Distance Is All You Need: Radial Dispersion for Uncertainty Estimation in Large Language Models

Manh Nguyen, Sunil Gupta, Hung Le
Applied Artificial Intelligence Initiative, Deakin University, Australia
{manh.nguyen, sunil.gupta, thai.le}@deakin.edu.au

Abstract

Detecting when large language models (LLMs) are uncertain is critical for building reliable systems, yet existing methods are overly complicated, relying on brittle semantic clustering or internal states. We introduce **Radial Dispersion Score (RDS)**, a simple, parameter-free, fully model-agnostic uncertainty metric that measures the radial dispersion of sampled generations in embedding space. A lightweight probability-weighted variant further incorporates the model's own token probabilities when available, outperforming different nine strong baselines. Moroever, RDS naturally extends to per-sample scoring, enabling applications such as best-of-N selection and confidence-based filtering. Across four challenging free-form QA datasets and multiple LLMs, our metrics achieve state-of-the-art hallucination detection and answer selection performance, while remaining robust and scalable with respect to sample size and embedding choice.

1 Introduction

Large language models (LLMs) exhibit remarkable reasoning and generation capabilities, yet they frequently produce *hallucinations*, fluent but factually incorrect or unsubstantiated outputs [10, 9]. Detecting when an LLM is uncertain remains essential for building reliable, trustworthy systems. Among existing approaches, uncertainty estimation has emerged as one of the most effective tools for identifying when a model is likely to be wrong.

Predictive entropy is the information-theoretic gold standard for quantifying uncertainty [18], but exact computation is infeasible for LLMs due to the exponential output space. As a result, recent work relies on Monte Carlo sampling. Semantic entropy (SE) [13, 5] and its extensions [17, 22, 25] cluster sampled responses in an external embedding space and compute entropy over semantic equivalence classes. While effective, these methods are *unnecessarily overcomplicated* and suffer from three critical drawbacks: (i) accurate semantic clustering is inherently brittle, i.e. a single response can belong to multiple plausible clusters, and (ii) by operating exclusively on the *generation* space, they discard the LLM's own probability estimates and thus ignore a rich source of epistemic uncertainty directly available from the model itself.

A parallel line of research approximates differential entropy through geometric properties of hidden representations. EigenScore [4] uses the trace of the covariance matrix of LLM internal states as a lightweight proxy. While computationally attractive, EigenScore is fundamentally *model-specific* (inaccessible for black-box APIs and even some open-weight models) and assumes isotropic dispersion. In realistic high-uncertainty regimes, especially bimodal or antipodal distributions of plausible completions, most eigenvalues collapse, causing EigenScore to severely underestimate true uncertainty (see Section 4.2).

More recently, PRO [21] approximates predictive entropy using only the top-K generation probabilities derived from negative log-likelihood. Although effective on open-weight models, PRO remains

Preprint. Under review.

model-specific, requires a calibration set to select the optimal K, and cannot be applied to black-box APIs.

We propose **Radial Dispersion Score (RDS)**, a simple, model-agnostic uncertainty metric with a clean geometric interpretation. RDS avoids semantic clustering, does not access hidden states, and requires no calibration or model internals. Given N sampled generations embedded on the unit hypersphere, RDS measures the total ℓ_1 radial dispersion from the empirical centroid:

$$\mathrm{RDS} = \sum_{i=1}^{N} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_1.$$

We further introduce a probability-weighted variant, RDS_w, which incorporates LLM token-level probabilities when available. Both variants scale gracefully with N, rely only on an external sentence encoder, and apply uniformly to both black-box and open-weight models. We also highlight that RDS naturally supports per-sample uncertainty scoring, which is crucial for applications such as best-of-N selection and confidence-based filtering. Unlike semantic-entropy or clustering-based metrics, which are inherently aggregate and cannot be decomposed without heuristics, RDS provides this granularity for free through each sample's radial deviation from the centroid. These per-sample scores are highly effective in practice, consistently improving answer selection accuracy across diverse datasets and models.

In summary, our contributions are threefold:

- We introduce RDS and RDS_w, a family of simple, parameter-free, model-agnostic uncertainty estimators grounded in radial dispersion geometry, with formal connections to differential entropy.
- We show that RDS naturally extends to per-sample uncertainty scoring, enabling powerful applications such as best-of-N selection and confidence-based filtering, which are not supported by most aggregate baselines.
- Across four challenging free-form QA datasets, four diverse LLMs, and both aggregate and per-sample evaluations, RDS and RDS_w deliver state-of-the-art performance and demonstrate strong robustness and scalability.

2 Related Work

Uncertainty estimation methods for LLMs can be grouped into three main families:

Probability-based methods Simple heuristics such as (average) negative log-likelihood remain surprisingly competitive [7, 19]. PRO [21] recently improved upon these by approximating predictive entropy using only the top-K generation probabilities and an adaptive threshold to filter noisy samples. While effective on open-weight models, these approaches cannot be used with black-box APIs and require calibration data.

Semantic entropy and its extensions [13, 5, 17, 22, 25] cluster sampled responses in an external embedding space and compute entropy over clusters' entropy (or density). Despite strong performance, they inherit the brittleness of clustering, are sensitive to the choice of encoder, and discard the LLM's own probability signal.

Geometric methods on hidden states EigenScore [4] uses the trace of the covariance of internal representations as a differential-entropy proxy. It is fast but restricted to models exposing hidden states and systematically underestimates uncertainty when variance is concentrated in few directions (e.g., bimodal distributions).

Our method belongs to none of these families: it is geometric yet model-agnostic, leverages generation probabilities only optionally, and avoids clustering entirely, and is provably more sensitive than trace-based metrics.

3 Preliminaries

From Discrete to Continuous Entropy The information-theoretic gold standard for predictive uncertainty is the conditional entropy of the output distribution:

$$H(Y|x) = -\int p(y|x) \log p(y|x) dy \tag{1}$$

where Y is the output random variable, x is the input. A low predictive entropy indicates a sharply concentrated output distribution, whereas a high value indicates that many possible outputs are comparably likely.

Differential entropy is the continuous analogue of this quantity, obtained by replacing discrete probabilities with a density $f(y|x)$. In the context of LLMs, such a density is induced by embedding sampled completions using either internal hidden states or an external encoder. These embeddings form a smooth point cloud in $\mathbb{R}^d$, which we treat as samples from a continuous distribution. Approximating this distribution as Gaussian $X \sim \mathcal{N}(\mu, \Sigma)$ yields a closed-form differential entropy [31]:

$$H_{\text{de}}(Y|x) = \tfrac{1}{2} \log \det(\Sigma) + \tfrac{d}{2}(\log 2\pi + 1) \tag{2}$$

$$= \tfrac{1}{2} \sum_{i=1}^{d} \log \lambda_i + C, \tag{3}$$

where λ_i are the eigenvalues of Σ, d is the embedding dimension, and C is an additive constant.

EigenScore as a Practical Proxy While differential entropy provides a principled measure, computing it requires estimating the full covariance matrix eigenspectrum. [4] introduce a computationally simple surrogate for differential entropy: the *EigenScore*, defined as the trace of the covariance of the hidden representations:

$$\text{EigenScore}(X) = \text{tr}(\Sigma) \tag{4}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2 \tag{5}$$

$$= \sum_{i=1}^{N} \lambda_i \propto H_{\text{de}}(X), \tag{6}$$

where $\mathbf{x}_i$ denotes the embedding of the i-th sampled completion and $\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$ is the sample mean embedding. EigenScore equals the average squared ℓ_2 distance from the centroid: it approaches 0 when embeddings collapse (low uncertainty) and increases with isotropic spread (high uncertainty).

Generation Probability The probability of a generated sequence $y = (y^1, \ldots, y^T)$ given prompt x factorizes autoregressively:

$$p(y|x) = \prod_{t=1}^{T} p(y^t | y^{<t}, x), \tag{7}$$

where $y^{<t} = (y^1, \ldots, y^{t-1})$ and T is the sequence length. The associated log-probability is:

$$\log p(y|x) = \sum_{t=1}^{T} \log p(y^t | y^{<t}, x). \tag{8}$$

Applying softmax to logits yields token probabilities and the Average Negative Log-Likelihood (ANLL) [19]:

$$\text{ANLL}(y|x) = -\frac{1}{T} \sum_{t=1}^{T} \log p(y^t | y^{<t}, x). \tag{9}$$

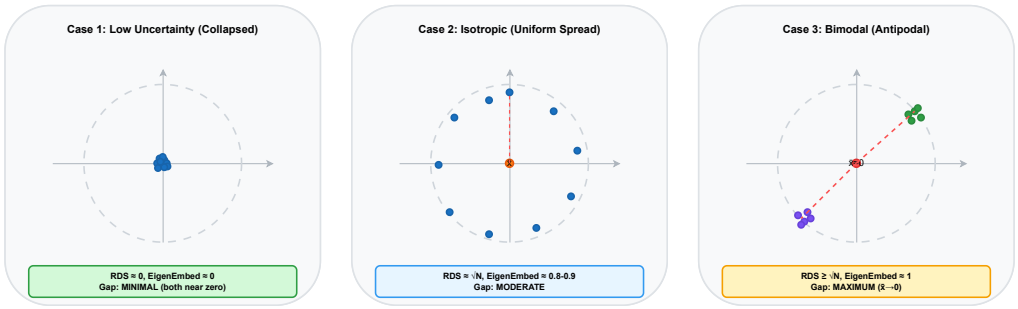

Figure 1: **RDS vs EigenEmbed across three uncertainty regimes.** (1) Collapsed: RDS $\approx$ EigenEmbed ≈ 0. (2) Isotropic: RDS $\approx \sqrt{N}$, EigenEmbed ≈ 0.8–0.9. (3) Bimodal: $\bar{\mathbf{x}} \to \mathbf{0}$, RDS $\geq \sqrt{N}$, EigenEmbed ≈ 1 (maximum gap). RDS is more sensitive to high-uncertainty regimes with semantic diversity.

Using ANLL, the sequence probability can then be estimated post-hoc without additional forward passes via:

$$p(y|x) = e^{-\,\mathrm{ANLL}(y|x)}, \tag{10}$$

making it well-suited for uncertainty scoring.

4 Methodology

4.1 Uncertainty Estimation via Radial Dispersion Score

To estimate the uncertainty of an LLM's output distribution given a prompt x, we leverage the geometry of semantic embeddings sampled from the model's generations. Specifically, we generate N ($N > 1$) sequences $\{y_1, y_2, \ldots, y_N\}$ conditioned on x using multinomial sampling. Given a prompt x, we sample $N > 1$ completions $\{y_1, \ldots, y_N\}$. Each completion is embedded via an encoder $\mathbf{E}$ into unit-normalized vectors $\mathbf{x}_i \in \mathbb{R}^d$:

$$\mathbf{x}_i = \mathbf{E}(y_i), \qquad \|\mathbf{x}_i\|_2 = 1. \tag{11}$$

These embeddings represent points on the unit hypersphere, where proximity reflects semantic similarity among plausible continuations. High uncertainty manifests as dispersed embeddings (diverse plausible outputs), while low uncertainty yields clustered embeddings [14]. We quantify this dispersion simply using the ℓ_1-norm dispersion from the centroid, termed Radial Dispersion Score (RDS), which provides a upper bound on the intrinsic variance captured by the EigenScore (see Section 4.2). This metric is defined as:

$$\mathrm{RDS}(x) = \sum_{i=1}^{N} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_1 , \tag{12}$$

where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ denotes the centroid of the sampled embeddings. Intuitively, RDS measures the total radial dispersion of each embedding from the centroid, with larger values indicating greater dispersion and, consequently, higher uncertainty, vice versa.

Mathematically, the RDS$_w$ can be interpreted as a special case of the 1-Wasserstein distance, where the discrete distribution of embeddings (with weights w_i) is transported to a single-point distribution located at the weighted centroid. While simpler than full OT, this perspective highlights that wRDS measures a form of distributional deviation in embedding space, analogous to optimal transport cost.

We choose the ℓ_1-norm over the ℓ_2-norm for two reasons: (1) it provides a tighter upper bound on variance-based measures (as formalized in Proposition 1), and (2) it is more sensitive to outliers and extreme dispersion, making it particularly effective in high-uncertainty regimes.

Probability-Weighted Variant Geometric dispersion alone overlooks variation in generation likelihoods. Prompt ambiguity, task difficulty, or knowledge gaps can make some outputs far more probable than others [13, 8], while generation probabilities have been shown to correlate with correctness [11]. To incorporate this, we define a *probability-weighted* variant that emphasizes dispersion among high-probability outputs while reducing the impact of low-probability, potentially noisy generations [21]. Let $p_i = p(y_i|x)/\sum_j p(y_j|x)$ denote the normalized generation likelihood of each sequence, the weighted RDS is then defined as:

$$\mathrm{RDS}_{\mathrm{w}}(x) = \sum_{i=1}^{N} p_i \left\|\mathbf{x}_i - \bar{\mathbf{x}}_w\right\|_1, \; \bar{\mathbf{x}}_w = \sum_{i=1}^{N} p_i \mathbf{x}_i. \tag{13}$$

This version emphasizes dispersion among high-probability outputs while reducing the impact of low-probability, potentially noisy generations [21].

Beyond its geometric simplicity, **RDS is inherently robust and broadly applicable**: it relies solely on generated outputs rather than LLM internal states, operates in a lower-dimensional embedding space, and avoids model-specific architectural assumptions. This makes RDS easy to compute, lightweight in practice, and compatible with any black-box LLM. Weighted RDS further refines this signal when generation probabilities are available (e.g., grey-box or open-weight models), yielding a more faithful estimate of uncertainty by prioritizing dispersion among high-probability outputs.

4.2 Connection to EigenScore

In this section, we introduce **EigenEmbed**, defined as EigenScore computed from an *external encoder* $\mathbf{E}$ rather than LLM internal hidden states. This allows fair comparison with RDS, which also operates on external embeddings. We then analyze the relationship between RDS and EigenEmbed.

Proposition 1. Let $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^d$ be unit-norm embeddings ($\|\mathbf{x}_i\|_2 = 1$). Let the centered embeddings be $\mathbf{y}_i = \mathbf{x}_i - \bar{\mathbf{x}}$. Then:

(1) $\mathrm{EigenEmbed} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2 \in [0, 1]$.

(2) $\mathrm{RDS} \geq \mathrm{EigenEmbed}$.

Equality in (2) holds if and only if all $\mathbf{x}_i$ are identical. The gap becomes larger as $\bar{\mathbf{x}} \to \mathbf{0}$.

Proof. See Appendix A.1. □

This lower bound is **tight and parameter-free**, holding for any embedding dimension $d \geq 1$ and sample count $N \geq 2$. When all embeddings coincide ($\mathbf{y}_i = \mathbf{0}\ \forall i$), both RDS and EigenEmbed are zero, indicating model generation is fully certain in this case. As $\bar{\mathbf{x}} \to \mathbf{0}$, which occurs when embeddings are maximally dispersed in opposing directions, EigenEmbed approaches its maximum value of 1, while RDS grows even more rapidly due to the ℓ_1-norm. This widening gap demonstrates that RDS is substantially more sensitive to extreme spread in the embedding space, making it a stronger discriminator in high-uncertainty regimes.

Figure 1 provides geometric intuition in 2D using ten unit-norm embeddings for the superiority of RDS over EigenEmbed. In isotropic or perfectly balanced bimodal settings (top row), both metrics correctly signal high uncertainty. However, the critical case arises when two tight semantic clusters are angularly close on the hypersphere (bottom-right) — a common failure mode in overconfident incorrect generations. Here, the centroid remains near the unit shell, causing almost all variance to collapse into the negligible radial direction; consequently, EigenEmbed drops close to zero and severely underestimates uncertainty. In contrast, RDS continues to capture the meaningful angular separation via the ℓ_1-norm and remains large, correctly reflecting the presence of distinct plausible outputs.

Extremal Case: $\bar{\mathbf{x}} = \mathbf{0}$ When the centroid is zero, the embeddings are placed in a perfectly *balanced* arrangement around the origin of the unit sphere (Figure 1c), and the centroid satisfy

$\mathbf{y}_i = \mathbf{x}_i$. Thus EigenEmbed reached its upper bound due to the total ℓ_2 variance is maximized:

$$\text{EigenEmbed} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_i\|_2^2 = 1.$$

For RDS, non-negativity of $\|\mathbf{y}_i\|_1\ \forall i$ yields

$$\text{RDS} = \sum_{i=1}^{N}\|\mathbf{y}_i\|_1 \ \geq\ \sqrt{\sum_{i=1}^{N}\|\mathbf{y}_i\|_2^2} = \sqrt{N}.$$

The gap between RDS and EigenEmbed in this extremal regime indicate that RDS reacts much more strongly to widespread, highly diverse generations, making RDS particularly sensitive in high-uncertainty regimes where samples diverge in many semantic directions.

Pairwise Cosine Geometry Under $\bar{\mathbf{x}} = \mathbf{0}$ This extremal regime also imposes strict constraints on the pairwise cosine similarities as follows.

Proposition 2. If the embeddings satisfy $\bar{\mathbf{x}} = \mathbf{0}$, then the average pairwise cosine similarity is

$$\frac{1}{N(N-1)}\sum_{i\neq j}\mathbf{x}_i^\top\mathbf{x}_j = -\frac{1}{N-1} < 0. \tag{14}$$

Consequently, for every i, there exists at least one $j \neq i$ such that $\mathbf{x}_i^\top\mathbf{x}_j < 0$.

Proof. See Appendix A.2. □

This negative-average structure is a *geometric signature of semantic disagreement*: the model's plausible continuations are not merely diverse but actively opposed in semantic space, forming at least two *antipodal* clusters, an explicit signature of semantic diversity and uncertainty. Consequently, RDS provides a sharper and more faithful quantification of uncertainty in regimes characterized by maximal semantic disagreement among the generations.

4.3 Per-Sample Uncertainty Metrics

A distinctive advantage of RDS is its natural extensibility to per-sample uncertainty scoring, allowing us to assign an individual uncertainty estimate to each generated sequence y_i, enabling powerful applications such as best-of-N selection and confidence-based filtering. Many existing uncertainty quantification methods (e.g., Semantic Entropy [14], Semantic Density [25], and probability-based baselines [21]) are inherently aggregate metrics and cannot be decomposed to the individual-sample level without additional heuristics. In contrast, RDS supports this granularity effortlessly. We define the per-sample variants as follows:

$$\text{RDS}^{\text{s}}(y_i) = \|\mathbf{x}_i - \bar{\mathbf{x}}\|_1, \tag{15}$$

$$\text{RDS}^{\text{s}}_{\text{w}}(y_i) = \|\mathbf{x}_i - \bar{\mathbf{x}}_w\|_1, \tag{16}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}_w$ are the (weighted) centroids introduced in Eq. (12) and Eq. (13), respectively. These scores measure the ℓ_1-distance of each embedding from the centroid of the sampled distribution. The intuition aligns closely with the principle underlying self-consistency [29]: generations that represent majority or high-probability answers tend to cluster tightly around the centroid and thus receive low per-sample scores, whereas outliers or rare answers are pushed farther away and obtain higher scores. The weighted variant $\text{RDS}^{\text{s}}_{\text{w}}$ is particularly powerful because the centroid $\bar{\mathbf{x}}_w$ is shifted toward high-probability generations. Consequently, low-probability or semantically deviant outputs are penalized more heavily, yielding a per-sample score that correlates better with generation correctness and model confidence. The availability of probability-aware per-sample uncertainty scores further strengthens the practical utility of the RDS family over prior geometric and probabilistic baselines.

Table 1: AUC performance comparison across datasets and models. All values are reported in percentage. Best scores are bolded, second-best values are underlined. Columns **RDS** and **RDS**$_w$ are highlighted with shaded background. ES is unavailable (–) on Gemma2 because the Gemma family does not expose hidden states.

Dataset	Model	ANLL	NLL	PRO	SE	Deg	SD	SC	ES	EE	RDS	wRDS
GPQA	Falcon3	**72.5**	62.2	60.2	50.4	66.3	65.3	51.6	64.9	66.2	67.5	67.0
	Gemma2	62.6	63.6	**65.2**	50.5	63.2	61.9	51.6	–	63.0	63.0	63.7
	Llama3.1	64.1	64.1	59.4	53.3	59.0	58.2	51.6	63.3	62.4	64.4	**66.0**
	Llama3.2	57.2	57.2	63.8	48.2	56.4	54.9	59.8	56.8	63.9	64.3	**67.1**
SciQ	Falcon3	60.0	57.5	62.1	57.0	70.2	69.4	69.1	62.4	73.2	73.4	**74.2**
	Gemma2	59.9	59.3	68.2	59.0	72.5	72.4	74.0	–	74.1	**75.4**	75.2
	Llama3.1	64.4	64.4	57.6	63.7	75.2	73.8	76.5	56.4	77.0	78.4	**78.8**
	Llama3.2	64.2	64.2	54.5	65.1	72.5	71.0	73.2	59.1	73.2	75.1	**75.3**
Arithematics	Falcon3	70.0	76.7	77.0	83.3	**89.9**	89.7	85.4	85.7	83.6	85.3	86.6
	Gemma2	49.6	49.6	56.4	83.4	84.7	84.1	83.2	–	82.6	85.1	**86.3**
	Llama3.1	71.3	71.3	56.8	84.7	87.5	87.9	86.3	64.4	83.6	**88.7**	88.4
	Llama3.2	71.2	71.2	67.7	87.9	87.8	87.9	88.8	58.4	87.6	87.9	**89.0**
SVAMP	Falcon3	91.7	91.7	92.3	89.2	90.9	93.0	94.4	66.6	94.2	94.7	**95.1**
	Gemma2	53.3	47.5	69.1	75.3	83.2	83.8	84.9	–	82.8	83.7	**85.0**
	Llama3.1	57.9	57.9	68.9	84.3	86.7	87.8	**92.4**	64.4	90.4	90.6	91.4
	Llama3.2	62.1	62.1	57.7	80.3	80.0	79.3	86.7	60.8	84.8	84.6	**86.4**
Average		64.5	63.8	64.8	69.7	76.6	76.3	75.6	63.6	77.7	78.9	**79.7**
Best Count		1	0	1	0	1	0	1	0	0	2	**10**

5 Experimental Results

5.1 Experiment Setup

We closely follow the evaluation protocol of recent uncertainty estimation works [5, 25, 21] and focus on free-form (open-ended) question answering.

Datasets We use four established benchmarks covering scientific and mathematical reasoning: (1) Scientific QA: SciQ [30], and GPQA [27], (2) Mathematical reasoning: Arithematics [3], and SVAMP [24].

Models We evaluate on four popular instruction-tuned open-weight models from distinct families: Falcon3-7B [2], Gemma2-9B [20], Llama3.1-8B, and Llama3.2-3B [6]. We refer to them as Falcon3, Gemma2, Llama3.1, and Llama3.2.

Baselines We compare our method against nine established uncertainty estimators: (1) average log likelihood (ANLL) [7], (2) negative log likelihood (NLL) [1], (3) PRO [21], (4) semantic entropy (SE) [13], (5) degree (Deg) [17], (6) semantic density (SD) [25], (7) self consistency (SC) [29] and (8) EigenScore (ES) [4], (9) EigenEmbed (EE), which is EigenScore computed on external embeddings. For the per-sample ranking experiment (Section 5.3), we additionally include Self-Certainty [12].

Evaluation protocol. Following [13, 25, 21], we measure the ability of each uncertainty score to separate correct from incorrect greedy generations using Area Under the ROC Curve (AUC, %). A generation is deemed correct via exact match on reasoning tasks (Arithematics, SVAMP) or ROUGE-L F1 > 0.3 on QA tasks (SciQ, GPQA), exactly as in prior work. To determine whether a generated answer is correct, we use exact matching for reasoning tasks, while following standard practice by computing the F1 score of the ROUGE-L metric [16] between the generation and the ground-truth reference for question-answering tasks. A generation is labeled as correct if this score exceeds 0.3, as in prior works.

Sampling & implementation. We sample $N{=}10$ completions per question using multinomial sampling at temperature $\tau{=}1$ via vLLM [15]. All sampling-based baselines use these same $N{=}10$ generations (except ANLL/NLL, which use only the greedy output). For self-consistency, uncertainty is com-

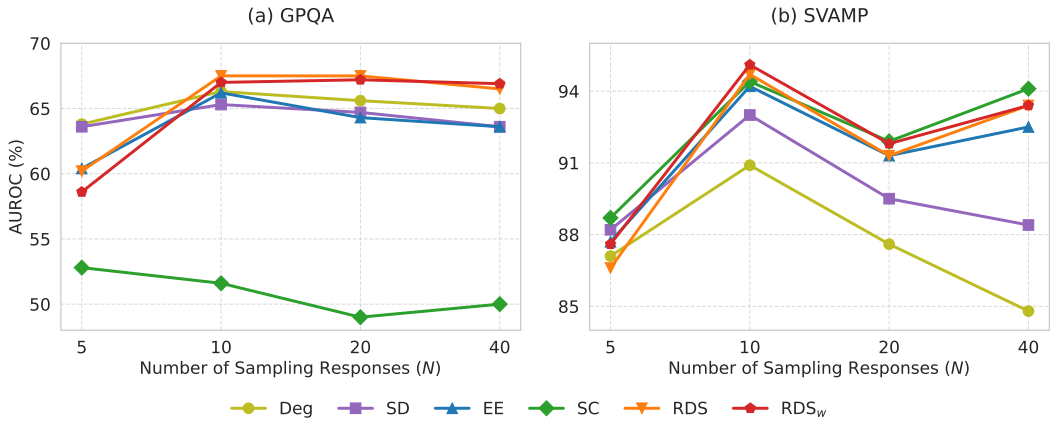

Figure 2: Ablation on the number of sampled responses N. Only top baselines are selected for illustration. Detailed results of all methods are provided in Appendix A.3.

puted as $1-$count of majority answer$/N$. Our RDS uses the widely adopted `all-MiniLM-L6-v2` sentence transformer [26]. All experiments run on a single H100-80GB GPU.

5.2 Main Result

Table 1 summarizes AUC performance across all 16 model–dataset pairs.

RDS and RDS$_w$ consistently outperform all baselines across domains On average, **RDS$_w$** achieves the highest AUC of 79.7%, followed by RDS at 78.9%, surpassing the second-best (EE) by a clear margin of 2.0 and 1.2%, respectively. Notably, RDS$_w$ ranks top-1 in 10 out of 16 settings, demonstrating remarkable robustness across diverse tasks and model architectures.

The superiority of our scores is particularly pronounced on mathematical datasets (ARITH and SVAMP), where the gap over EE widens to 3–5% in several cases. This aligns with our theoretical analysis in Section 4: when generated answers exhibit high lexical diversity (common in free-form math solutions, where responses differing by even a single character or magnitude, e.g., "1" vs. "10", are semantically distant), the proposed RDS metric better captures the underlying semantic clustering than eigenvector-based methods.

Self-Consistency (SC) performs strongly on math problems (frequently ranking in the top-3 on ARITH and SVAMP), but its effectiveness drops considerably on datasets where exact-match evaluation is not applicable (GPQA and SciQ). This confirms that SC remains highly dependent on the availability of a verifiable correct answer.

Other baselines such as SD and Deg show competitive results on specific tasks (mostly QA), but show limited performance on mathematics datasets. Simple probability-based methods (NLL, ALL) generally lag behind, underscoring the importance of modeling response similarity in the embedding space.

Overall, the proposed RDS and weighted variant RDS$_w$ establish a new state-of-the-art for verbalized confidence estimation on open-generation LLMs, delivering consistent gains on both scientific QA and mathematical reasoning while maintaining robustness across model families and dataset characteristics.

5.3 Ranking Performance

Table 2 shows the accuracy when each method is used to rank multiple generations per question and the least uncertain candidate is selected as the final answer. Importantly, not all methods from Table 1 naturally provide an independent per-sample score such as Semantic Entropy, Degree, EigenEmbed, which are inherently set-level and require clustering over all generations of a given question. Only methods that directly output a comparable score for each individual response are evaluated here.

Table 2: Best-of-N selection accuracy (%) using per-sample uncertainty scores (lower uncertain score indicates better sample).

Dataset	Model	ANLL	NLL	SC	RDS	RDS$_w$
GPQA	Falcon3	28.0	24.4	22.8	26.4	**28.5**
	Gemma2	23.8	20.7	19.2	24.4	**25.9**
	Llama3.1	20.7	20.2	17.6	23.8	**26.4**
	Llama3.2	19.7	20.2	16.6	22.8	**23.8**
SciQ	Falcon3	64.3	66.8	67.8	**68.3**	67.3
	Gemma2	73.0	74.0	74.6	**75.1**	74.9
	Llama3.1	62.2	62.7	67.6	67.5	**68.0**
	Llama3.2	52.8	53.5	59.0	**60.0**	58.9
Arithematics	Falcon3	94.6	94.6	**95.0**	**95.0**	**95.0**
	Gemma2	97.8	97.9	**98.0**	**98.0**	**98.0**
	Llama3.1	94.3	**94.6**	94.3	94.2	94.2
	Llama3.2	87.4	87.0	88.0	**88.2**	88.1
SVAMP	Falcon3	71.2	**72.9**	71.2	71.2	71.2
	Gemma2	72.5	71.9	72.2	72.5	**72.9**
	Llama3.1	65.1	65.4	**68.0**	67.5	67.8
	Llama3.2	52.2	53.2	**60.3**	58.0	58.0
Average		61.2	61.3	62.0	63.3	**63.7**

Among these, RDS and its weighted variant achieve the highest average accuracy, outperforming the second-best (SC) by 1.7% and 1.3%, respectively. The advantage is especially striking on the high-difficulty GPQA dataset, where RDS$_w$ yields gains of up to +8.8% over SC on Llama3.1-8B, while outperforms ANLL and NLL 6.2% and 5.7% on the same setting. These results further confirm that our proposed per-sample scoring functions not only improve calibration (AUC in Table 1) but also deliver substantial real-world gains when deployed for answer selection in open-generation settings.

5.4 Ablation Study

In this section, we investigate the effects of hyperparameters including number of sampling response N and choice of embedding model $\mathbf{E}$. To reduce computational overhead, we use Falcon3-7B for experiments over two representative datasets: GPQA and SVAMP.

Effect of the Number of Sampling Responses We vary $N \in \{5, 10, 20, 40\}$ using Falcon3-7B on GPQA and SVAMP (Figure 2). On GPQA (Figure 2a), most semantic-entropy and clustering-based baselines (SE, Deg, SD) peak at N=10–20 and degrade substantially at N=40, indicating that additional low-probability samples introduce noise that harms clustering quality. Self-consistency (SC) performs near random (approximately 50%) when N is large, as majority voting fails without a clearly dominant answer. In contrast, both RDS and RDS$_w$ remain highly robust: performance either continues to improve or plateaus gracefully even at N=40. RDS$_w$ shows essentially no degradation and delivers the highest AUC across the larger the sampling budget. On SVAMP (Figure 2b), the same overall trend holds, though variance is smaller and SC exhibits strong stability at high N, which is expected on mathematics tasks where correct solutions are often repeated verbatim. These results demonstrate that RDS-based metrics are uniquely tolerant to noisy or low-probability generations, making them especially practical when generous sampling budgets are available.

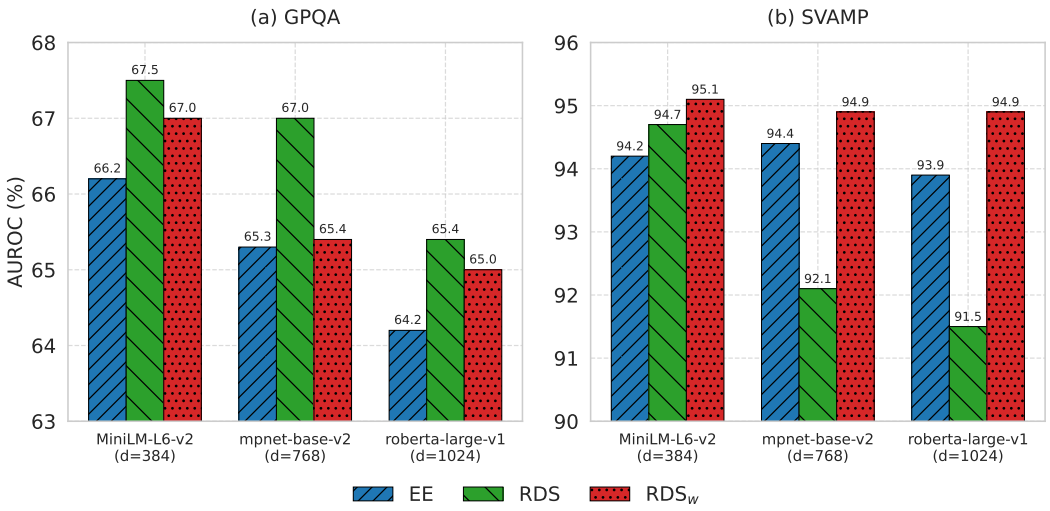

Figure 3: Effect of the sentence embedding model. Results are reported on SVAMP (a) and GPQA (b) using $N{=}10$ sampled responses. Detailed results are provided in Appendix A.4.

Effect Of Embedding Models We compare three widely-used sentence transformers of increasing capacity and embedding dimension: `all-miniLM-L6-v2` (384-d, our default), `all-mpnet-base-v2` (768-d), and `all-roberta-large-v1` (1024-d) in Figure 3. On GPQA, all methods (EigenEmbed, RDS, and RDS$_w$) experience a performance drop when scaling up the encoder. The degradation is about 2% when switching from `all-miniLM-L6-v2` to `all-roberta-large-v1`, consistent with prior observations that larger contrastively trained encoders often struggle with long, synthetic reasoning traces due to representation collapse and domain shift [28, 23]. In constrast, the probability-weighted variant RDS$_w$ is far more stable on SVAMP: its AUC varies by at most $\pm 0.2\%$ across all three encoders. Overall, these results indicate that incorporating generation probabilities through sample-level reweighting noticeably reduces sensitivity to the choice and size of the embedding model, while still preserving most of the performance achieved with the lightweight default encoder.

6 Conclusion

We proposed Radial Dispersion Score (RDS), a simple, parameter-free, and fully model-agnostic uncertainty estimator that directly measures radial dispersion of sampled generations on the unit hypersphere. Its probability-weighted variant optionally incorporates the model's own generation likelihoods. Across four challenging reasoning datasets and four diverse open-weight LLMs, our approach consistently outperforms all previous semantic-entropy, eigenvector-based, and probability-only methods, establishing a new state-of-the-art while proving especially effective for per-sample ranking and best-of-N selection. Crucially, it exhibits strong robustness to both the sampling budget and the choice of sentence embedding model.

Limitations

RDS requires an external sentence encoder. While a lightweight encoder already delivers excellent performance and probability weighting substantially reduces dependence on the specific encoder, heavier models can still introduce modest degradation on the most difficult tasks. In scenarios where any external embedder is undesirable, purely probability-based alternatives remain the only option. Like all sampling-based methods, RDS requires generating multiple completions per input, though robust results are achieved with moderate sampling budgets.

Reproducibility Statement

We have submitted the source code separately for the reviewing process. Upon publication, we will release the implementation as open-source with the necessary instructions to ensure reproducibility.

References

[1] Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking uncertainty estimation in natural language generation. *arXiv preprint arXiv:2412.15176*, 2024.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*, 2024.

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

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

[7] Nuno M Guerreiro, Elena Voita, and André FT Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, 2023.

[8] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In *International Conference on Machine Learning*, pages 19023–19042. PMLR, 2024.

[9] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[10] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

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

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

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

[14] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *NeurIPS ML Safety Workshop*, 2023.

[15] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[16] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

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

[18] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021.

[19] Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, 2023.

[20] Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[21] Manh Nguyen, Sunil Gupta, and Hung Le. Probabilities are all you need: A probability-only approach to uncertainty estimation in large language models. *arXiv preprint arXiv:2511.07694*, 2025.

[22] Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929, 2024.

[23] Dmitry Nikolaev and Sebastian Padó. Representation biases in sentence transformers. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3701–3716, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

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

[25] Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[26] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[27] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[28] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online, June 2021. Association for Computational Linguistics.

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

[30] Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017.

[31] Zhanghao Zhouyin and Ding Liu. Understanding neural networks with logarithm determinant entropy estimator. *arXiv preprint arXiv:2105.03705*, 2021.

A Appendix

A.1 Proof of Proposition 1

Proof. Let $\mathbf{y}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ denote the centered embeddings.

Step 1: EigenEmbed bounds We first compute the squared ℓ_2 norms of the centered embeddings:

$$\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2 = \sum_{i=1}^{N} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2 \tag{17}$$

$$= \sum_{i=1}^{N} \big(\|\mathbf{x}_i\|_2^2 - 2\mathbf{x}_i^\top \bar{\mathbf{x}} + \|\bar{\mathbf{x}}\|_2^2\big). \tag{18}$$

Since $\|\mathbf{x}_i\|_2^2 = 1$ and $\sum_i \mathbf{x}_i = N\bar{\mathbf{x}}$, the middle term simplifies, yielding

$$\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2 = N - 2N\|\bar{\mathbf{x}}\|_2^2 + N\|\bar{\mathbf{x}}\|_2^2 = N\big(1 - \|\bar{\mathbf{x}}\|_2^2\big).$$

Hence, by definition,

$$\mathrm{EigenEmbed} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2 = 1 - \|\bar{\mathbf{x}}\|_2^2 \in [0, 1].$$

Step 2: RDS lower bound By definition, the ℓ_1 norm of each centered embedding is non-negative: $\|\mathbf{y}_i\|_1 \geq 0$ for all i, which implies

$$\sum_{i=1}^{N} \|\mathbf{y}_i\|_1 \geq \sqrt{\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2}. \tag{19}$$

We can rewrite the right-hand side in terms of $\mathrm{EigenEmbed}$:

$$\mathrm{RDS} = \sum_{i=1}^{N} \|\mathbf{y}_i\|_1 \tag{20}$$

$$\geq \sqrt{\sum_{i=1}^{N} \|\mathbf{y}_i\|_2^2} \tag{21}$$

$$= \sqrt{N \cdot \mathrm{EigenEmbed}} \tag{22}$$

$$\geq \sqrt{N} \cdot \mathrm{EigenEmbed} \tag{23}$$

$$\geq \mathrm{EigenEmbed}\,. \tag{24}$$

Eq. (23) and Eq. (24) uses the fact that $\mathrm{EigenEmbed} \in [0, 1]$ (step 1) and $N > 1$, respectively. □

Equality conditions Equality in Eq. (19) occurs if and only if all centered embeddings $\mathbf{y}_i = \mathbf{0}\ \forall i$, this corresponds to all original embeddings $\mathbf{x}_i$ being identical. The final inequality $\sqrt{N \cdot \mathrm{EigenEmbed}} \geq \mathrm{EigenEmbed}$ is strict for $N > 1$, which ensures that $\mathrm{RDS} > \mathrm{EigenEmbed}$ unless the embeddings are identical (EigenEmbed = 0). Intuitively, the gap between RDS and EigenEmbed grows as the mean embedding $\bar{\mathbf{x}} \to \mathbf{0}$, reflecting greater dispersion among the embeddings.

A.2 Proof of Proposition 2

Proof. Compute the squared norm of the total sum:

$$\Big\| \sum_{i=1}^{N} \mathbf{x}_i \Big\|_2^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{x}_i^\top \mathbf{x}_j.$$

If $\bar{\mathbf{x}} = \mathbf{0}$, then $\sum_i \mathbf{x}_i = \mathbf{0}$ and the left-hand side is zero. Expanding the double sum separates diagonal and off-diagonal terms:

$$0 = \sum_{i=1}^{N} \|\mathbf{x}_i\|_2^2 + \sum_{i \neq j} \mathbf{x}_i^\top \mathbf{x}_j = N + \sum_{i \neq j} \mathbf{x}_i^\top \mathbf{x}_j.$$

Hence $\sum_{i \neq j} \mathbf{x}_i^\top \mathbf{x}_j = -N$, and dividing by the $N(N-1)$ off-diagonal entries gives the stated average:

$$\frac{1}{N(N-1)} \sum_{i \neq j} \mathbf{x}_i^\top \mathbf{x}_j = -\frac{1}{N-1} < 0.$$

If for some fixed i we had $\mathbf{x}_i^\top \mathbf{x}_j \geq 0$ for all $j \neq i$, then summing would give

$$\mathbf{x}_i^\top \Big(\sum_{j \neq i} \mathbf{x}_j \Big) \geq 0.$$

But $\sum_{j \neq i} \mathbf{x}_j = -\mathbf{x}_i$ (since the total sum is zero), so the left-hand side equals $-\|\mathbf{x}_i\|_2^2 = -1$, a contradiction. Thus every i has at least one j with $\mathbf{x}_i^\top \mathbf{x}_j < 0$. □

A.3 Extended Results: Number Of Samples

Table 3 and Table 4 report the full metric-wise results on GPQA and SVAMP using Falcon3-7B, which are used to construct Figure 2.

Table 3: Full results on GPQA using Falcon3-7B across different number of sampling responses.

N	PRO	SE	Deg	SD	ES	EE	SC	RDS	RDS$_w$
5	0.547	0.513	0.638	0.636	0.606	0.604	0.528	0.602	0.586
10	0.602	0.504	0.663	0.653	0.649	0.662	0.516	0.675	0.670
20	0.612	0.531	0.656	0.647	0.653	0.643	0.490	0.675	0.672
40	0.603	0.525	0.650	0.636	0.669	0.636	0.500	0.665	0.669

Table 4: Full results on SVAMP using Falcon3-7B across different number of sampling responses.

N	PRO	SE	Deg	SD	ES	EE	SC	RDS	RDS$_w$
5	0.918	0.845	0.871	0.882	0.636	0.877	0.887	0.866	0.876
10	0.923	0.892	0.909	0.930	0.666	0.942	0.944	0.947	0.951
20	0.921	0.862	0.876	0.895	0.693	0.913	0.919	0.913	0.918
40	0.920	0.817	0.848	0.884	0.693	0.925	0.941	0.934	0.934

A.4 Extended Results: Effect Of Embedding Models

Table 5 presents results used to create Figure 3.

Table 5: AUROC scores for EE, RDS, and RDS_w across embedding models.

Model	SVAMP			GPQA		
	EE	RDS	RDS_w	EE	RDS	RDS_w
all-miniLM-L6-v2	0.942	0.947	0.951	0.662	0.675	0.670
all-mpnet-base-v2	0.944	0.921	0.949	0.653	0.670	0.654
all-roberta-large-v1	0.939	0.915	0.949	0.642	0.654	0.650