Test-Time Augmentation for LLMs: When Input Diversity Beats Output Diversity at Matched Compute

Nikita Kozodoi
Amazon Web Services
kozodoi@amazon.com

Zainab Afolabi
Amazon Web Services
zafolabi@amazon.com

Jack Butler
Amazon Web Services
jackbtlr@amazon.com

Abstract

Test-time scaling improves LLM accuracy but multiplies inference cost, making the accuracy gained per unit of compute the metric that matters in deployment. Self-consistency is one of the established approaches, which spends this budget entirely on the output side by sampling repeated reasoning paths. We study Test-Time Augmentation (TTA), which extends self-consistency by also perturbing the input, aggregating predictions across transformed versions of the input, and ask whether input-side diversity converts compute into accuracy more efficiently than output-side diversity. We perform a systematic, matched-compute comparison: we evaluate three simple input-side strategies (semantic rephrasing, lexical perturbations, and visual transformations) across six datasets covering general and multilingual knowledge, mathematical reasoning, multi-modal question answering, and sentiment classification, against chain-of-thought prompting and self-consistency. Semantic rephrasing delivers consistent and statistically significant accuracy gains while Pareto-dominating self-consistency on cost-effectiveness, delivering roughly 1.8× more accuracy per dollar and outperforming it on five of six tasks. We further analyze the number of augmentations, multi-modal strategies, and base model scaling, finding that TTA is most cost-effective for mid-tier models where a stronger model is unavailable or too expensive. Our findings indicate that for current mid-tier LLMs, varying the input converts inference compute into accuracy more efficiently than varying the reasoning path alone. The TTA implementation is available at https://github.com/aws-samples/sample-genai-reflection-for-bedrock.

1 Introduction

Large Language Models (LLMs) demonstrate strong performance across diverse tasks, from mathematical reasoning to multi-modal understanding. Many of the strongest accuracy gains now come from spending more compute at inference time, but this compute is not free: every additional sample or reasoning step adds latency and cost, so the quantity that matters in deployment is the accuracy gained per unit of compute. Practitioners therefore seek inference-time techniques that improve accuracy without retraining and that convert a fixed compute budget into accuracy as efficiently as possible. Test-Time Augmentation (TTA) is a well-established technique in supervised learning that aggregates predictions across multiple transformed versions of a test input (Shanmugam et al., 2020). In computer vision, TTA combines predictions over rotated, flipped, or cropped images, and prior work has shown smaller gains in NLP applications such as text classification (Lu et al., 2022) and factual probing (Kamoda et al., 2023). Despite its success in supervised settings, TTA has not been systematically studied for generative LLMs.

Applied to LLMs, TTA generates multiple variants of an input (e.g., paraphrases of a question) and aggregates predictions over them via majority voting, leveraging input-side diversity. Figure 1 illustrates the TTA pipeline for multi-modal question answering. The intuition is that LLMs are sensitive to surface form: minor changes in phrasing can shift predictions, so aggregating across phrasings reduces the variance contributed by any

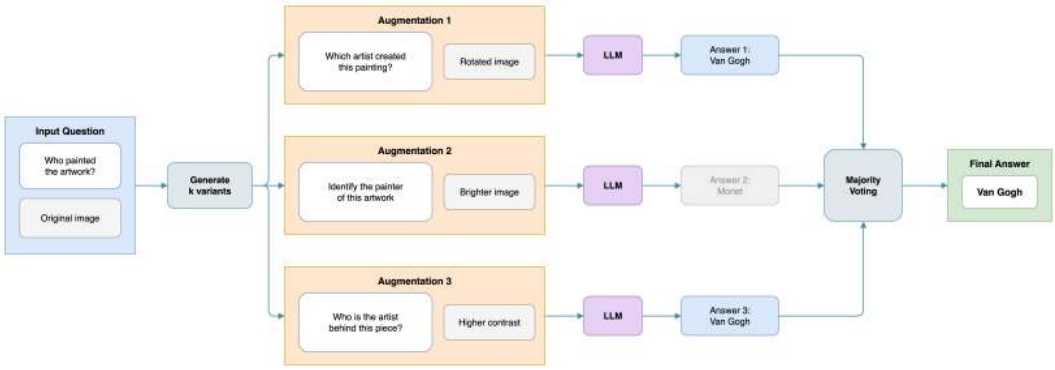


Figure 1: TTA framework. The input question and image are augmented into k variants through text paraphrasing and image transformations. Each variant is processed independently by the LLM, producing answer candidates that are aggregated via majority voting.

single one. TTA is training-free and independent of the choice of base model, making it straightforward to layer onto existing inference pipelines.

Most inference-time scaling techniques sample, refine, or restructure the model's output: examples include self-consistency (Wang et al., 2023), which samples multiple reasoning paths from chain-of-thought (CoT) prompting (Wei et al., 2022), Self-Refine (Madaan et al., 2023), and Tree of Thoughts (Yao et al., 2023). These methods spend the entire compute budget on the output side, repeatedly re-running the model on the same input. TTA can be viewed as extending self-consistency: rather than sampling repeatedly from a fixed input, it also perturbs the input before sampling, adding input-side diversity on top of output-side diversity. This input-side regime is comparatively under-explored, yet it draws on the same budget and is therefore a direct competitor for where to invest each additional inference call. The individual augmentation techniques we study, paraphrasing, character-level noise, and image transforms, are all deliberately simple and established, which lets us isolate the efficiency question itself: our focus is a systematic, matched-compute comparison of input-side against output-side diversity. Prior work has applied paraphrase aggregation to narrow tasks such as mathematical reasoning (Zhou et al., 2024) and intent classification (Yadav et al., 2024), but to the best of our knowledge, no study has systematically compared input augmentation strategies for LLMs across diverse tasks, nor benchmarked them against CoT and self-consistency at matched compute.

This paper investigates the following question: *at fixed compute, does input-side or output-side diversity convert that compute into accuracy more efficiently?* We answer it on six diverse benchmarks: MMLU, MMMLU, MMMU, HLE, Math500, and IMDB Reviews.

Our contributions are three-fold. First, we compare input-side against output-side diversity at matched compute, benchmarking TTA against CoT prompting and self-consistency on both accuracy and cost per additional LLM call. Second, we provide evidence that semantic rephrasing outperforms the strongest baseline on five of six benchmarks and Pareto-dominates it on cost-effectiveness, with statistically significant gains over the baselines and roughly 1.8× more accuracy per dollar than self-consistency. Third, we conduct ablation studies on the number of augmentations, cost-accuracy trade-offs, multimodal strategies, and base model scaling, distilling practical deployment guidance on when the extra inference compute is worth spending and identifying the mid-tier model regime as where TTA is most cost-effective. The TTA implementation is available at `https://github.com/aws-samples/sample-genai-reflection-for-bedrock`.

2 Related Work

A growing body of work scales test-time compute to improve LLM accuracy without retraining. Snell et al. (2024) show that repeated sampling and verification can outperform

scaling model parameters, while Butler et al. (2025) examine cost-quality-speed trade-offs in iterative reflection. Self-Refine (Madaan et al., 2023) and self-debugging (Chen et al., 2023) use iterative self-critique, although the reliability of self-verification is contested (Stechly et al., 2024; Valmeekam et al., 2023). Structured reasoning approaches such as Tree of Thoughts (Yao et al., 2023), Graph of Thoughts (Besta et al., 2024), and process reward models (Lightman et al., 2023; Setlur et al., 2024) modify the reasoning process itself, and test-time training (Akyürek et al., 2024; Hübotter et al., 2024) adapts model parameters during inference. Unlike these approaches, TTA operates only on input representations and requires neither parameter updates nor self-verification.

Among output-side methods, self-consistency (Wang et al., 2023) is the most widely used baseline, with extensions including Mirror-Consistency (Huang et al., 2025) and confidence-weighted voting (Taubenfeld et al., 2025). The input-side regime we study is motivated by the well-documented sensitivity of LLMs to prompt phrasing (Seleznyov et al., 2025; Chatziveroglou et al., 2025; Agrawal et al., 2025; Wahle et al., 2024), where minor surface changes can shift predictions substantially: aggregating across phrasings reduces the variance contributed by any single one.

Several methods exploit input rephrasings, but each targets a single task type and none frame the technique as an efficiency question benchmarked against self-consistency at matched compute, which is the gap we address. Deng et al. (2023) propose Rephrase-and-Respond (RaR), which uses a single rephrasing for clarification rather than aggregation; in contrast, semantic TTA generates k rephrasings and aggregates predictions through majority voting, leveraging diversity for variance reduction. Zhou et al. (2024) propose SCoP, which paraphrases mathematical problems to diversify reasoning, while Yadav et al. (2024) introduce PAG-LLM for intent classification by generating paraphrases and aggregating by confidence. CAPE (Jiang et al., 2023) ensembles augmented prompts for calibration rather than accuracy, and Kamoda et al. (2023) apply TTA to factual probing with mixed results. In computer vision, the closest analog is ZERO (Farina et al., 2024), which augments visual inputs N times for vision-language models. These works collectively span math, classification, and multi-modal tasks, but each in isolation and against task-specific baselines; our contribution is to unify them under one framework and to ask, across diverse tasks and at a fixed compute budget, whether input-side or output-side diversity is the more cost-effective use of each additional inference call.

A separate line of research leverages multiple distinct prompts through prompt ensembling: boosted ensembles (Pitis et al., 2023), multi-prompt decoding (Guo et al., 2024), and PREFER (Zhang et al., 2023), all of which typically require prompt optimization or selection. Model ensembles (Ai et al., 2025; Niimi, 2024) combine outputs from multiple models, which is often impractical at deployment scale. In contrast, TTA generates rephrasings on-the-fly without task-specific prompt engineering. Augmentation has also been extensively studied at training time (Chai et al., 2025; Costa-Jussà et al., 2022; Luong et al., 2024; Yao et al., 2025; Choi, 2025); our work differs by applying augmentation purely at inference time, with no parameter updates or additional training data.

3 Test-Time Augmentation for LLMs

3.1 Framework Overview

Given an input query x and an LLM f, standard inference produces a single response $y = f(x)$. TTA extends this by generating k augmented versions of the input $\{x_1, \ldots, x_k\}$, obtaining a prediction for each, and aggregating the results.

For tasks with discrete answers, we aggregate responses by majority voting. Let $\{y_1, \ldots, y_k\}$ denote the predictions for the augmented inputs. The final prediction is:

$$\hat{y} = \arg\max_{c} \sum_{i=1}^{k} \mathbf{1}[y_i = c] \tag{1}$$

where $\mathbf{1}[\cdot]$ is the indicator function and c ranges over possible answer choices. Ties are broken at random.

Table 1: Examples of inputs for each method. CoT prompting is the single-call baseline; TTA varies the input (input-side diversity); self-consistency feeds the same input k times and varies only the reasoning path (output-side diversity). Typos in lexical TTA are intentional.

Method	Diversity	Transformation	Input example
CoT prompting	None	None	What is the capital of France?
Semantic TTA	Input	Paraphrasing	Which city serves as France's capital? Tell me the capital city of France
Lexical TTA	Input	Character noise	What is the captial of Frnace? Whatt is the capital of France?
Self-consistency	Output	None	What is the capital of France? What is the capital of France?

The key design choice is the augmentation function. We investigate three TTA strategies (semantic and lexical for text, visual for images) and include self-consistency as a canonical inference-time scaling baseline. Table 1 shows examples for the text-based strategies.

3.2 Semantic TTA

Semantic TTA generates paraphrased versions of the input that preserve meaning while varying surface form. Given a question x, we prompt an LLM to produce k semantically equivalent rephrasings $\{x_1^{\text{sem}}, \ldots, x_k^{\text{sem}}\}$ in a single LLM call. Each rephrasing is then answered independently and predictions are aggregated by majority voting. This leverages the well-documented sensitivity of LLMs to prompt phrasing (Seleznyov et al., 2025; Chatziveroglou et al., 2025): aggregating across phrasings reduces variance from any single surface form. The rephrasing prompt instructs the LLM to preserve meaning, intent, and answer format while varying vocabulary and sentence structure. The prompt templates are provided in Appendix A.

The diversity of the augmented inputs depends on the rephraser model and its sampling temperature. We treat both as design choices and study them in our experiments, including using a stronger model than the answering model to generate the rephrasings. Producing all k rephrasings in a single call keeps the augmentation overhead small relative to the k answer calls. To ensure the rephrasings remain answerable, we explicitly instruct the rephraser to preserve answer choices, formatting requirements, and references to images so that the question stays well-posed under aggregation.

3.3 Lexical TTA

Lexical augmentation applies character-level perturbations to the input, requiring no additional LLM call for augmentation and thus being computationally cheaper. We apply three transformation types: random character swaps within words, character insertions and deletions, and injection of typos and spelling mistakes. We use a perturbation probability of 5% per word with a maximum of 10 perturbations per question. Prior work on character-level noise (Agrawal et al., 2025) suggests that higher rates degrade comprehension, while rates below 5% produce near-identical inputs.

Lexical TTA is motivated by the observation that LLMs are largely robust to small character-level corruption such as typos and reordering, so perturbed inputs should yield correct answers most of the time while still inducing diversity in the model's predictions. The trade-off is that the perturbations preserve surface form rather than meaning: they can shift predictions on borderline questions where the perturbed token coincides with a content word. As a result, lexical TTA is essentially free to apply but is expected to provide weaker gains than semantic TTA, a hypothesis we verify in Section 5.1.

3.4 Visual TTA

For multi-modal tasks involving images, we extend TTA to visual transformations of the image inputs while keeping the textual question unchanged. We apply three transformation types: small-angle rotation, brightness adjustments, and contrast adjustments. The rotation range $\pm\alpha^\circ$ and the photometric ranges $\pm\beta\%$ act as hyperparameters that control augmentation strength.

Following common practice in visual TTA (Shanmugam et al., 2020), we keep transformations mild to preserve the semantic content of the image: small rotations and modest brightness or contrast shifts produce visibly distinct yet recognizable variants of the original. For each test input, we sample k independent transformation parameter triples and apply them to obtain k augmented images, each paired with the original question. The resulting k predictions are aggregated by majority voting. We also explore combining visual and text-based augmentations on multi-modal benchmarks.

3.5 Self-Consistency Baseline

Self-consistency (Wang et al., 2023) is one of the most established inference-time scaling methods. Starting from CoT prompting (Wei et al., 2022), it samples k reasoning paths from the same input at temperature $T > 0$ and aggregates the resulting answers by majority voting. We include self-consistency in our comparison as the canonical output-side counterpart to input-side TTA. Concretely, given input x, we sample k responses $\{y_1, \ldots, y_k\}$ via independent inference calls at $T = 0.75$ with CoT prompting, then majority-vote over the extracted answers. Semantic TTA can be seen as a strict extension of this procedure: because it also samples at $T = 0.75$, each answer carries the same output-side diversity as self-consistency, plus additional input-side diversity from the distinct rephrasings. The comparison against self-consistency at matched k therefore isolates the marginal contribution of input variation: any improvement of input-side TTA over self-consistency is attributable to varying the input rather than the reasoning path alone.

4 Experimental Setup

4.1 Datasets

We evaluate TTA across six LLM benchmarks covering diverse domains and task types. MMLU (Hendrycks et al., 2021) tests multitask language understanding across 57 subjects, and its multilingual extension MMMLU (OpenAI, 2024), translated into 14 languages, lets us assess TTA robustness beyond English. MMMU (Yue et al., 2024) requires joint image and text reasoning, while HLE (Phan et al., 2025) poses expert-crafted questions across math, sciences, and humanities. Math500 (Lightman et al., 2023) covers math problems spanning algebra, arithmetic, geometry, and calculus, and IMDB Reviews (Maas et al., 2011) is a binary sentiment classification task over movie reviews. We use a randomly sampled subset of 400 examples from each dataset for evaluation, balancing comprehensive coverage with computational efficiency. Statistical significance testing is conducted in Section 5.1.

4.2 Models and Method Implementations

We conduct experiments using Claude 4.5 Haiku (Anthropic, 2025) as the primary model. All methods produce answers via the same CoT-prompting template at $T = 0.75$ (see Appendix A.1). They differ only in how the k candidate answers are generated. The compared methods are:

- **CoT prompting (baseline)** (Wei et al., 2022): single LLM call with no aggregation.
- **Self-consistency** (Wang et al., 2023): same input fed k times, majority voting over k independent responses.
- **Semantic TTA:** k rephrasings generated in one LLM call, majority voting over k independent responses.

- **Lexical TTA:** k character-perturbed copies of the input (5% probability per word, capped at 10 perturbations), majority voting over k independent responses.
- **Visual TTA:** k image transformations (rotation $\pm 3^\circ$, contrast and brightness $\pm 5\%$), majority voting over k independent responses on multi-modal inputs.

We evaluate each method with $k \in \{2, 4, 6\}$, and conduct extended experiments with k between 1 and 10 for ablation studies. For each method and dataset, the reported k is selected on a held-out sample disjoint from the evaluation subset, so the comparison does not tune k on the same examples used to report accuracy. On multi-modal datasets, we also experiment with combining visual and semantic TTA. Claude 4.5 Haiku as a base model balances capability and cost; we also ablate model size in Section 5.5.

4.3 Evaluation

We employ task-appropriate accuracy metrics for each benchmark. For MMLU, MMMLU, MMMU and HLE, we measure the accuracy as a fraction of correctly answered questions. The LLM response to each multiple-choice question is parsed to extract the answer, which is compared with the ground truth. For IMDB, we calculate the binary classification accuracy of the LLM-predicted sentiment. Math500 uses additional verification procedures to assess semantic equivalence between the LLM response and the ground truth. We apply string matching on normalized LaTeX expressions, followed by symbolic equivalence checking with SymPy (Meurer et al., 2017) to identify equivalent answers expressed in different forms.

All methods incur computational cost proportional to k. Semantic TTA incurs an additional cost for generating rephrasings, while lexical augmentation is nearly free. Self-consistency requires k independent inference calls. We analyze cost-accuracy trade-offs in Section 5.2, providing practical guidance for deploying TTA under different compute budgets.

All experiments are conducted using Amazon Bedrock with a fixed random seed. Token costs are recorded as of June 2026 using on-demand pricing. The prompt templates are provided in Appendix A.

5 Results and Analysis

5.1 Main Results

Figure 2 reports the accuracy improvement of each TTA method and the self-consistency baseline over single-call CoT prompting for each dataset. Across all six benchmarks, semantic TTA delivers the largest average gain of 1.8 percentage points (pp), and outperforms self-consistency on five of six benchmarks. The gap over self-consistency is largest on

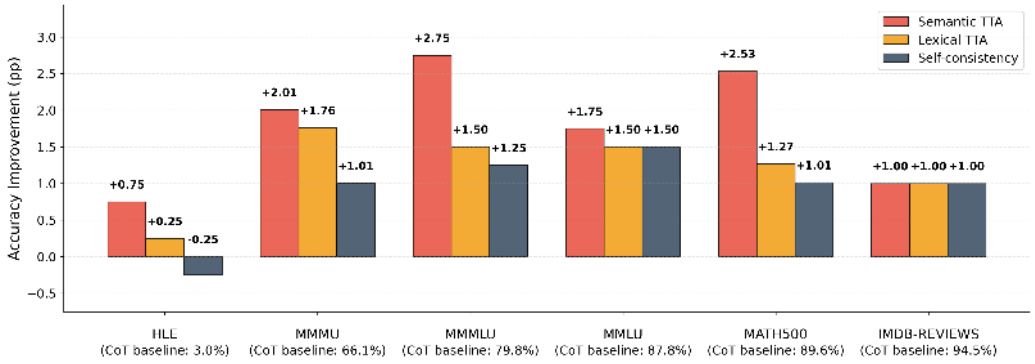


Figure 2: Accuracy improvement over the CoT prompting baseline across benchmarks (in pp). Semantic TTA achieves the highest gains across most benchmarks, outperforming the self-consistency baseline on five of six tasks. The number of augmentations for each method is selected via grid search over $k \in \{2, 4, 6\}$.

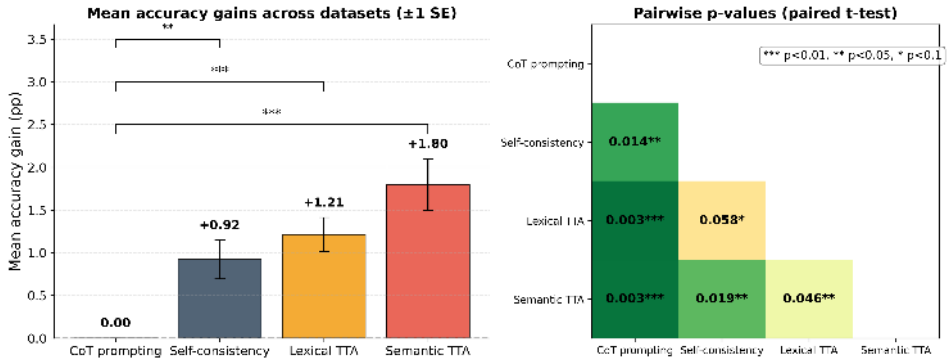


Figure 3: Pairwise statistical significance of accuracy gains across datasets. Semantic TTA shows statistically significant improvements over CoT and over self-consistency ($p < 0.05$).

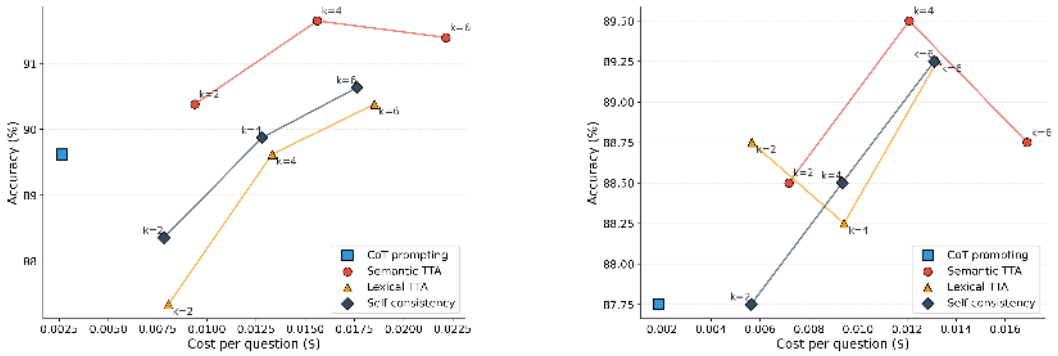
Math500 (+1.52 pp) and MMMLU (+1.50 pp), where aggregating across alternative phrasings overcomes prompt-formulation sensitivity that repeated sampling alone does not address. Lexical TTA achieves smaller gains (1.2 pp on average), suggesting that character-level perturbations introduce noise that partially offsets the benefits of aggregation.

Self-consistency improves over single-call CoT but is consistently outperformed by semantic TTA. This indicates that gains from aggregation have two sources: variance reduction from repeated sampling (captured by self-consistency), and additional diversity introduced by varying the input (captured by input-side TTA). Semantic TTA benefits from both.

Figure 3 reports paired t-tests on mean accuracy gains aggregated across all six datasets. Semantic TTA achieves statistically significant improvements over both single-call CoT ($p < 0.01$) and self-consistency ($p < 0.05$). Lexical TTA also significantly outperforms single-call CoT ($p < 0.01$), as does self-consistency ($p < 0.05$), but their gains are not significantly different from each other at the 5% level. A non-parametric paired bootstrap over the pooled per-question gains ($n = 2{,}400$) confirms these conclusions, with the 95% confidence interval for semantic TTA well above zero at $[+0.88, +2.71]$ pp (see Appendix C for details).

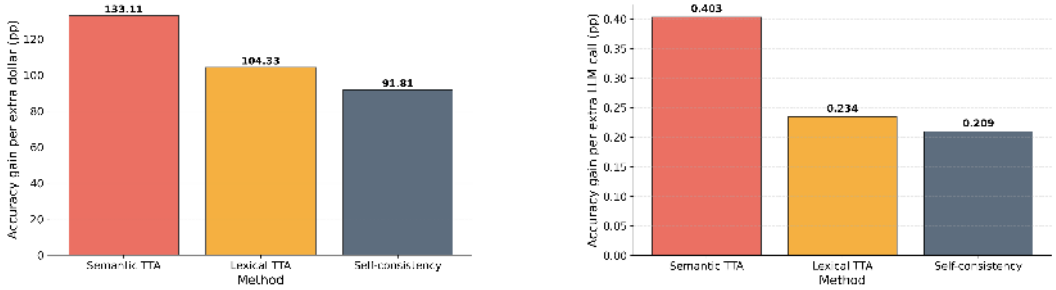
5.2 Cost-Accuracy Trade-offs

Figure 4 shows cost-accuracy frontiers on Math500 and MMLU. Semantic TTA incurs a slightly higher cost per question due to the rephrasing call, but achieves the highest accuracy at every cost level on both datasets except for dropped accuracy at $k = 6$ on MMLU. At $k = 4$, semantic TTA reaches its peak accuracy, while self-consistency requires more samples to approach a comparable level. Lexical TTA traces a frontier close to self-consistency at a



(a) Math500 (b) MMLU

Figure 4: Cost-accuracy frontiers (Math500, MMLU).



(a) Accuracy gain per extra dollar spent.

(b) Accuracy gain per extra LLM call.

Figure 5: Cost-effectiveness of each method, measured as accuracy gain relative to additional cost (left) and number of LLM calls (right). Semantic TTA is most effective on both.

lower cost, since it does not require an extra rephrasing call, making it a reasonable fallback when even small augmentation overhead is undesirable.

Figure 5 compares accuracy gain per unit cost and per LLM call across methods. Semantic TTA achieves the highest accuracy gain per dollar spent and per additional LLM call, delivering roughly 1.8× more accuracy per dollar than self-consistency despite its higher per-call cost, making it the most cost-effective option in our comparison.

While statistically significant, we note that gains of 1–2pp may not justify a 2–6× cost increase in all settings; TTA is most valuable when baseline accuracy is moderate (40–80%).

5.3 Number of Augmentations

Table 2 reports the optimal k for each dataset and method. Semantic TTA reaches its peak with fewer augmentations on average ($k = 4.33$), indicating higher-quality diversity compared to lexical TTA and self-consistency.

Figure 6 shows extended experiments on Math500 with k up to 10. Accuracy is non-monotonic in k: small drops at certain k reflect stochastic paraphrase generation and tie-breaking dynamics, particularly for odd vs. even k. These fluctuations are small (within 1 pp). Semantic TTA peaks at $k = 5$ with diminishing returns thereafter; self-consistency continues to improve up to $k = 10$, consistent with Wang et al. (2023)'s finding that self-consistency benefits from larger sample counts, and at $k = 10$ it nearly matches semantic

Dataset	Semantic TTA	Lexical TTA	Self-consistency
HLE	6	6	**2**
IMDB	4	**2**	6
Math500	**4**	6	6
MMLU	**4**	6	6
MMMLU	6	6	**2**
MMMU	**2**	6	6
Average	**4.33**	5.33	4.67

Table 2: Optimal number of augmentations per dataset and method, selected on a held-out sample via grid search over $k \in \{2, 4, 6\}$. Bold marks the smallest optimal k in each row. Semantic TTA peaks at smaller k on average, reaching its best accuracy with less compute.

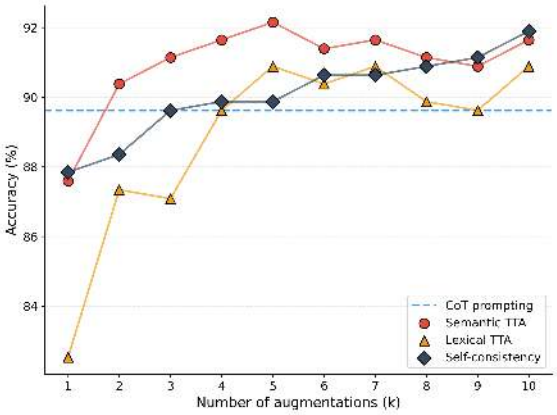


Figure 6: Effect of number of augmentations on Math500. Semantic TTA peaks at $k = 5$; self-consistency continues to improve up to $k = 10$.

Table 3: Multi-modal results on MMMU. Visual TTA improves over baseline but falls behind text-based TTA. The k is selected on a held-out sample via grid search over $k \in \{2, 4, 6\}$.

Modality	Method	Selected k	Accuracy (%)
–	CoT prompting	–	66.08
	Self-consistency	6	67.09
Text	Lexical TTA	6	67.84
	Semantic TTA	**2**	**68.09**
Image	Visual TTA	6	67.59
Text + Image	Alternating	2	66.33
	Combined	4	65.08

TTA on Math500. Based on these results, we recommend starting with $k = 4$ for semantic TTA, balancing accuracy and cost.

5.4 Augmentation Modality

For multi-modal tasks, we examine whether text or image augmentation is more effective. Each MMMU example includes a textual question and at least one image. This allows us to compare augmenting text, images, or both. Table 3 reports results on MMMU.

Text-based semantic TTA achieves the highest accuracy (68.09%) with $k = 2$, outperforming visual TTA (67.59%) at $k = 6$. This suggests that LLM predictions are more sensitive to question phrasing than to the mild visual perturbations we apply: the model's image understanding is already robust to the small rotations and brightness or contrast shifts we test. We emphasize that this conclusion is scoped to these mild photometric and geometric transforms; stronger or structurally different visual augmentations may behave differently and are left to future work. Visual TTA still outperforms self-consistency (67.09%), confirming that input-side diversity helps even when applied to the image modality alone.

Combining text and image TTA decreases performance, as applying both modalities at once introduces inconsistent variations that confuse majority voting. The combined variant (65.08%) falls below single-call CoT (66.08%), indicating that simultaneous perturbation of both modalities accumulates enough noise to outweigh the benefits of aggregation. For multi-modal tasks, we therefore recommend text-based semantic augmentation alone.

5.5 Base Model Size

We examine whether TTA benefits vary across model sizes by evaluating all methods on Claude 4.5 Haiku, Sonnet, and Opus on MMMLU. Table 4 reports the results.

Semantic TTA achieves the highest accuracy at every model size. The magnitude of improvement decreases with model scale: Haiku gains 2.75pp, Sonnet – 2pp, and Opus – only 0.25pp. This pattern suggests that TTA provides larger benefits when baseline accuracy leaves more room for improvement. For high-performing LLMs already near ceiling, TTA offers marginal gains as the model is already robust to prompt variations. The same pattern

Table 4: Performance across model sizes on MMMLU. Semantic TTA shows the largest gains on smaller models. Gains diminish as base accuracy approaches ceiling.

Method	Claude Haiku	Claude Sonnet	Claude Opus
CoT prompting	79.75	85.50	88.25
Self-consistency	81.00	86.75	88.25
Lexical TTA	81.25	86.75	87.00
Semantic TTA	**82.50**	**87.50**	**88.50**
Semantic TTA gain over CoT	+2.75	+2.00	+0.25

is even more pronounced for lexical TTA, which improves Haiku and Sonnet but degrades Opus (87.00% vs. 88.25% baseline), indicating that character-level noise can hurt strong models whose predictions are already close to ceiling.

Comparing across tiers, Haiku with semantic TTA (82.50%) does not match Sonnet's single-call accuracy (85.50%), and Sonnet with TTA (87.50%) remains below Opus's single-call accuracy (88.25%). TTA is therefore not a substitute for upgrading to a stronger model when one is available within budget. Rather than a limitation, this delineates where TTA belongs on the efficiency frontier: it is a compute-efficiency tool for the mid-tier regime, most valuable precisely when a larger model is unavailable or prohibitively expensive and the practitioner must extract more accuracy from a fixed, smaller base model.

6 Conclusion

This paper presents a systematic study of Test-Time Augmentation (TTA) for LLMs. We evaluate three TTA strategies, semantic rephrasing, lexical perturbations, and visual transformations, across six benchmarks spanning general and multilingual knowledge, mathematical reasoning, multi-modal understanding, and sentiment classification, and benchmark them at matched compute against CoT prompting (Wei et al., 2022) and self-consistency (Wang et al., 2023).

Several key findings emerge from our experiments. On accuracy, both semantic and lexical TTA achieve statistically significant gains over single-call CoT prompting, with semantic TTA additionally outperforming self-consistency on five of six benchmarks. On cost, semantic TTA Pareto-dominates self-consistency, achieving higher accuracy per dollar and per additional LLM call, while lexical TTA offers a lower-cost alternative that requires no rephrasing call.

We also draw practical guidance from our ablations. Semantic TTA reaches near-optimal accuracy at $k = 4$, while self-consistency continues to improve at larger k. For multi-modal tasks, text-based augmentation outperforms image-based augmentation, and combining modalities harms performance. Finally, TTA benefits are larger for smaller LLMs whose baseline accuracy leaves more room for improvement, and lexical TTA can even degrade strong models whose predictions are already close to ceiling.

Taken together, these results indicate that for current mid-tier LLMs, varying the input converts inference compute into accuracy more efficiently than varying the reasoning path alone. TTA is a simple, training-free way to push a fixed base model further along the cost-accuracy frontier, most useful when a stronger model is unavailable or too expensive. We hope these findings will encourage further exploration of input-side scaling as a route to more efficient reasoning, and inform practitioners deciding where to invest additional inference compute.

Our study has several limitations. First, TTA applies only to tasks with a well-defined notion of answer equivalence, where majority voting is meaningful; extending to open-ended generation such as summarization or translation would require different aggregation mechanisms, so TTA is not a universal drop-in for every LLM workload. Second, our evidence is limited to a single proprietary model family: while we observe consistent patterns across Claude Haiku, Sonnet, and Opus, behavior on open-weight models or substantially different families remains to be verified, and claims about LLMs in general should be read as claims about current mid-tier models. Third, our multilingual evidence aggregates over the fourteen languages of MMMLU; paraphrase quality varies by language and model capability, so gains may not transfer uniformly, especially to lower-resource languages. Fourth, we use the same model for rephrasing and answering, and decoupling them may yield further gains. Fifth, majority voting can amplify confidently wrong answers, so aggregated predictions should not be treated as calibrated confidence estimates in high-stakes settings without additional oversight. Finally, semantic TTA introduces additional latency and cost that may be prohibitive in some deployments, and since TTA extends rather than replaces self-consistency, tuning the balance of input-side and output-side diversity is a natural direction that we leave to future work.

References

Aryan Agrawal, Lisa Alazraki, Shahin Honarvar, and Marek Rei. Enhancing llm robustness to perturbed instructions: An empirical study. *arXiv preprint arXiv:2504.02733*, 2025.

Rui Ai, Yuqi Pan, David Simchi-Levi, Milind Tambe, and Haifeng Xu. Beyond majority voting: LLM aggregation by leveraging higher-order information. *arXiv preprint arXiv:2510.01499*, 2025.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. *arXiv preprint arXiv:2411.07279*, 2024.

Anthropic. Introducing Claude Haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5, 2025.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Jack Butler, Nikita Kozodoi, Zainab Afolabi, Brian Tyacke, and Gaiar Baimuratov. Finding the sweet spot: Trading quality, cost, and speed during inference-time llm reflection. *arXiv preprint arXiv:2510.20653*, 2025.

Yaping Chai, Haoran Xie, and Joe S. Qin. Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities. *arXiv preprint arXiv:2501.18845*, 2025.

Giannis Chatziveroglou, Richard Yun, and Maura Kelleher. Exploring llm reasoning through controlled prompt variations. *arXiv preprint arXiv:2504.02111*, 2025.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*, 2023.

Minjoon Choi. RoParQ: Paraphrase-aware alignment of large language models towards robustness to paraphrased questions. *arXiv preprint arXiv:2511.21568*, 2025.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*, 2023.

Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. *arXiv preprint arXiv:2405.18330*, 2024.

Jiaxin Guo, Daimeng Wei, Yuanchang Luo, Shimin Tao, Hengchao Shang, Zongyao Li, Shaojun Li, Jinlong Yang, Zhanglin Wu, Zhiqiang Rao, and Hao Yang. M-Ped: Multi-prompt ensemble decoding for large language models. *arXiv preprint arXiv:2412.18299*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2021.

Siyuan Huang, Zhiyuan Ma, Jintao Du, Changhua Meng, Weiqiang Wang, and Zhouhan Lin. Mirror-consistency: Harnessing inconsistency in majority voting. *arXiv preprint arXiv:2410.10857*, 2025.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of llms. *arXiv preprint arXiv:2410.08020*, 2024.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Grosse, and Jimmy Ba. Calibrating language models via augmented prompt ensembles. In *ICML 2023 Workshop on Deployment Challenges for Generative AI*, 2023.

Go Kamoda, Benjamin Heinzerling, Keisuke Sakaguchi, and Kentaro Inui. Test-time augmentation for factual probing. *arXiv preprint arXiv:2310.17121*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Helen Lu, Divya Shanmugam, Harini Suresh, and John Guttag. Improved text classification via test-time augmentation. *arXiv preprint arXiv:2206.13607*, 2022.

Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 2024.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594, 2023.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

Junichiro Niimi. Dynamic sentiment analysis with local large language models using majority voting: A study on factors affecting restaurant evaluation. *arXiv preprint arXiv:2407.13069*, 2024.

OpenAI. Multilingual massive multitask language understanding (MMMLU). HuggingFace Datasets, 2024. Dataset available at https://huggingface.co/datasets/openai/MMMLU.

Long Phan et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Silviu Pitis, Michael R. Zhang, Andrew Wang, and Jimmy Ba. Boosted prompt ensembles for large language models. *arXiv preprint arXiv:2304.05970*, 2023.

Mikhail Seleznyov, Mikhail Chaichuk, Gleb Ershov, Alexander Panchenko, Elena Tutubalina, and Oleg Somov. When punctuation matters: A large-scale comparison of prompt robustness methods for llms. *arXiv preprint arXiv:2508.11383*, 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. *arXiv preprint arXiv:2011.11156*, 2020.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint arXiv:2402.08115*, 2024.

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

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023.

Jan Philip Wahle, Terry Ruas, Yang Xu, and Bela Gipp. Paraphrase types elicit prompt engineering capabilities. *arXiv preprint arXiv:2406.19898*, 2024.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Vikas Yadav, Zheng Tang, and Vijay Srinivasan. Paraphrase and aggregate with large language models for minimizing intent classification errors. *arXiv preprint arXiv:2406.17163*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Yihang Yao, Zhepeng Cen, Miao Li, William Han, Yuyou Zhang, Emerson Liu, Zuxin Liu, Chuang Gan, and Ding Zhao. Your language model may think too rigidly: Achieving reasoning consistency with symmetry-enhanced training. *arXiv preprint arXiv:2502.17800*, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2024.

Chenrui Zhang, Lin Liu, Jinpeng Wang, Chuyuan Wang, Xiao Sun, Hongyu Wang, and Mingchen Cai. PREFER: Prompt ensemble learning via feedback-reflect-refine. *arXiv preprint arXiv:2308.12033*, 2023.

Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. *arXiv preprint arXiv:2404.11500*, 2024.

A Prompt Templates

A.1 Answering and Rephrasing Prompts

Section A.1 presents the answering and rephrasing prompts. The answering prompt is used by both the baseline and all TTA methods. The rephrasing prompt is used exclusively by semantic TTA to generate question variations.

Answering

Answer the following question. First, think through the problem step-by-step, then provide your final answer.
Guidelines:

- Think through your reasoning in <thinking></thinking> tags (use maximum 200 words)
- Provide your final answer inside <answer></answer> XML tags
  - For multiple-choice questions: Output only the letter (A, B, C, or D) in the answer tags
  - For mathematical questions: Output only the numerical answer or mathematical expression in the answer tags (e.g., <answer>16</answer> or <answer>\frac{1}{2}</answer>)
  - For open-ended questions: Output the specific answer requested in the answer tags

Question:
<question>
{question}
</question>
Your response format:
<thinking>
Your step-by-step reasoning in less than 200 words
</thinking>
<answer>
Your final answer here
</answer>

Rephrasing

You are given a question that may be either:

- A multiple-choice question with answer options (A, B, C, D)
- An open-ended question requiring a specific answer

Your task is to rephrase this question into {num_augmentations} different variations. Each variation should:

- Preserve the exact same meaning and correct answer
- Use different wording or sentence structure
- If there are answer choices, maintain them exactly as they are (same letters, same options)
- If the question includes specific output format instructions (e.g., "output as array", "use XML tags"), preserve these EXACTLY as written - do not paraphrase formatting requirements
- If the question references images (e.g., "<image 1>", "in the diagram"), maintain these references in the same form
- Keep all the essential information needed to answer the question

Output each rephrased question in XML tags: <q1></q1>, <q2></q2>, <q3></q3>, etc.
Do not explain your thinking. Do not add any information to your answer except for the rephrased questions.
Original Question:
<question>
{question}
</question>
Generate {num_augmentations} rephrased variations now:

A.2 Dataset-Specific Prompts

Section A.2 presents dataset-specific prompts that are inserted into the `{question}` placeholder of the answering and rephrasing templates.

Math500

{problem}
Provide your final answer in the simplest form.

IMDB Reviews

Classify the sentiment of the following review as either 'positive' or 'negative'.
Review: {review}

MMLU / MMMLU

{question}
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}

MMMU

{question}
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
[Images are provided as visual input to the model]

HLE (Multiple Choice)

{question}
Provide your answer as a single letter (A, B, C, or D).
[Images are provided as visual input to the model when available]

HLE (Exact Match)

{question}
Provide your answer in the simplest form.
[Images are provided as visual input to the model when available]

B Prediction Variance Analysis

Table 5 reports the standard deviation of prediction-level accuracy across dataset questions for each method. All aggregation methods reduce variance compared to single-call CoT prompting, with semantic TTA achieving the lowest variance on most benchmarks. The exception is HLE, where semantic TTA increases variance (17.06% → 19.00%). This likely stems from HLE's very low baseline accuracy (3%), where paraphrasing introduces additional variation without helping the model answer questions it fundamentally cannot solve.

Table 5: Standard deviation (%) of predictions across datasets and methods.

Dataset	CoT prompting	Semantic TTA	Lexical TTA	Self-consistency
HLE	17.06	19.00	17.73	**16.35**
IMDB	22.80	**20.73**	**20.73**	**20.73**
Math500	30.50	**26.89**	28.78	27.29
MMLU	32.79	**30.66**	**30.66**	**30.66**
MMMLU	40.19	**38.00**	39.03	39.23
MMMU	47.34	**46.61**	46.71	46.99

C Bootstrap Significance Analysis

As a non-parametric complement to the paired t-tests in Section 5.1, we run a paired bootstrap over the per-question accuracy gains relative to single-call CoT prompting. We pool the per-question gains across all six datasets (n = 2,400) and resample with replacement (5,000 resamples) to obtain a 95% confidence interval on the mean gain for each method. Table 6 reports the results.

The bootstrap confirms the parametric conclusions. Semantic TTA has the largest mean gain (+1.79 pp) and its confidence interval sits well above zero, matching the +1.8 pp average reported in Section 5.1. Both lexical TTA and self-consistency also clear zero, with the consistent ordering: input-side semantic TTA sits furthest above the noise floor, while self-consistency is the weakest, with its lower confidence bound only marginally positive.

Table 6: Paired bootstrap on pooled per-question accuracy gains over single-call CoT prompting (n = 2,400, 5,000 resamples). All methods clear zero; semantic TTA sits furthest above the noise floor.

Method	Mean gain (pp)	95% CI (pp)	t
Semantic TTA	**+1.79**	$[+0.88, +2.71]$	**3.86**
Lexical TTA	+1.25	$[+0.25, +2.21]$	2.47
Self-consistency	+0.92	$[+0.04, +1.79]$	2.06