

[31] S. Lin, J. Hilton, and O. Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252, 2022.

[32] Z. Lin, S. Trivedi, and J. Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

[33] X. Liu, T. Chen, L. Da, C. Chen, Z. Lin, and H. Wei. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117, 2025.

[34] A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

[35] P. Manakul, A. Liusie, and M. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

[36] Z. Mao and A. Venkat. Recurrent confidence chain: Temporal-aware uncertainty quantification in large language models, 2026. URL https://arxiv.org/abs/2601.13368.

[37] D. Nohara, T. Nakamura, and R. Yokota. On the optimal reasoning length for rl-trained language models. *arXiv preprint arXiv:2602.09591*, 2026.

[38] C. Oh, S. Park, T. E. Kim, J. Li, W. Li, S. Yeh, X. Du, H. Hassani, P. Bogdan, D. Song, et al. Uncertainty quantification in llm agents: Foundations, emerging challenges, and opportunities. In *Agentic AI in the Wild: From Hallucinations to Reliable Autonomy*.

[39] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In P. Isabelle, E. Charniak, and D. Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[40] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024.

[41] G. Portillo Wightman, A. Delucia, and M. Dredze. Strength in numbers: Estimating confidence of large language models by prompt agreement. In A. Ovalle, K.-W. Chang, N. Mehrabi, Y. Pruksachatkun, A. Galystan, J. Dhamala, A. Verma, T. Cao, A. Kumar, and R. Gupta, editors, *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.trustnlp-1.28. URL https://aclanthology.org/2023.trustnlp-1.28/.

[42] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[43] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

[44] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[45] Z. Wang and C. Holmes. On subjective uncertainty quantification and calibration in natural language generation, 2024. URL https://arxiv.org/abs/2406.05213.

[46] Z. Xia, J. Xu, Y. Zhang, and H. Liu. A survey of uncertainty estimation methods on large language models. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1101. URL https://aclanthology.org/2025.findings-acl.1101/.

[47] Z. Xia, J. Xu, Y. Zhang, and H. Liu. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, 2025.

[48] M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

[49] G. Yona, R. Aharoni, and M. Geva. Can large language models faithfully express their intrinsic uncertainty in words? In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7764, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.443. URL https://aclanthology.org/2024.emnlp-main.443/.

[50] D. Yuan, T. Xie, S. Huang, Z. Gong, H. Zhang, C. Luo, F. Wei, and D. Zhao. Efficient rl training for reasoning models via length-aware optimization. *arXiv preprint arXiv:2505.12284*, 2025.

[51] B. Zhang and R. Zhang. Cot-uq: Improving response-wise uncertainty quantification in llms with chain-of-thought, 2025. URL https://arxiv.org/abs/2502.17214.

[52] X. Zhao. Entropy trajectory shape predicts llm reasoning reliability: A diagnostic study of uncertainty dynamics in chain-of-thought. *arXiv preprint arXiv:2603.18940*, 2026.

[53] H. Zhou and H. Y. Chan. Orch: many analyses, one merge-a deterministic multi-agent orchestrator for discrete-choice reasoning with ema-guided routing. *arXiv preprint arXiv:2602.01797*, 2026.

A Definitions of Metrics and Uncertainty Scores

A.1 Metrics

A.1.1 Prediction Rejection Ratio (PRR).

PRR measures how well an uncertainty score u can be used to selectively abstain from answering. Given N samples with correctness indicators $\mathbf{c} = (c_1, \ldots, c_N)$ and uncertainty scores for different methods $\mathbf{U}(\mathbf{Method}) = (u_1, \ldots, u_N)$, we define a *rejection curve* as follows. For each rejection fraction $r \in [0, 1)$, the $\lfloor rN \rfloor$ samples with the *highest* uncertainty are discarded and accuracy is computed on the remaining $N - \lfloor rN \rfloor$ samples:

$$\mathrm{Acc}(r;\, \pi) = \frac{1}{N - \lfloor rN \rfloor} \sum_{i \in \mathcal{K}(r,\pi)} c_i, \tag{13}$$

where π is an ordering of samples and $\mathcal{K}(r, \pi)$ denotes the indices of retained samples after rejecting the first $\lfloor rN \rfloor$ elements of π. Three orderings are considered:

- **Uncertainty** (π_u): sort by u_i descending (most uncertain first).
- **Oracle** (π_o): sort by c_i ascending (incorrect answers first); this is the best achievable ordering.
- **Random** (π_{rand}): uniformly random ordering, averaged over $T = 200$ independent trials.

The area under each rejection curve is

$$\mathrm{AUC}(\pi) = \int_0^1 \mathrm{Acc}(r;\, \pi)\, \mathrm{d}r, \tag{14}$$

approximated numerically with step size $\Delta r = 0.05$. PRR normalises the uncertainty AUC relative to the oracle and random baselines:

$$\mathrm{PRR} = \frac{\mathrm{AUC}(\pi_u) - \mathrm{AUC}(\pi_{\mathrm{rand}})}{\mathrm{AUC}(\pi_o) - \mathrm{AUC}(\pi_{\mathrm{rand}})}. \tag{15}$$

$\mathrm{PRR} = 1$ corresponds to perfect oracle rejection; $\mathrm{PRR} = 0$ corresponds to random rejection; negative values indicate that the uncertainty score is anti-correlated with error.

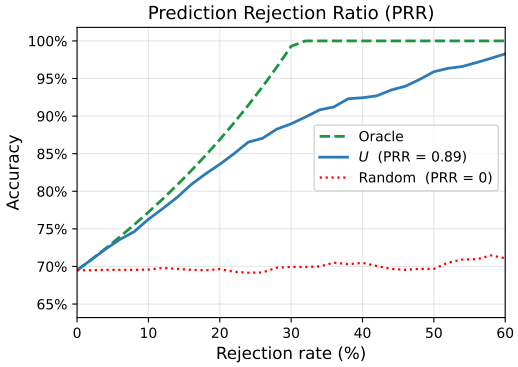

Figure 7: Example visualization of PRR.

A.1.2 Spearman Rank Correlation.

To measure the monotonic relationship between an uncertainty score $u_i \in \mathbb{R}$ and binary correctness $c_i \in \{0, 1\}$ for a set of N questions, we compute the Spearman rank correlation

$$\rho = \frac{\sum_{i=1}^{N} (r_i - \bar{r})(s_i - \bar{s})}{\sqrt{\sum_{i=1}^{N} (r_i - \bar{r})^2 \;\sum_{i=1}^{N} (s_i - \bar{s})^2}}, \tag{16}$$

where $r_i = \mathrm{rank}(u_i)$ and $s_i = \mathrm{rank}(c_i)$ are the average-tied ranks of the uncertainty scores and correctness labels, respectively, and $\bar{r}$, $\bar{s}$ denote their means. A negative ρ indicates that higher uncertainty is associated with lower correctness, which is the desired behaviour for a well-calibrated uncertainty estimate.

A.2 Uncertainty Scores

A.2.1 Argument Graph Uncertainty.

$U_{\mathrm{AGU}} = H_{\mathrm{norm}}$ as defined in Section 3.

A.2.2 Consistency Uncertainty.

U_{cons} measures disagreement across S independent runs of the same model on the same question. Let c_k be the number of runs that predicted option k, and $\hat{p}_k = c_k/S$ the empirical vote fraction. The consistency uncertainty is the normalised Shannon entropy of the vote distribution:

$$U_{\mathrm{cons}} = \frac{-\sum_{k=1}^{K} \hat{p}_k \ln \hat{p}_k}{\ln K}, \tag{17}$$

where the sum is taken only over options that received at least one vote. A value of 0 indicates unanimous agreement across runs; a value of 1 indicates maximally uniform disagreement.

A.2.3 Token-Level Uncertainty.

U_{token} is derived from the log-probabilities the model assigns to each answer-option letter at the final answer token position. Let $\ell_k = \log p_\theta(\text{option } k \mid \text{context})$ be the log-probability of option k extracted from the model's top-κ logits. These are converted to a probability distribution via a softmax over the K options:

$$p_k = \frac{\exp(\ell_k)}{\sum_{j=1}^{K} \exp(\ell_j)}. \tag{18}$$

The token uncertainty is then the normalised Shannon entropy of this distribution:

$$U_{\mathrm{token}} = \frac{-\sum_{k=1}^{K} p_k \ln p_k}{\ln K}. \tag{19}$$

B Additional Results

B.1 Additional Benchmarks

Tables 3 and 4 report PRR on MMLU-Pro and TruthfulQA MC4 for the same signals as 1. Among the single signals, U_{AGU} is the best on two of five MMLU-Pro models and two of nine TruthfulQA models, with U_{Cons} leading elsewhere and U_{Token} only once. The equal-weight combination $U_{\mathrm{AGU}}{+}U_{\mathrm{Cons}}$ is the best signal for every model on both benchmarks, improving on the stronger of its two components by 0.10 PRR on average on MMLU-Pro and 0.09 on TruthfulQA. This further empirically supports our finding that U_{AGU} is orthogonal to consistency-based uncertainty.

Table 3: **PRR on MMLU-Pro.** Mean ± std over seeds. Bold marks the best signal per model.

Model	U_{AGU}	U_{Token}	U_{Cons}	$U_{\mathrm{AGU}}{+}U_{\mathrm{Cons}}$
GPT-OSS-120B	0.54 ± .01	0.05 ± .00	0.48 ± .01	**0.66 ± .01**
Nemotron-30B	0.52 ± .01	0.11 ± .01	0.51 ± .00	**0.67 ± .00**
Qwen3-30B-T	0.43 ± .01	0.05 ± .02	0.49 ± .00	**0.63 ± .01**
Phi-4-mini-R	0.41 ± .01	0.07 ± .00	0.60 ± .01	**0.64 ± .00**
R1-Llama-70B	0.40 ± .01	0.04 ± .01	0.51 ± .00	**0.63 ± .01**
Mean	0.46	0.06	0.52	**0.64**

B.2 NLI Edge Statistics

Fig. 8 provides summary statistics for GPQA diamond. Neutral dominates both edge types (65% of segment–segment, 78% of segment–option edges), as consecutive reasoning steps mostly add new content and most steps neither support nor rule out a given option. Segment–option edges are almost never entailment (1.6%) but contradiction in 20% of cases, so the per-option evidence rests on a few confident entailment edges from

Table 4: **PRR on TruthfulQA MC4.** Mean ± std over seeds. Bold marks the best signal per model.

Model	U_{AGU}	U_{Token}	U_{Cons}	$U_{\text{AGU}}+U_{\text{Cons}}$
Qwen3-30B-T	0.48 ± .05	0.13 ± .02	0.37 ± .03	**0.60 ± .03**
R1-Llama-70B	0.42 ± .04	0.33 ± .01	0.44 ± .02	**0.59 ± .02**
Qwen3-4B-T	0.39 ± .04	0.18 ± .04	0.36 ± .03	**0.51 ± .04**
Phi-4-mini-R	0.38 ± .03	0.11 ± .05	0.46 ± .02	**0.55 ± .02**
GPT-OSS-120B	0.35 ± .03	0.11 ± .02	0.37 ± .03	**0.48 ± .03**
Nemotron-30B	0.34 ± .04	0.17 ± .03	0.46 ± .01	**0.51 ± .01**
R1-Qwen-14B	0.34 ± .03	0.49 ± .03	0.47 ± .01	**0.55 ± .02**
GPT-OSS-20B	0.29 ± .06	0.11 ± .03	0.45 ± .03	**0.53 ± .04**
R1-Llama-8B	0.24 ± .05	0.25 ± .02	0.33 ± .02	**0.41 ± .05**
Mean	0.36	0.21	0.41	**0.53**

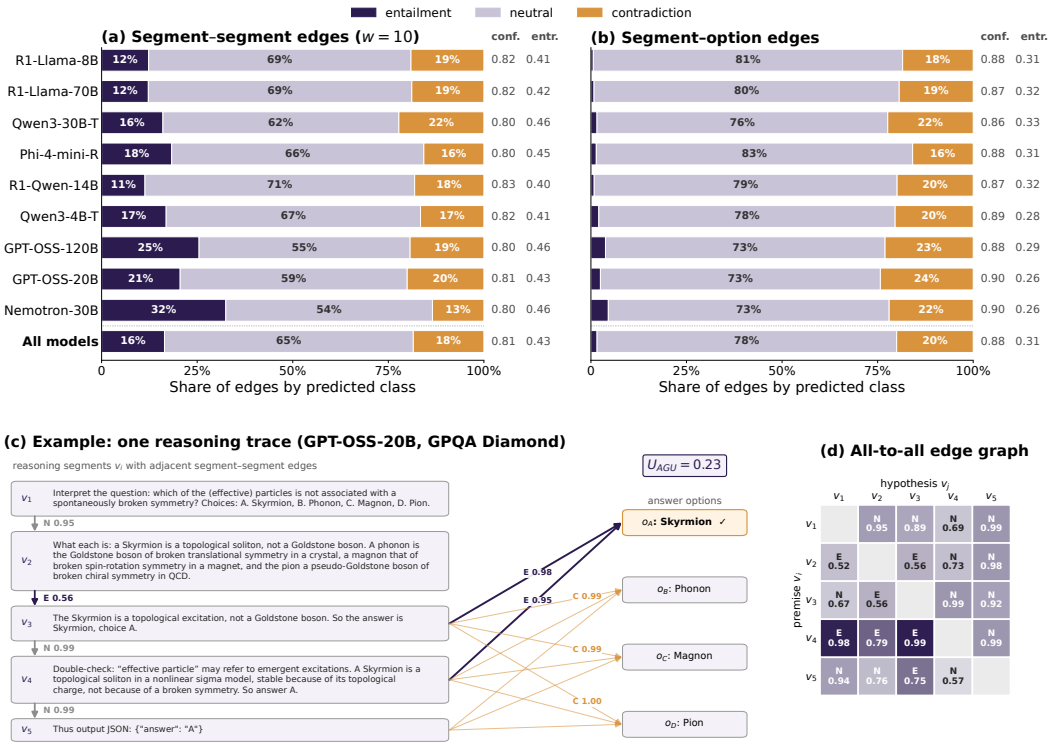

Figure 8: **NLI edge statistics.** (a, b) Share of edges by predicted class per model for segment–segment edges within the window (w=10) and segment–option edges; conf. and entr. are the mean probability of the predicted class and the mean three-class entropy normalised by $\log 3$. (c) A complete five-segment trace of GPT-OSS-20B with the scorer's actual outputs; segment text is abbreviated. Adjacent segment–segment edges are shown along the chain. For visibility, only segment–option edges with $p \geq 0.75$ are drawn as arrows; v_1 and v_2 are neutral towards all options and v_5 towards o_A. (d) All ordered segment–segment edges of the same trace; row is premise, column is hypothesis, shading is the scorer's confidence.

the concluding segments, while contradiction edges rule out the remaining options. The profile is generally stable across models; the segment–segment entailment share can varies notably (11% to 32%). The scorer is more decisive on segment–option edges (confidence 0.88 vs. 0.81, entropy 0.31 vs. 0.43); this is the channel that enters the Dirichlet evidence directly. The example in panels (c, d) shows a correctly answered trace with $U_{\text{AGU}} = 0.23$: the two concluding segments entail the correct option at $p \geq 0.95$ and contradict the others at $p \geq 0.99$, and the double-check segment v_4 entails all preceding steps, giving it the highest coherence weight.

B.3 Analysis of NLI Robustness

We measure how errors of the NLI scorer at the level of individual edges propagate to U_{AGU}. For each of the nine GPQA Diamond models we take the scorer's class probabilities for every segment–segment and segment–option edge, perturb them, and recompute U_{AGU} from the perturbed edges. Each perturbation is applied at several noise levels with three random seeds per level, and we report PRR as a fraction of the PRR obtained from the unperturbed edges, averaged over seeds and models.

Random label flips. With probability ε per edge, the scorer's predicted class is replaced by one of the two other classes chosen uniformly at random, for ε from 0.05 to 0.4. This simulates hard classification errors that are uncorrelated with the content of the edge. PRR decays smoothly and stays positive for every model: 90% of the clean PRR is retained at ε=0.05, 78% at 0.1, 62% at 0.2 and 52% when 40% of all edges carry a wrong label. The decline flattens beyond ε=0.15, since the remaining signal is carried by the majority of edges that are still correct.

Probability jitter. Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$ is added to the class log-probabilities of every edge, which are then renormalised, $\tilde{p} \propto \exp(\log p + \eta)$, for σ from 0.25 to 2. This simulates a miscalibrated scorer: the predicted class changes only where the margin between classes is smaller than the noise. PRR retention is 97% at σ=0.5, 85% at σ=1 and 65% at σ=2, where the noise exceeds the typical margin and a large share of predicted classes change. U_{AGU} thus relies on the scorer's ordinal judgment rather than on calibrated probabilities.

Both quantities that enter the Dirichlet parameters are averages over many edges, so a single misclassified edge shifts the per-option evidence only marginally.

B.4 Windowed vs. All-to-All Segment Graph

Scoring every ordered segment pair is quadratic in the number of segments and could become a bottleneck for long traces. The windowed variant of Section 3.4 scores only pairs with $|i - j| \leq w/2$, so that with $w = 10$ each segment is compared against its five preceding and five following segments in both directions. All segment–option edges are still scored, and Conflict, Asym and the global coherence become averages over the windowed edge set; the local trajectory term is unaffected. We compare the two graphs on GPQA Diamond for all nine models (Table 5). Both columns are computed from the same NLI outputs of a single seed per model, so this is a paired comparison; the absolute PRR values therefore differ from the five-seed means of Table 1.

The window uses 18 times fewer segment–segment pairs while PRR is essentially unchanged, 0.42 against 0.42 on average and at most 0.01 apart for any model. We attribute this to the locality of logical structure in a reasoning trace: a step supports or attacks the steps in its immediate neighbourhood, while distant pairs are predominantly neutral and contribute little signal. The NLI stage can therefore stay linear in the number of segments as traces grow, without affecting the quality of the uncertainty estimate.

Table 5: **Windowed (w=10) vs. all-to-all segment graph on GPQA Diamond.** PRR and number of scored segment–segment pairs, one seed per model.

		PRR		Pairs	
Model	**Mean segs**	**Window**	**All-to-all**	**Window**	**All-to-all**
GPT-OSS-20B	69.4	0.67	0.67	94.9k	1.64M
GPT-OSS-120B	61.8	0.59	0.60	102.3k	1.43M
Nemotron-30B	64.4	0.54	0.53	85.3k	1.02M
Qwen3-30B-T	99.2	0.48	0.47	190.4k	3.51M
R1-Qwen-14B	92.8	0.34	0.33	168.8k	2.86M
Qwen3-4B-T	59.7	0.33	0.33	111.2k	1.48M
R1-Llama-70B	123.7	0.32	0.32	239.1k	4.53M
R1-Llama-8B	146.9	0.28	0.27	274.9k	6.88M
Phi-4-mini-R	98.6	0.28	0.27	189.3k	3.45M
Mean / total	90.7	0.42	0.42	1.46M	26.81M

B.5 Trace Length versus Graph Signal

We test whether U_{AGU} carries information beyond the length of the reasoning trace. For each GPQA Diamond model we compute three uncertainty scores on the same items and evaluate each by PRR: (i) the number of segments n used directly as the score, (ii) U_{AGU}, and (iii) U_{AGU} with the length component removed. For (iii) we convert both U_{AGU} and n to ranks, fit a linear regression of $\mathrm{rank}(U_{\mathrm{AGU}})$ on $\mathrm{rank}(n)$, and use the residuals

as the score; this removes any monotone dependence on length while keeping the remaining ordering of U_{AGU}. We also report the Spearman correlation ρ between U_{AGU} and n.

Segment count alone attains a mean PRR of 0.33 against 0.45 for U_{AGU}, and is the weaker signal on seven of nine models. The two scores are moderately correlated ($\rho = 0.43$ on average). Some correlation is expected, since harder questions elicit both longer traces and more internal conflict, so length and graph structure partly respond to the same underlying difficulty. Correlation does not imply redundancy, however: the residual score retains a PRR of 0.33, so the majority of the graph signal is independent of length. The residual is smallest for GPT-OSS-120B and R1-Llama-70B, the two models with the longest traces, where length accounts for the largest share of the signal.

Table 6: **Trace length versus graph signal on GPQA Diamond.** PRR of the segment count used as an uncertainty score, of U_{AGU}, and of U_{AGU} after rank-regressing out the segment count; ρ is the Spearman correlation between U_{AGU} and segment count. Mean $\pm$ std over seeds; bold marks the better of the two single scores.

Model	**Segment count**	U_{AGU}	$U_{\mathrm{AGU}} \mid$ length	ρ
Nemotron-30B	$0.46 \pm .14$	$\mathbf{0.65 \pm .07}$	$0.45 \pm .11$	$0.52 \pm .07$
GPT-OSS-120B	$\mathbf{0.65 \pm .03}$	$0.56 \pm .09$	$0.28 \pm .02$	$0.59 \pm .01$
GPT-OSS-20B	$0.52 \pm .06$	$\mathbf{0.55 \pm .10}$	$0.49 \pm .08$	$0.45 \pm .07$
Qwen3-4B-T	$0.15 \pm .04$	$\mathbf{0.51 \pm .07}$	$0.29 \pm .05$	$0.36 \pm .07$
Qwen3-30B-T	$0.41 \pm .07$	$\mathbf{0.43 \pm .09}$	$0.38 \pm .05$	$0.54 \pm .04$
Phi-4-mini-R	$0.06 \pm .07$	$\mathbf{0.43 \pm .07}$	$0.32 \pm .08$	$0.29 \pm .07$
R1-Qwen-14B	$0.23 \pm .09$	$\mathbf{0.41 \pm .08}$	$0.31 \pm .06$	$0.29 \pm .06$
R1-Llama-70B	$\mathbf{0.34 \pm .03}$	$\mathbf{0.34 \pm .04}$	$0.22 \pm .05$	$0.46 \pm .04$
R1-Llama-8B	$0.13 \pm .06$	$\mathbf{0.29 \pm .08}$	$0.22 \pm .04$	$0.31 \pm .09$
Mean	0.33	**0.46**	0.33	0.43

B.6 Ablation on Uncertainty Score Combination

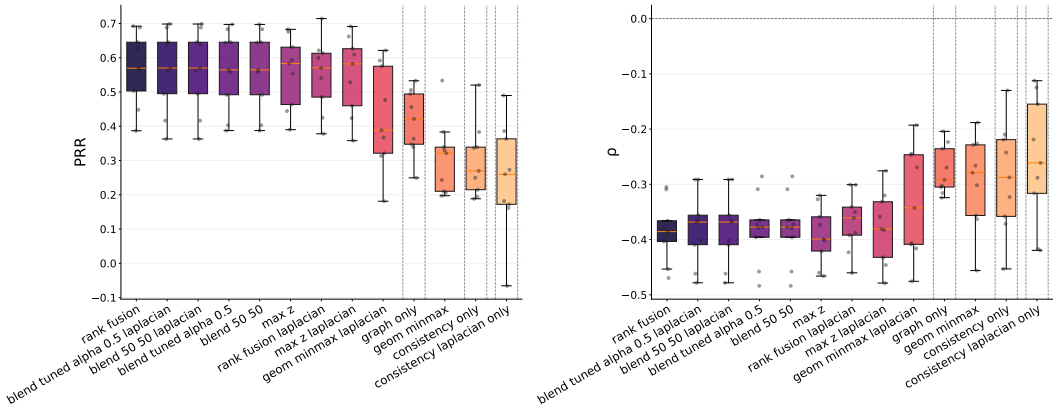

Figure 9: Different uncertainty score combination techniques aggregated over models.

In this section we provide additional experimental results. u_{agu} denotes graph-based uncertainty and u_{cons} denotes a consistency uncertainty signal (either normalized vote entropy or Laplacian-based consistency uncertainty). For each model, combinations are computed at the question level.

Normalization primitives. Given vectors of per-question uncertainties, we define:

$$z_{\mathrm{agu}} = \frac{u_{\mathrm{agu}} - \mu_{\mathrm{agu}}}{\sigma_{\mathrm{agu}}}, \qquad z_{\mathrm{cons}} = \frac{u_{\mathrm{cons}} - \mu_{\mathrm{cons}}}{\sigma_{\mathrm{cons}}}, \tag{20}$$

$$m_{\mathrm{agu}} = \frac{u_{\mathrm{agu}} - \min(u_{\mathrm{agu}})}{\max(u_{\mathrm{agu}}) - \min(u_{\mathrm{agu}})}, \qquad m_{\mathrm{cons}} = \frac{u_{\mathrm{cons}} - \min(u_{\mathrm{cons}})}{\max(u_{\mathrm{cons}}) - \min(u_{\mathrm{cons}})}, \tag{21}$$

$$r_{\mathrm{agu}} = \mathrm{rankpct}(u_{\mathrm{agu}}), \qquad r_{\mathrm{cons}} = \mathrm{rankpct}(u_{\mathrm{cons}}), \tag{22}$$

where $\mathrm{rankpct}(\cdot)$ is the percentile rank in $[0, 1]$. If a denominator is zero (constant signal), the corresponding normalized values are set to 0.

Laplacian consistency uncertainty. Given m sampled predictions (votes) $\{v_i\}_{i=1}^m$, define the agreement matrix

$$W_{ij} = \mathbf{1}[v_i = v_j].$$

Let $d_i = \sum_{j=1}^m W_{ij}$ and $D = \mathrm{diag}(d_1, \ldots, d_m)$. The normalized graph Laplacian is

$$L = I - D^{-1/2} W D^{-1/2}.$$

If $\{\lambda_k\}_{k=1}^m$ are eigenvalues of L, the Laplacian consistency uncertainty is

$$u_{\text{cons}}^{\text{lap}} = \sum_{k=1}^{m} \max(0,\, 1 - \lambda_k).$$

Blending methods. We evaluate the following fused uncertainty scores:

$$\text{Equal-weight z-blend (blend_50_50):} \quad s_{\text{blend50}} = \tfrac{1}{2} z_{\text{agu}} + \tfrac{1}{2} z_{\text{cons}}, \tag{23}$$

$$\text{Tuned-}\alpha\text{ z-blend (blend_tuned_alpha_}\alpha\text{):} \quad s_{\text{blend},\alpha} = \alpha z_{\text{agu}} + (1-\alpha) z_{\text{cons}}, \quad \alpha \in [0,1], \tag{24}$$

$$\text{Rank fusion (rank_fusion):} \quad s_{\text{rank}} = \tfrac{1}{2} r_{\text{agu}} + \tfrac{1}{2} r_{\text{cons}}, \tag{25}$$

$$\text{Max-z fusion (max_z):} \quad s_{\max z} = \max(z_{\text{agu}}, z_{\text{cons}}), \tag{26}$$

$$\text{Geometric min–max fusion (geom_minmax):} \quad s_{\text{geom}} = \sqrt{m_{\text{agu}}\, m_{\text{cons}}}. \tag{27}$$

Tuning of α. For the tuned blend, we search $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$ and select

$$\alpha^\star = \arg\max_{\alpha} \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathrm{PRR}_m(\alpha), \tag{28}$$

where $\mathrm{PRR}_m(\alpha)$ is the PRR of model m using $s_{\text{blend},\alpha}$.

Consistency variants. All formulas above are applied with:

- $u_{\text{cons}} = u_{\text{cons}}$ (as defined in A), and
- $u_{\text{cons}} = u_{\text{cons}}^{\text{lap}}$,

yielding the corresponding "_laplacian" method variants in the latter case in Fig.B.6.

B.7 Uncertainty Score Distributions

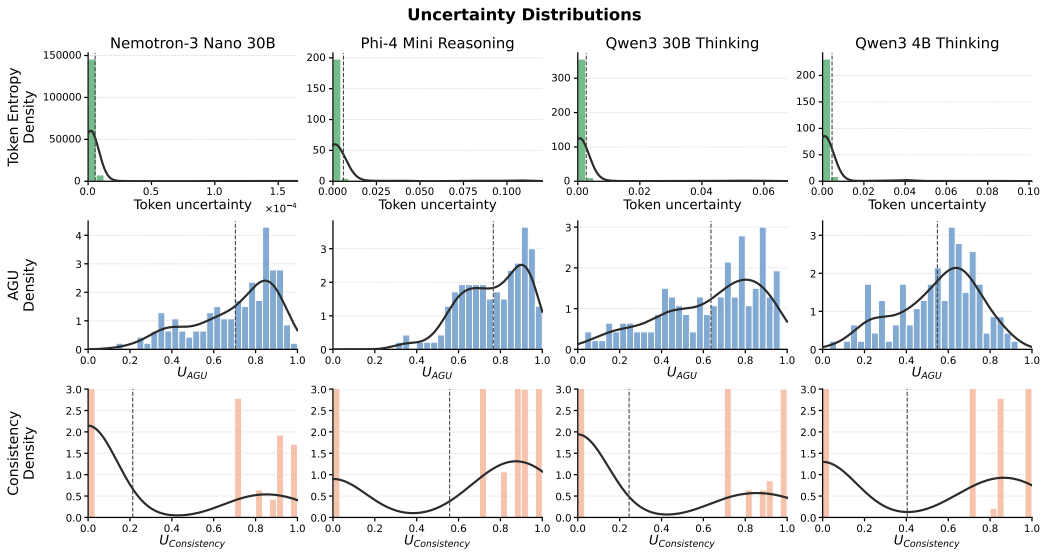

Figure 10: Histograms of uncertainty distributions per model and method.

Figure 10 offers uncertainty scores distributions for the models not shown in the main manuscript. Distributions are consistent across all models.

C Computational Costs

Notation. Let P denote the parameter count of the evaluated reasoning model, L_p the prompt length and L the length of one reasoning trace T in tokens, and s the number of sampled traces used by the consistency baseline ($s = 5$ in our experiments). We measure compute in parameter–token products, which is proportional to FLOPs for a transformer forward pass ($\approx 2P$ FLOPs per token).

Consistency-based uncertainty. Every sample requires a full forward pass of the evaluated model over the prompt and an autoregressive decode of a fresh trace T,

$$C_{\text{sample}} \;=\; P\,(L_p + L) \;\in\; \mathcal{O}(P\,L), \qquad L_p \ll L. \tag{29}$$

Since one trace is generated in any case to obtain the answer, the *overhead* of U_{Cons} is the cost of the $s-1$ additional traces,

$$C_{\text{Cons}} \;=\; (s-1)\,P\,(L_p + L) \;\in\; \mathcal{O}(s\,P\,L). \tag{30}$$

The overhead therefore scales linearly with the size of the evaluated model and with trace length, and is incurred on the evaluated model itself, i.e. at its full GPU footprint.

Argument Graph Uncertainty. AGU operates post hoc on the single trace T that is generated anyway and makes no further calls to the evaluated model. Its overhead is incurred entirely on two small auxiliary models: a segmenter with p_{seg} parameters and an NLI encoder with p_{NLI} parameters. Let T be split into n segments of average length $\ell = L/n$ tokens, and let K denote the number of answer options.

Segmentation. With the instruction-tuned segmenter, T is read once as a prompt and a short list of segment boundaries is emitted,

$$C_{\text{seg}} \;=\; p_{\text{seg}}\,(L + \mathcal{O}(n)) \;\in\; \mathcal{O}(p_{\text{seg}}\,L). \tag{31}$$

With the deterministic paragraph split this stage requires no model call and $C_{\text{seg}} = 0$.

NLI scoring. Every scored pair is one encoder pass over a premise and a hypothesis of $\approx 2\ell$ tokens in total, so the cost is the number of pairs times $2\ell\,p_{\text{NLI}}$. The segment–option graph always contributes nK pairs. The segment–segment graph contributes $n(n-1)$ pairs for the all-to-all variant and $\approx n\,w$ pairs for a window of width w:

$$C_{\text{NLI}}^{\text{window}} = \big(n\,w + n\,K\big)\,2\ell\,p_{\text{NLI}} = 2\,(w+K)\,p_{\text{NLI}}\,L \;\in\; \mathcal{O}\big(p_{\text{NLI}}\,(w{+}K)\,L\big), \tag{32}$$

$$C_{\text{NLI}}^{\text{all}} = \big(n(n-1) + n\,K\big)\,2\ell\,p_{\text{NLI}} \;\approx\; 2\,p_{\text{NLI}}\,\frac{L^2}{\ell} \;\in\; \mathcal{O}\big(p_{\text{NLI}}\,L^2/\ell\big). \tag{33}$$

The windowed cost is linear in L and independent of n: a finer segmentation produces more pairs, but each pair is proportionally shorter. Only the all-to-all graph is quadratic in trace length.

Total overhead and comparison. Combining the two stages gives four configurations,

$$C_{\text{AGU}} \;=\; \underbrace{[\,p_{\text{seg}}\,L\,]}_{\text{LLM segmenter only}} \;+\; \begin{cases} 2\,(w+K)\,p_{\text{NLI}}\,L & \text{window } w, \\ 2\,p_{\text{NLI}}\,L^2/\ell & \text{all-to-all.} \end{cases} \tag{34}$$

For the windowed graph, the ratio to the consistency overhead of Eq. (30) is independent of trace length,

$$\frac{C_{\text{AGU}}}{C_{\text{Cons}}} \;=\; \frac{[\,p_{\text{seg}}\,] + 2\,(w+K)\,p_{\text{NLI}}}{(s-1)\,P} \;\in\; \mathcal{O}\Big(\frac{p_{\text{NLI}}}{P}\Big), \tag{35}$$

and decreases as the evaluated model grows.

C.1 Hyper Parameter Ablation

We conduct a sweep over κ and α_0 with γ_c and γ_a fixed at their default values of $(1.2, 1.0)$, selecting the combination that maximises PRR per model. Table 7 reports the best-found configuration alongside, with bold entries indicating models that improve over the default ($\kappa = 12.0, \alpha_0 = 1.0$) setting.

Five of nine models already achieve their best performance at the default hyperparameters. The remaining four models show moderate gains under tuned configurations, typically favouring a larger κ and smaller α_0, which concentrate the Dirichlet evidence more aggressively and reduce the influence of the symmetric prior.

The sensitivity plots in Figure 11 shows PRR is largely stable across a broad range of κ and α_0 values, with performance degrading only at extreme settings. γ_{conflict} and γ_{asymm} exhibit similarly flat sensitivity curves, indicating that the penalty terms in Equation 9 are not critical to tune precisely. Taken together, these results support the use of a single fixed hyperparameter configuration across all models, as adopted in the main experiments, while careful choice of hyperparameters can lead to moderate improvments in some cases.

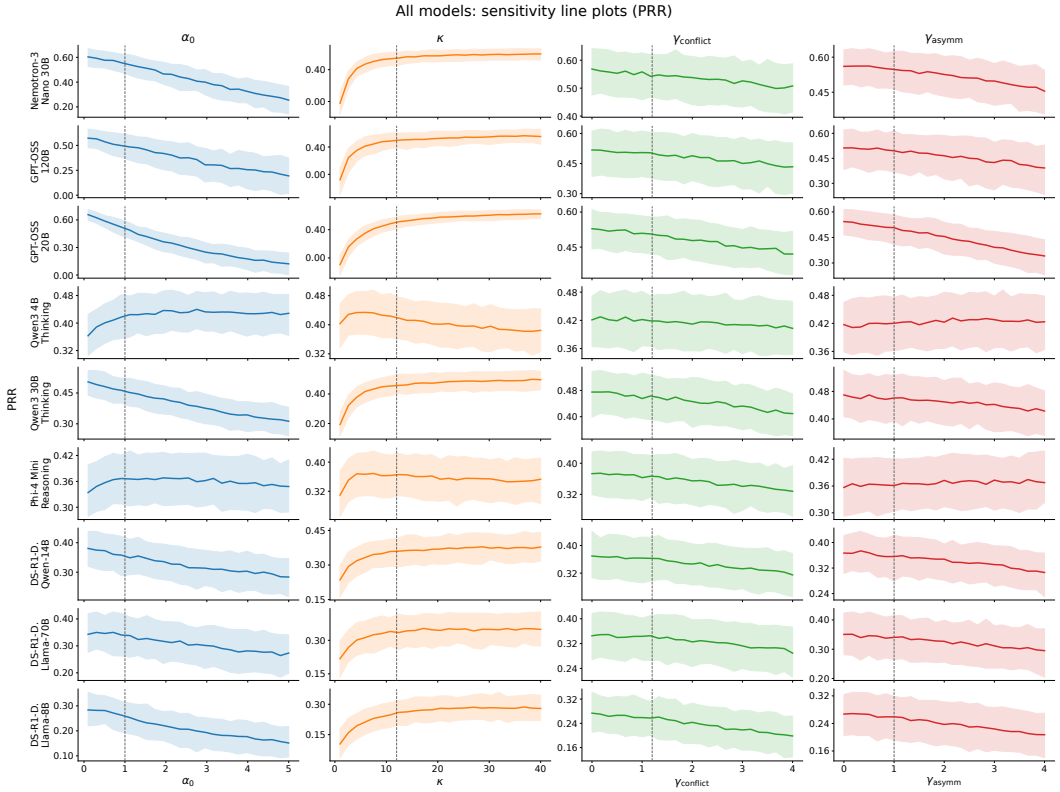

Figure 11: Influence of hyperparameters evaluated for every model.

Table 7: Results of sweep over κ, α_0 and fixed (γ_c, γ_a). Improved models are **bold**.

Model	**PRR** ↑	ρ ↓	(κ, α_0)	(γ_c, γ_a)
GPT-OSS-20B	**0.66**	**-0.39**	**(27.68, 0.10)**	**(1.20, 1.00)**
Nemotron-30B	0.65	-0.37	(12.00, 1.00)	(1.20, 1.00)
GPT-OSS-120B	**0.68**	**0.26**	**(12.00, 1.00)**	**(1.20, 1.00)**
Qwen3-30B-T	**0.51**	**-0.35**	**(37.95, 0.10)**	**(1.20, 1.00)**
Qwen3-4B-T	0.51	-0.39	(12.00, 1.00)	(1.20, 1.00)
Phi-4-mini-R	0.43	-0.35	(12.00, 1.00)	(1.20, 1.00)
R1-Qwen-14B	0.41	-0.32	(12.00, 1.00)	(1.20, 1.00)
R1-Llama-70B	**0.36**	**-0.25**	**(33.84, 0.87)**	**(1.20, 1.00)**
R1-Llama-8B	0.29	-0.27	(12.00, 1.00)	(1.20, 1.00)

D Segmentation Prompt and NLI Scorer Input

In this section we provide the prompt we used to segment reasoning chains into segments $\mathcal{V} = \{v_1, \ldots, v_n\}$ and the design of NLI-Scorer inputs.

Prompt used for indexed segmentation

You segment reasoning traces into logical blocks for argument-graph analysis. The reasoning trace has already been split into numbered units using a rule-based preprocessor. Return only compact segment ranges over those numbered units. Return exactly one valid JSON object with this schema and nothing else:

```
{ "segment_ranges": [[start_unit_index, end_unit_index], ...] }
```

Rules:

1. Do not rewrite, paraphrase, or quote the reasoning text.
2. Each range is 1-based and inclusive, referring to the numbered units below.
3. Ranges must be contiguous, non-overlapping, and cover every unit exactly once.
4. Group consecutive units that serve the same immediate reasoning purpose into one segment.
5. Start a new segment when the reasoning changes function (setup, sub-conclusion, computation, option evaluation, final answer, etc.).
6. Keep closely related false starts or self-corrections together if they resolve the same local issue.
7. If uncertainty resolves into a conclusion used in the next step, start a new segment at that boundary.
8. Keep brief scene-setting with the reasoning move it introduces unless it is clearly its own step.
9. A segment may span multiple units forming one coherent logical move.
10. Use as many segments as the reasoning naturally requires.
11. No markdown, no code fences, no extra keys, no prose outside the JSON.

Reasoning trace to segment: `[1] ... [2] ...`

As mentioned earlier, segmentation by an instruct model can be substituted by a simple deterministic split without losing significant calibration quality. We can simply divide the trace at blank lines (the regular expression `\n\s*\n+`, i.e. two or more consecutive line breaks), yielding one segment per paragraph.

We designed the input to our NLI-Scorer as follows:

Input format for NLI edge scoring

Segment–segment edge (v_i, v_j)**:**

Premise	text of segment v_i
Hypothesis	text of segment v_j

Segment–option edge (v_i, h_k)**:**

Premise	text of segment v_i
Hypothesis	`The answer is Option X: [option text]`

Example (segment v_3 of Fig. 8c against option A):

Premise	`The Skyrmion is a topological excitation, not a Goldstone boson. So the answer is Skyrmion, choice A.`
Hypothesis	`The answer is Option A: Skyrmion`
Output	$(p_E, p_N, p_C) = (0.98,\ 0.02,\ 0.00)$

E Computing Infrastructure

Experiments were run on 2 NVIDIA H200 GPUs.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Section 3, 4

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6, 4

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3, 4

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 3, 4

   Guidelines:

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

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

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

   Answer: [No]

   Justification: The code is not central part of our contribution (e.g. new open-source benchmark).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: 4, Appendix E

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: 5

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our approach does not raise such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Section 4

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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