Latent Self-Consistency for Reliable Majority-Set Selection in Short- and Long-Answer Reasoning

Jungsuk Oh, Jay-Yoon Lee*

Graduate School of Data Science, Seoul National University
{luke0112, lee.jayyoon}@snu.ac.kr

Abstract

Probabilistic decoding in Large Language Models (LLMs) often yields inconsistent outputs, particularly on complex or long-form questions. Self-Consistency (SC) mitigates this for short-form QA by majority voting over exact strings, whereas Universal Self-Consistency (USC) and Weighted Unigram Consistency Score (WUCS) extend to long-form responses but lose accuracy on short-form benchmarks.

We introduce **Latent Self-Consistency (LSC)**, which selects the most semantically consistent response using learnable token embeddings. LSC's lightweight forward processing of summary tokens only introduces negligible runtime overhead (at most 0.9%) on top of standard decoding of the base LLM, and requires no changes to the model architecture.

Across 6 short-form and 5 long-form reasoning benchmarks (e.g., MATH, MMLU, TruthfulQA), LSC surpasses SC, USC, and WUCS on both short-form and long-form on average performance, while adding negligible computational overhead on vanilla inference. These results position LSC as a reliable consistency-selection method that works effectively across various answer formats. Additionally, LSC provides well-calibrated confidence estimates, maintaining low expected calibration error across both answer formats.

Code — https://github.com/jeongseokO/LatentSC_official
Datasets — https://huggingface.co/jeongseokoh
Extended version — https://arxiv.org/abs/2508.18395

Introduction

Large Language Models (LLMs) have achieved remarkable success across diverse natural language processing tasks, from mathematical reasoning to code generation (Brown et al. 2020; Chowdhery et al. 2023; OpenAI 2023). However, as inherently probabilistic generators, they frequently yield divergent outputs on repeated runs, undermining consistency and reliability in downstream applications. To mitigate this variability, Self-Consistency (SC) (Wang et al. 2023) generates multiple candidate responses through independent sampling and applies majority voting to select the most frequent answer. This mechanism has proven highly effective, achieving substantial accuracy improvements in mathematical reasoning and multiple-choice benchmarks by filtering out anomalous reasoning paths.

SC's reliance on exact string matching, however, severely limits its applicability. While effective for "short-answer" tasks such as numerical problems or multiple-choice questions where alternative surface forms are rare, SC fails in "long-answer" scenarios including code generation, summarization, or detailed explanations, where semantically equivalent responses often differ lexically and structurally. To address this limitation, several consistency-based extensions have emerged. Universal Self-Consistency (USC) (Chen et al. 2024) employs the LLM itself as a semantic judge by concatenating all candidates into a single prompt for evaluation; however, this approach incurs substantial computational overhead (approximately 10% additional inference time) and significant memory usage (around 15% additional memory), while introducing potential judge-model biases. Weighted Unigram Consistency Score (WUCS) (Jain et al. 2024) offers a lightweight alternative using term-frequency–weighted Jaccard similarity with probabilistic weighting, but sacrifices accuracy on short-answer tasks and fails to capture semantic coherence in complex reasoning. These alternative methods sometimes achieve competitive accuracy, but such gains only partly arise from putting together truly consistent reasoning paths. For example, USC often relies on LLM reasoning when it selects a more accurate response that disagrees with the majority response.

In practice, this leads to an unsatisfying trade-off: SC performs strongly on short-answer benchmarks but breaks down on long-form tasks, whereas USC and WUCS are better suited to free-form generation yet often degrade performance on short-answer evaluation. Empirically, our experiments (Appendix K) show that, given only the input question, it is difficult to reliably predict whether the model will produce a short-answer–style or a long-answer–style response, making it impractical to route queries to different decoding schemes based on an assumed answer format.

In this work, we introduce *Latent Self-Consistency* (LSC), a framework that aims to achieve the best of both worlds: matching SC's high accuracy on short-answer tasks while extending robustly to long-answer scenarios, with minimal additional computational cost. LSC appends a small set of trainable summary tokens after the EOS token of each gen-

*Corresponding author: lee.jayyoon@snu.ac.kr

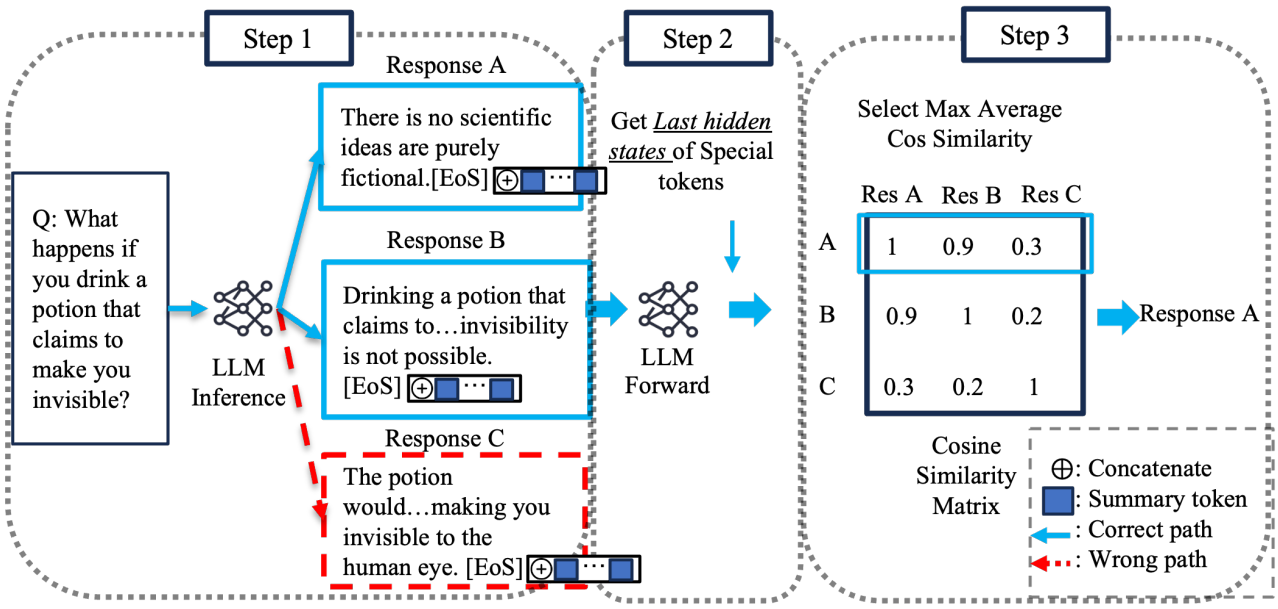

Figure 1: Overview of the Latent Self-Consistency (LSC) inference framework.

erated response and reuses the key–value (KV) cache from the original generation to compute their final-layer representations, yielding compact embeddings that capture the semantic essence of each response. LSC introduces a negligible runtime overhead (at most 0.9%) compared to standard decoding and requires no changes to the model architecture. By computing pairwise cosine similarities with these learned embeddings, LSC identifies the most consistent response without constructing new prompts or performing any complex text processing.

Our contributions are as follows:

- **Universal majority selection across answer formats.** We present the first consistency-based selection method that delivers strong performance on both short- and long-answer tasks while using a single unified approach, with no format-dependent tuning or heuristics.
- **Efficient semantic consistency via learned suffix embeddings.** By encoding response semantics into compact learned embeddings obtained from a few summary tokens, LSC attains USC-level semantic discrimination with substantially reduced runtime and memory overhead, making it practical for real-time deployment.
- **Reliable and calibrated confidence estimates across tasks.** LSC enables uncertainty quantification, achieving competitive expected calibration error (ECE) scores for both short- and long-answer benchmarks.
- **Dynamic boundary detection for robust majority set identification.** We introduce a dynamic Top-K boundary detection algorithm that automatically filters out noisy outliers, further enhancing selection accuracy by focusing on the most semantically coherent subset.

Related Works

Consistency-Based Selection in LLMs

Self-Consistency (SC) (Wang et al. 2023) generates N independent samples and selects the most frequent answer via exact string matching, yielding strong gains on "short answer form" tasks but failing when semantically equivalent outputs vary lexically. Universal Self-Consistency (USC) (Chen et al. 2024) uses the LLM itself as a referee by concatenating all N candidates into a single prompt and performing one additional forward pass. However, this approach requires constructing entirely new prompts, preventing KV cache reuse and introducing substantial latency and potential judge-model bias. Unigram Consistency Score (UCS) and its weighted variant WUCS (Jain et al. 2024) compute probability-weighted overlap measures for text similarity assessment, which while efficient, struggle to capture deep chain-of-thought coherence. Fine-grained Self-Consistency (FSC) (Wang et al. 2024) integrates segment-level consensus by re-generating and synthesizing responses from scratch, incurring extra latency due to these additional generative passes. In contrast, our LSC extracts small, trainable token embeddings by simply appending learnable tokens after the EOS token of each generated response, enabling KV cache reuse for efficient forward passes. This approach unifies SC's accuracy on short-answer tasks with robust, low-latency selection for long-answer tasks.

Prompt Tuning and Learnable Token Embeddings

Existing prompt tuning approaches (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021) typically append learnable tokens to user prompts for domain adaptation without full model fine-tuning. These methods operate in a **pre-generation** manner, modifying the input before the model generates responses.

Models	Methods	Short	Long	Overhead on top of Vanilla	
		Average(%)	Average(%)	Time(%)	Memory(%)
LLaMA3.1-8B-Instruct	SC	72.2	N/A	2E-3	0.0
	WUCS	70.0	63.8	3E-2	0.0
	USC	67.7	64.8	7.4	16.2
	LSC(Ours)	72.3	65.2	0.2	5E-3
	+Dynamic TopK	**72.8**	**65.8**	0.2	5E-3
Qwen3-8B	SC	**77.1**	N/A	1E-3	0.0
	WUCS	74.9	**75.4**	2E-2	0.0
	USC	75.4	74.7	18.2	20.2
	LSC(Ours)	76.8	74.9	0.6	5E-3
	+Dynamic TopK	76.5	74.9	0.6	5E-3
LLaMA3.3-70B-Instruct	SC	**82.7**	N/A	1E-4	0.0
	WUCS	81.1	76.7	8E-3	0.0
	USC	82.5	76.8	9.7	7.5
	LSC(Ours)	82.6	**77.6**	0.9	1E-3
	+Dynamic TopK	**82.7**	**77.6**	0.9	1E-3

Table 1: Overall performance–efficiency trade-off of LSC. Time/Memory columns report percentage overhead over vanilla inference when generating 10 candidates.

Our work introduces the first approach to learn semantic summary representations of generated responses through learnable token embeddings. Our LSC method operates in a **post-generation** manner, appending learnable tokens after the assistant's response generation is complete to capture the semantic essence of each response. We refer to these as "*summary tokens*" since they are trained to encode compact semantic summaries of the full responses. Crucially, since we only update the summary token embeddings while keeping the base LLM frozen, our method preserves the model's original natural language capabilities without any degradation. This design enables efficient semantic consistency evaluation through lightweight forward passes while maintaining complete separation between consistency measurement and the model's core competencies.

Method

Inference Phase

Figure 1 illustrates the Latent Self-Consistency (LSC) inference pipeline in three phases: Candidate generation, Post-generation encoding, and Majority answer detection.

For (1) Candidate generation, given an input question we prompt the base LLM and sample N candidate responses. For (2) Post-generation encoding, we append K learnable summary tokens at the end of each response and run a lightweight forward pass to produce response representations $\{z_i\}$. Unlike USC, which concatenates all responses into a new prompt and recomputes the entire sequence, LSC appends only a few tokens at the end of each candidate and reuses the existing key–value (KV) cache from the candidate generation phase. As a result, the model needs to compute representations only for the K additional tokens rather than processing entire sequences from scratch. This makes LSC highly efficient as it only adds negligible overhead on top of base LM: at most 0.9% runtime and 0.005% memory overhead in our experiments.

For (3) Majority answer detection, we first compute the cosine similarity matrix between all pairs of response embeddings, $S_{ij} = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}$. We then assign an exponentially weighted similarity score to each response i,

$$w_i = \frac{1}{N-1} \sum_{j \neq i} \exp\big(S_{ij}/\tau'\big),$$

with $\tau' = 0.5$. A simple arithmetic mean of $\{S_{ij}\}_{j \neq i}$ treats all the responses equally and will likely pull w_i toward the global centroid, especially in the presence of outliers. In contrast, the exponential weighting effectively suppresses low-similarity responses, resulting in w_i being primarily determined by candidates from the majority set.

To further suppress the influence of residual outliers, we additionally propose a Dynamic Top-K scheme that estimates the size of the majority set directly from the similarity structure. For each $K \geq 2$, let σ_i sort other responses in decreasing similarity to i, and define the average similarity to the K nearest neighbors

$$w_i^{(K)} = \frac{1}{K} \sum_{j=1}^{K} S_{i,\sigma_i(j)},$$

with the best score at that neighborhood size

$$w_{\max}^{(K)} = \max_i w_i^{(K)}.$$

We then track the drop between K and $K-1$,

$$\Delta^{(K)} = w_{\max}^{(K-1)} - w_{\max}^{(K)},$$

and choose

$$K^* = \arg\max_K \Delta^{(K)} - 1,$$

which corresponds to the neighborhood size just before the largest decrease in maximum similarity. Finally, we select the index

$$i^* = \arg\max_i w_i^{(K^*)}$$

as the majority choice.

Models	Decoding Methods	In-domain				Out-of-domain		Average
		GSM8K	MATH	TriviaQA	MMLU	TruthfulQA MC1	CommonsenseQA	
LLaMA3.1 8B-Inst.	Vanilla (1 path)	83.2	39.4	62.4	68.8	57.2	74.9	64.3
	Self Consistency	92.2	52.5	72.7	73.6	**63.0**	**79.4**	72.2
	WUCS	89.8	45.6	73.2	72.8	61.8	76.6	70.0
	USC	86.2	44.8	70.1	71.1	59.6	74.6	67.7
	LSC (Ours)	**92.3**	52.6	74.1	73.8	62.1	79.0	72.3
	+Dynamic TopK	92.2	**52.9**	**75.7**	**73.9**	62.7	79.1	**72.8**
Qwen3-8B	Vanilla (1 path)	92.8	71.9	48.1	76.7	72.6	81.6	74.0
	Self Consistency	**94.5**	**76.7**	**56.1**	**77.2**	75.2	82.6	**77.1**
	WUCS	93.8	72.1	49.9	76.8	75.0	81.9	74.9
	USC	93.7	75.6	50.9	76.0	74.4	81.8	75.4
	LSC (Ours)	**94.5**	75.9	55.5	77.1	74.9	**82.7**	76.8
	+Dynamic TopK	94.4	76.6	53.2	77.0	**75.4**	82.6	76.5
LLaMA3.3 70B-Inst.	Vanilla (1 path)	95.9	65.1	85.2	84.1	73.9	84.4	81.4
	Self Consistency	96.9	68.6	86.3	**84.6**	75.0	84.7	**82.7**
	WUCS	95.5	63.7	84.6	84.3	74.2	84.1	81.1
	USC	95.8	68.2	**86.5**	**84.6**	74.8	**84.9**	82.5
	LSC (Ours)	96.9	**68.7**	86.3	84.2	**75.3**	84.3	82.6
	+Dynamic TopK	**97.0**	**68.7**	86.4	84.2	**75.3**	84.5	**82.7**

Table 2: Performance comparisons on short-answer benchmarks. Only **In-domain** sets were used for summary-token training. The highest score in each column is in **bold**, and the second-highest is underlined. All scores are reported as percentages.

Training Phase

Dataset Construction We assemble a balanced mix of short-answer and long-answer samples using LLM-based pseudo-labeling. For each question, an LLM generates an answer from which we extract the answer string as the label, identifying which responses converge on the same conclusion. Short-answer data are drawn from GSM8K (Cobbe et al. 2021), MATH (Hendrycks et al. 2021b), and TriviaQA (Joshi et al. 2017). Long-answer samples are derived from MMLU (Hendrycks et al. 2021a) by prompting chain-of-thought rationales and removing the short-form answer to yield 'free-form explanations with known pseudo-labels'. We maintain a 50:50 ratio of short to long formats. Detailed dataset construction procedures, statistics, and examples are provided in Appendix A, with prompt templates in Appendix B.

Learning Representations via Supervised Contrastive Learning To bring semantically consistent responses closer together, we use supervised contrastive learning: embeddings of responses sharing the same answer label are pulled together, while those with different labels are pushed apart. Concretely, we append K learnable special tokens after each response and fine-tune only their embeddings (e.g. $K = 6$, $d = 4096$ for 8B LLM, updating $< 3 \times 10^{-6}$ of parameters). Let $\{h_{i,1}, \ldots, h_{i,K}\}$ be the final-layer representations of those tokens for response i; we compute

$$\bar{h}_i = \frac{1}{K} \sum_{m=1}^{K} h_{i,m}, \quad z_i = \frac{\bar{h}_i}{\|\bar{h}_i\|}.$$

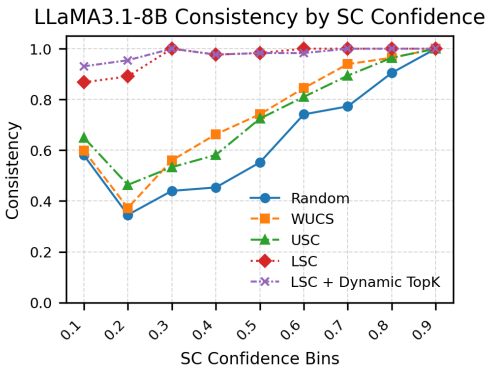

Figure 2: Consistency (fraction of examples where the selected response is in the true majority set) vs. majority-set size on MATH with LLaMA3.1-8B-Instruct, showing LSC outperforming all baselines and Dynamic Top-K giving additional gains for small majority sets.

With y_i the answer label, the supervised contrastive loss (He et al. 2020; Khosla et al. 2020) over a batch of size N is

$$\mathcal{L}_{\text{SCL}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{I}_i^+|} \sum_{j \in \mathcal{I}_i^+} \log \frac{\exp(z_i \cdot z_j / \tau)}{\sum_{k \neq i} \exp(z_i \cdot z_k / \tau)},$$

where $\mathcal{I}_i^+ = \{j \neq i : y_j = y_i\}$ and $\tau = 0.07$.

Experiments

Experimental Setup

We evaluate LSC across diverse reasoning tasks covering both short-answer and long-answer formats.

Models	Decoding Methods	MSMARCO v2.1 Natural Language Generation						TruthfulQA
		Semantic Similarity	BLEU 1	BLEU 2	BLEU 3	BLEU 4	ROUGE L	Truth(%)
LLaMA3.1-8B-Instruct	Vanilla (1 path)	0.8805	0.4122	0.3137	0.2488	0.2005	0.5075	39.9
	WUCS	0.8815	0.3904	0.2996	0.2392	0.1938	0.4987	42.8
	USC	0.8812	0.4065	0.3103	0.2463	0.1989	0.5086	44.2
	LSC(Ours)	0.8838	0.4269	0.3274	0.2613	0.2118	0.5207	43.5
	+Dynamic TopK	**0.8861**	**0.4336**	**0.3333**	**0.2667**	**0.2167**	**0.5270**	**44.4**
Qwen3-8B	Vanilla (1 path)	**0.8842**	0.3750	0.2876	0.2303	0.1875	0.4947	43.6
	WUCS	0.8835	0.3620	0.2788	0.2242	0.1835	0.4886	44.7
	USC	0.8824	0.3608	0.2772	0.2223	0.1812	0.4871	43.1
	LSC(Ours)	0.8838	0.3756	0.2896	**0.2330**	**0.1906**	**0.4964**	**44.9**
	+Dynamic TopK	0.8836	**0.3761**	**0.2897**	**0.2330**	0.1905	0.4958	44.6
LLaMA3.3-70B-Instruct	Vanilla (1 path)	**0.8926**	0.4694	0.3662	0.2981	0.2463	0.5618	48.0
	WUCS	0.8914	0.4277	0.3343	0.2716	0.2242	0.5500	48.7
	USC	0.8913	0.4623	0.3603	0.2930	0.2423	0.5596	**50.6**
	LSC(Ours)	0.8918	0.4724	0.3698	0.3016	0.2499	0.5629	48.8
	+Dynamic TopK	0.8922	**0.4768**	**0.3733**	**0.3042**	**0.2519**	**0.5652**	49.3

(a) Results on free-form generation QA tasks MSMARCO-NLG and TruthfulQA.

Models	Decoding Methods	HumanEval	HumanEval+	MBPP	MBPP+	CNN Dailymail (Summarization Task)			
		Pass@1	Pass@1	Pass@1	Pass@1	ROUGE 1	ROUGE 2	ROUGE L	BertScore
LLaMA3.1-8B-Instruct	Vanilla (1 path)	54.47	47.13	64.83	54.50	0.3091	0.1087	0.2077	0.8699
	WUCS	55.50	51.00	67.03	55.10	0.3033	0.1113	0.2033	0.8689
	USC	56.93	50.80	68.43	58.57	0.3033	0.1048	0.2013	0.8689
	LSC(Ours)	57.93	51.23	69.23	58.90	**0.3173**	**0.1127**	0.2130	0.8712
	+Dynamic TopK	**58.50**	**51.80**	**69.30**	**60.60**	0.3168	0.1124	**0.2149**	**0.8713**
Qwen3-8B	Vanilla (1 path)	77.80	73.40	81.03	69.23	0.2978	0.0977	0.1968	0.8683
	WUCS	**80.30**	**75.20**	82.43	70.17	0.2904	0.0966	0.1897	0.8668
	USC	78.03	73.17	**82.97**	**71.00**	0.2864	0.0950	0.1888	0.8660
	LSC(Ours)	79.30	74.20	81.20	69.40	0.3033	**0.1007**	**0.1997**	**0.8691**
	+Dynamic TopK	78.57	73.50	82.00	70.10	**0.3034**	0.1000	0.1992	0.8690
LLaMA3.3-70B-Instruct	Vanilla (1 path)	82.30	**76.80**	84.90	68.31	0.2536	0.0930	0.1700	0.8609
	WUCS	81.67	73.21	87.33	70.60	0.2418	0.0902	0.1626	0.8595
	USC	79.93	74.44	86.20	71.42	0.2538	0.0941	0.1699	0.8610
	LSC(Ours)	82.30	75.00	**87.80**	**73.81**	0.2540	0.0939	0.1703	0.8611
	+Dynamic TopK	**83.51**	75.57	86.82	72.50	**0.2558**	**0.0947**	**0.1714**	**0.8613**

(b) Performance on code generation (HumanEval+, MBPP+) and summarization (CNN/DailyMail) tasks.

Table 3: Long-answer task performance. The best result in each column is **bold** and the second-best is underlined. For TruthfulQA, the Truth score was obtained using GPT-4.1. (See appendix B)

Short-Answer Tasks: We use 6 benchmarks including mathematical reasoning (GSM8K (Cobbe et al. 2021), MATH (Hendrycks et al. 2021b)), factual knowledge (TriviaQA (Joshi et al. 2017), MMLU (Hendrycks et al. 2021a)), and commonsense reasoning (CommonsenseQA (Talmor et al. 2018), TruthfulQA-MC1 (Lin, Hilton, and Evans 2022)). The first four datasets are considered *in-domain* (used during representations learning), while the latter two are *out-of-domain*.

Long-Answer Tasks: We evaluate on 5 benchmarks requiring extended responses: open-ended QA (TruthfulQA (Lin, Hilton, and Evans 2022), MSMARCO-NLG (Bajaj et al. 2016)), code generation (HumanEval (Chen et al. 2021), MBPP (Austin et al. 2021) and their enhanced versions HumanEval+, MBPP+ (Liu et al. 2023)), and summarization (CNN/DailyMail (Hermann et al. 2015)). All long-answer benchmarks are strictly *out-of-domain* and were not used during summary-token training.

Setup: For each dataset, we sample up to 1,000 examples and generate 10 candidate responses per question using nucleus sampling (top-$p = 0.95$, temperature 0.9). We evaluate three instruction-tuned models: LLaMA3.1-8B-Instruct, LLaMA3.3-70B-Instruct (Dubey et al. 2024), and Qwen3-8B (Bai et al. 2023). All models use six learnable summary tokens (<|Summary1|>, ..., <|Summary6|>) trained on a balanced mixture of short and long-answer samples.

Results

Overall Performance and Efficiency Table 1 compares all consistency-based selection methods. LSC achieves the

Models	Decoding Methods	Short Answer				Long Answer			Average
		GSM8K	MATH	TriviaQA	MMLU	TruthfulQA	MSMARCO-NLG	HumanEval	
LLaMA3.1-8B-Instruct	Random	83.70	64.20	74.90	84.00	73.70	77.50	66.46	74.92
	WUCS	89.30	71.10	79.90	88.90	77.10	86.20	69.51	80.29
	USC	86.20	72.20	76.80	86.50	81.00	78.90	71.95	79.08
	LSC(Ours)	99.60	94.30	**91.90**	98.80	**93.00**	**93.20**	**80.49**	**93.04**
	+Dynamic TopK	**100.00**	**96.90**	91.40	**99.00**	91.90	91.80	78.66	92.81
Qwen3-8B	Random	96.60	84.50	67.10	95.40	73.70	78.50	79.88	82.24
	WUCS	97.30	86.30	77.60	96.70	74.90	85.80	86.59	86.46
	USC	97.10	89.40	75.50	94.00	76.50	76.70	81.71	84.42
	LSC(Ours)	**99.80**	97.40	**93.50**	**100.00**	**94.00**	93.60	**94.51**	**96.12**
	+Dynamic TopK	**99.80**	**98.40**	**93.50**	99.90	91.80	**94.50**	91.46	95.62
LLaMA3.3-70B-Instruct	Random	98.30	83.70	91.70	95.90	77.40	79.70	76.83	86.22
	WUCS	97.90	81.80	91.00	97.60	80.80	80.10	76.83	86.58
	USC	98.70	91.10	93.00	98.10	86.50	87.80	75.61	90.12
	LSC(Ours)	**100.00**	99.20	**99.20**	**100.00**	**94.10**	95.40	**93.29**	**97.31**
	+Dynamic TopK	**100.00**	**99.50**	99.00	**100.00**	92.30	**96.30**	89.02	96.59

Table 4: Consistency scores for each decoding method and model are defined as the fraction of samples in which the selected answer matches the most frequent answer among 10 generated reasoning paths. Results are reported for short-form benchmarks and long-form tasks. For short-answer datasets, consistency is measured by the direct frequency of answers. For the long-answer tasks, GPT-4.1 was used to identify which response occurred most often among the 10 paths. All scores are reported as percentages.

best overall performance, matching Self-Consistency (SC) on short-answer tasks while delivering the highest scores on long-answer benchmarks.

Each existing method faces critical limitations: SC remains limited to short-answer tasks due to exact string matching, making it inapplicable to free-form generation. WUCS suffers performance degradation on short-answer task. USC incurs substantial computational overhead, increasing inference time by approximately 10% and requiring over 15% additional memory, making it impractical for real-time deployment.

In contrast, LSC achieves comparable or superior accuracy while adding at most 0.9% to inference time and virtually no additional memory overhead. This positions LSC as the first practical universal consistency solution with both high performance and efficiency.

Short-Answer Results Table 2 shows that LSC and Self-Consistency (SC) outperform all other methods on short-answer benchmarks. LSC consistently surpasses SC on mathematical reasoning tasks by overcoming SC's exact string matching limitation through representation-based semantic comparison.

LSC performs well on both in-domain and out-of-domain benchmarks, demonstrating strong generalization. While baseline methods (WUCS and USC) improve over vanilla generation, they remain substantially behind SC and LSC. The optional dynamic Top-K boundary detection further improves performance, particularly on MATH where majority cluster sizes are small, by effectively filtering low-similarity outliers (Figure 2).

Long-Answer Results Tables 3a and 3b report long-answer results. LSC generally outperforms baselines across models and tasks. On TruthfulQA, LSC attains the highest Truth scores, surpassing USC. On MSMARCO-NLG, LSC yields clear gains on all metrics, whereas WUCS and USC sometimes even fall below the vanilla baseline.

Although the summary tokens are trained only on QA data, LSC matches or exceeds other methods on code generation benchmarks, indicating effective transfer. On CNN/-DailyMail summarization, LSC obtains the best scores on all metrics. Overall, these results show that LSC handles diverse long-form generation tasks while adding only minimal computational overhead.

Consistency Analysis To verify that our selection reflects the most frequent "majority" response, we define a *consistency* metric: the fraction of times the chosen response belongs to the true majority set. For short-answer tasks, the majority set is given by exact-match voting; for long answers, we use GPT-4.1 to identify it (see Appendix B).

Table 4 shows large gaps across methods. LSC attains near-perfect consistency, averaging 97.8% on short-answer benchmarks versus WUCS (88.0%) and USC (88.2%), and 92.4% on long-answer tasks versus WUCS (79.8%) and USC (79.6%). This indicates that LSC's representations more reliably capture semantic agreement among responses.

Consistency matters because all such methods are intended to gain accuracy by implementing majority voting: higher consistency means closer adherence to this principle. Notably, there are settings where USC or WUCS achieve slightly higher task accuracy than LSC despite substantially lower consistency, suggesting that their gains are driven by inductive biases in their scoring rules rather than by more accurate detection of the true majority set. By contrast, when a method has higher consistency but lower accuracy, the bottleneck lies in the base model's generations, not in the selec-

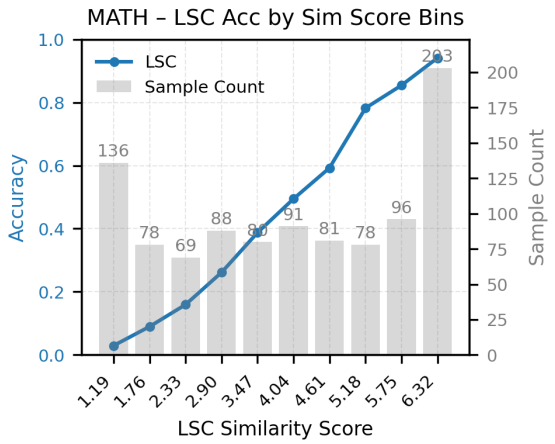

Figure 3: Calibration curve on the MATH dataset with LSC

Methods	Expected Calibration Error (ECE)			
	MATH	TriviaQA	TruthfulQA	HumanEval
SC	**0.076**	**0.044**	N/A	N/A
WUCS	0.315	0.675	0.349	0.316
LSC (Ours)	0.140	0.052	**0.217**	**0.170**

Table 5: Expected calibration error (ECE) of each method

tor. Overall, LSC most faithfully realizes the intended benefits of majority voting: its improvements come from reliably selecting the consensus response, supporting the assumption that majority agreement correlates with correctness.

Ablation Studies

Calibration Analysis

A key advantage of Self-Consistency is its built-in calibration, where SC confidence correlates with accuracy. To assess whether LSC preserves and extends this property, we evaluate two complementary notions of calibration.

First, Figure 3 shows LSC's calibration curves for MATH dataset. LSC maintains strong calibration while WUCS's calibration collapses (see Appendix D), demonstrating that higher LSC scores reliably indicate higher accuracy.

Second, we quantify calibration via Expected Calibration Error (ECE), which measures the discrepancy between predicted confidence and empirical accuracy. Table 5 shows that LSC achieves ECE values close to SC on short-answer tasks (MATH, TriviaQA), while substantially outperforming WUCS on long-answer tasks. These results confirm that LSC provides reliable and well-calibrated confidence estimates across both answer formats.

EoS Token vs. Learned Summary Tokens

To validate the necessity of learning specialized summary tokens, we compared LSC against EoS token-based variants. While representations from EoS token achieve reasonable accuracy on short-answer tasks, their consistency scores drop significantly on long-answer formats (detailed results in Appendix E). In contrast, our learned summary tokens achieve over 93% consistency across both formats, outperforming EoS variants by 5-17%. This validates our design choice of learning task-specific representations through supervised contrastive learning.

Method	MATH	TriviaQA	TruthfulQA	HumanEval
SBERT	43.9	71.8	42.7	56.7
Fine-tuned SBERT	50.4	**74.4**	43.1	56.7
LSC	**52.6**	74.1	**43.5**	**57.9**

Table 6: Comparison between LSC's learned summary-token embeddings and an off-the-shelf SBERT encoder

Off-the-Shelf Sentence Embeddings vs. Learned Summary Tokens

We also ask whether a generic sentence embedding model can be used in place of LSC's learned summary-token representations. As a baseline, we feed each candidate response to SBERT[1] and use its sentence embeddings in place of LSC's summary-token representations, while keeping the remainder of the selection procedure unchanged. Table 6 reports results for LLaMA3.1-8B-Instruct on four benchmarks.

Table 6 shows that off-the-shelf SBERT is clearly weaker than LSC on short-answer benchmarks, although it is reasonably competitive on long-answer tasks. When we train SBERT using our supervised contrastive learning framework, its short-answer performance rises to a level comparable to LSC while retaining good long-answer performance. This supports the effectiveness of our training framework itself; additional details and results are provided in Appendix L.

Conclusion

We introduced Latent Self-Consistency (LSC), a framework that unifies consistency-based selection across both short-answer and long-answer reasoning tasks. By learning compact semantic representations via learnable summary tokens and supervised contrastive learning, LSC matches the accuracy of Self-Consistency on short-answer benchmarks while generalizing effectively to long-answer scenarios.

Across 11 diverse reasoning benchmarks, LSC consistently outperforms existing selection methods while adding at most 0.9% inference-time overhead compared to vanilla decoding (versus roughly 10% for USC) and incurring negligible extra memory usage. The framework attains near-perfect majority-set identification, provides well-calibrated confidence estimates across answer formats, and preserves the original capabilities of the base model. As a result, LSC offers a practical, efficient, and broadly applicable consistency mechanism for large language models, suggesting a promising direction for universal consistency-based decoding methods.

[1] `all-mpnet-base-v2`

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant (RS-2023-00280883, RS-2023-00222663); by the National Research Foundation, Korea, under project BK21 FOUR(Dept. of Data Science, SNU, No. 5199990914569); by the National Super computing Center with super computing resources including technical support (KSC-2023-CRE-0176, KSC-2024-CRE-0065); by the Korea Institute of Science and Technology Information (KISTI) in 2025 (No.(KISTI) K25L1M1C1), aimed at developing KONI (KISTI Open Neural Intelligence), a large language model specialized in science and technology; and by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-02263754, Human-Centric Embodied AI Agents with Autonomous Decision-Making); by grant (25202MFDS003) from Ministry of Food and Drug Safety in 2025; by AI-BIO Research Grant through Seoul National University; Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain).

References

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021. Program Synthesis with Large Language Models. arXiv:2108.07732.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen Technical Report. arXiv:2309.16609.

Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 Co-Located with the 30th Annual Conference on Neural Information Processing Systems (NeurIPS 2016)*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models Are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chen, X.; Aksitov, R.; Alon, U.; Ren, J.; Xiao, K.; Yin, P.; Prakash, S.; Sutton, C.; Wang, X.; and Zhou, D. 2024. Universal Self-Consistency for Large Language Model Generation. In *The Twelfth International Conference on Learning Representations*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2023. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The LLaMA 3 Herd of Models. arXiv:2407.21783.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021a. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving with the Math Dataset. arXiv:2103.03874.

Hermann, K. M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems*, 1693–1701.

Jain, S.; Ma, X.; Deoras, A.; and Xiang, B. 2024. Lightweight Reranking for Language Model Generations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6960–6984. Bangkok, Thailand: Association for Computational Linguistics.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, 18661–18673.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 4582–4597.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252.

Liu, J.; Xia, C. S.; Wang, Y.; and Zhang, L. 2023. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Advances in Neural Information Processing Systems*, volume 36, 21558–21572.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2018. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. arXiv:1811.00937.

Wang, X.; Li, Y.; Feng, S.; Yuan, P.; Pan, B.; Wang, H.; Hu, Y.; and Li, K. 2024. Integrate the Essence and Eliminate the Dross: Fine-Grained Self-Consistency for Free-Form Language Generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11782–11794. Bangkok, Thailand: Association for Computational Linguistics.

Wang, X.; Wei, J.; Schuurmans, D.; Ma, Q.; Chi, E.; Sharan, S.; Narang, A.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.