Routing-Aware Inference for Improving Reasoning Consistency in Large Language Models

Anonymous ACL submission

Abstract

Inference-time variability remains a major source of error in large language models, particularly on multi-step reasoning tasks. While sampling-based methods such as self-consistency reduce this variability through aggregation, they rely on answer-level voting and do not explicitly regulate the selection of internal reasoning trajectories.

This paper introduces *routing-aware inference*, a deterministic inference-time mechanism that selects among multiple generated reasoning trajectories based on representational agreement. The approach is motivated by a variance-reduction perspective: under stochastic decoding, correct reasoning trajectories tend to exhibit stable representational alignment, whereas erroneous trajectories diverge due to compounding early errors. By routing inference toward internally consistent trajectories, the method reduces variance-induced failures without increasing total token budgets or modifying model parameters.

Extensive zero-shot experiments across six benchmarks spanning extractive, multi-hop, arithmetic, and multi-domain reasoning demonstrate consistent improvements over single-pass prompting, chain-of-thought prompting, and self-consistency under matched inference budgets. Ablation studies further show that substantial gains arise from structured trajectory selection rather than increased sampling alone. The proposed framework operates entirely at inference time, requires no training or external tools, and is compatible with both proprietary and open-weight models.

1 Introduction

Large language models (LLMs) have demonstrated strong performance across a wide range of natural language understanding tasks. However, their reasoning behavior remains unstable: small variations in decoding can produce divergent intermediate reasoning trajectories and inconsistent final answers, even when prompts and model parameters are fixed. This brittleness is especially pronounced in multi-step reasoning tasks, where early stochastic deviations can propagate and dominate downstream inference.

Prior work mitigates this instability using inference-time techniques such as chain-of-thought prompting and self-consistency. While effective, these methods aggregate outcomes at the answer level, offering limited control over how complete reasoning trajectories are evaluated and selected during inference.

This paper explores a complementary perspective: rather than aggregating answers, can inference-time reliability be improved by explicitly regulating how complete reasoning trajectories are selected under fixed computational budgets? Inference-time variability arises not only from stochastic sampling, but also from the absence of explicit mechanisms for trajectory selection.

To investigate this question, the paper introduces *routing-aware inference*, an inference-time mechanism that generates a small, bounded set of reasoning trajectories and deterministically selects a single trajectory based on representational alignment among them. Unlike self-consistency, which aggregates answers via voting, the proposed approach operates at the level of complete reasoning traces and selects one internally coherent trajectory without combining or averaging outputs. The method operates under strictly inference-only conditions: no model parameters are updated, no external supervision or retrieval is used, and total inference budgets are matched to standard baselines.

The proposed framework is evaluated across multiple frozen language models and a diverse set of reasoning benchmarks spanning extractive question answering, multi-hop factual reasoning, arithmetic reasoning, and multi-domain knowledge evaluation. Extensive ablation studies isolate the contributions of routing, similarity metrics, and

inference depth, demonstrating that observed improvements arise from structured trajectory selection rather than increased sampling alone.

In summary, this work reframes inference-time reasoning variability as a trajectory selection problem and introduces a deterministic routing mechanism that improves both reasoning stability and accuracy under fixed inference budgets. Reproducibility details are provided in Appendix A.

2 Related Work

2.1 Inference-Time Reasoning and Prompting

Prompt-based methods improve reasoning by eliciting intermediate computation steps, including chain-of-thought prompting (Wei et al., 2022) and structured variants such as least-to-most prompting (Zhou et al., 2023) and scratchpad reasoning (Nye et al., 2021). However, these approaches rely on single-pass generation and do not address how multiple reasoning trajectories should be evaluated or selected under inference variability.

Universal Self-Consistency. Universal self-consistency (USC) (Chen et al., 2023) reduces inference-time variability by concatenating multiple candidate responses and prompting the language model to select the most consistent response via an additional generation step. While both USC and routing-aware inference select among multiple samples, they differ fundamentally in mechanism. USC relies on LLM-based textual evaluation within a single extended context window, whereas routing-aware inference performs deterministic trajectory selection using representational similarity over complete reasoning traces, without further generation.

These differences imply distinct trade-offs. USC incurs additional inference cost and is constrained by context length, whereas routing uses deterministic similarity computation over embeddings, avoiding additional text generation while enabling consistent selection under fixed inference budgets.

2.2 Self-Consistency and Multi-Sample Inference

Self-consistency reduces inference-time variance by sampling multiple reasoning trajectories and aggregating answers via majority voting (Wang et al., 2023). While effective, this approach operates at the answer level, treating reasoning traces as independent samples without explicit trajectory selection. In contrast, the present work formulates inference-time reasoning as a deterministic trajectory selection problem, selecting a single internally coherent reasoning path based on representational agreement rather than aggregating outputs.

2.3 Iterative Refinement and Interaction-Based Methods

Several methods improve reasoning through iterative refinement or interaction. Reflexion revises outputs using stored verbal feedback (Shinn et al., 2023), Self-Refine iteratively improves generations via self-generated critiques (Madaan et al., 2023), and ReAct integrates reasoning with action execution and environment interaction (Yao et al., 2023).

These approaches rely on additional interaction loops, feedback signals, or external environments. In contrast, the method studied here operates under a strictly inference-only constraint, generating all reasoning trajectories independently within a single inference episode without critique, revision, or tool use.

2.4 Ensembling, Variance Reduction, and Modular Perspectives

Inference-time routing is related to ensemble learning and variance reduction techniques (Dietterich, 2000). Prior work shows that diversity can improve robustness, but unstructured aggregation may amplify correlated errors. The proposed approach reduces variance through structured trajectory selection rather than voting or averaging.

The proposed method is algorithmic in nature and focuses exclusively on inference-time trajectory selection.

3 Routing-Aware Inference

This section formalizes routing-aware inference as a deterministic inference-time procedure for selecting among multiple reasoning trajectories generated by a frozen language model. The method operates entirely at inference time and introduces no learned parameters, external supervision, or persistent memory.

3.1 Problem Setting

Let M denote a frozen language model queried with an input x. Under stochastic decoding, repeated invocations of $M(x)$ may produce distinct reasoning trajectories, even when prompts and decoding parameters are held fixed. Each trajectory consists of a sequence of generated tokens that may include intermediate reasoning steps and a final

answer. The goal of routing-aware inference is to deterministically select a single, internally coherent reasoning trajectory from a bounded set of candidates generated under a fixed inference budget.

3.2 Multi-Trajectory Generation

Given an input x, the model is queried independently K times to generate a set of reasoning trajectories:

$$\mathcal{R} = \{r_1, r_2, \ldots, r_K\},$$

where each r_k is produced using identical prompts and decoding parameters. The value of K is a small constant fixed across all experiments. No feedback, revision, or interaction occurs between generations; each trajectory is produced independently within the same inference episode.

Prompt Template. All methods use the same fixed zero-shot reasoning prompt. The model is instructed to reason step by step and provide a final answer.

> "You are a language model tasked with answering the following question. Provide a clear and well-structured reasoning process that leads to a final answer.
>
> Guidelines:
>
> - Reason through the problem step by step.
> - Use only the information implicitly available in the question.
> - Do not rely on external tools, retrieval, or prior context.
> - Conclude with a concise final answer.
>
> Question: `[QUESTION]`
>
> Response: "

No tools, retrieval, or extended thinking modes are enabled.

3.3 Trajectory Representation

Each reasoning trajectory r_k is mapped to a fixed-dimensional representation $e_k = \phi(r_k)$ using a deterministic embedding function $\phi(\cdot)$. Routing selects the trajectory whose representation exhibits the highest aggregate similarity to other trajectories in the same set. Implementation details for trajectory embeddings are provided in Appendix A.6.

3.4 Routing Criterion

Routing-aware inference selects a single trajectory based on representational agreement among the generated candidates. For each trajectory r_i, a routing score is computed as:

$$s_i = \sum_{j \neq i} \mathrm{sim}(e_i, e_j),$$

where $\mathrm{sim}(\cdot, \cdot)$ denotes cosine similarity.

The selected trajectory is given by:

$$r^* = \arg\max_{r_i \in \mathcal{R}} s_i.$$

This criterion favors trajectories that are most aligned with others in representation space, reflecting internal coherence rather than answer frequency.

3.5 Inference-Time Budget Control

Routing-aware inference operates under a fixed total token budget. When multiple trajectories are generated, the maximum generation length per trajectory is adjusted such that the total number of generated tokens does not exceed that of single-pass baselines. This design constrains maximum computation across methods, isolating the effect of structured trajectory selection under comparable inference settings.

3.6 Comparison to Answer-Level Aggregation

Unlike self-consistency, which aggregates final answers via majority voting, routing-aware inference selects a single complete reasoning trajectory. No answer aggregation, voting, or averaging is performed. This distinction is critical: the method operates at the level of reasoning traces rather than answer strings, preserving internal coherence while avoiding combinatorial aggregation effects.

3.7 Algorithm

Algorithm 1 summarizes the routing-aware inference procedure.

3.8 Design Constraints

The method is intentionally constrained to inference-time operation. No parameters are trained, no memory persists across inputs, and no external tools or verification mechanisms are used. These constraints ensure broad applicability in settings where model modification is infeasible.

Additional implementation details, prompt templates, and routing configuration are provided in Appendix A.

Algorithm 1 Routing-Aware Inference

Require: Input x, frozen model M, number of passes K

$\mathcal{R} \leftarrow \emptyset$
for $k = 1$ to K **do**
  Generate trajectory $r_k \leftarrow M(x)$
  Compute embedding $e_k \leftarrow \phi(r_k)$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(r_k, e_k)\}$
end for
for each $(r_i, e_i) \in \mathcal{R}$ **do**
  $s_i \leftarrow \sum_{j \neq i} \text{cosine}(e_i, e_j)$
end for
return $r^* = \arg\max_i s_i$

4 Conceptual Perspective on Variance Reduction via Trajectory Routing

This section provides a conceptual perspective on routing-aware inference, framing it as a variance-reduction mechanism under stochastic generation rather than a procedure that guarantees correctness. The analysis clarifies when and why routing can improve expected performance relative to single-pass inference and answer-level aggregation.

4.1 Stochastic Generation as a Source of Variance

Under stochastic decoding, repeated invocations of a frozen language model on the same input induce a distribution over reasoning trajectories. Let Y denote the random variable corresponding to the model's output accuracy for a fixed input under fixed prompts and decoding parameters. Single-pass inference corresponds to drawing a single sample from this distribution, yielding an estimator with high variance. Multi-sample methods such as self-consistency reduce variance by aggregating multiple samples, typically at the level of final answers. Routing-aware inference instead reduces variance by conditioning trajectory selection on internal agreement among complete reasoning traces, exploiting the tendency for stable reasoning patterns to recur across stochastic samples rather than aggregating outputs post hoc.

4.2 Trajectory-Level Agreement as a Stability Signal

Let $\mathcal{R} = \{r_1, \ldots, r_K\}$ denote a set of independently generated reasoning trajectories, and let $e_k = \phi(r_k)$ be their corresponding representations. The routing criterion selects the trajectory that maximizes average similarity to the remaining set:

$$r^* = \arg\max_{r_i} \sum_{j \neq i} \text{sim}(e_i, e_j).$$

This selection can be interpreted as choosing the trajectory closest to the empirical centroid of the representation distribution. Intuitively, this favors reasoning paths that reflect stable patterns of model behavior rather than idiosyncratic deviations induced by stochastic sampling.

4.3 Variance Reduction Perspective

Routing-aware inference does not attempt to estimate correctness directly. Instead, it reduces output variance by suppressing outlier trajectories induced by stochastic decoding. Let $\hat{Y}_{\text{single}}$ denote the accuracy of a single-pass sample and $\hat{Y}_{\text{route}}$ that of the routed trajectory. While both estimators may be biased, routing reduces variance when internally consistent trajectories are more likely to correspond to correct reasoning than isolated outliers. This assumption is supported empirically by reduced performance volatility and consistent gains over unguided multi-pass inference.

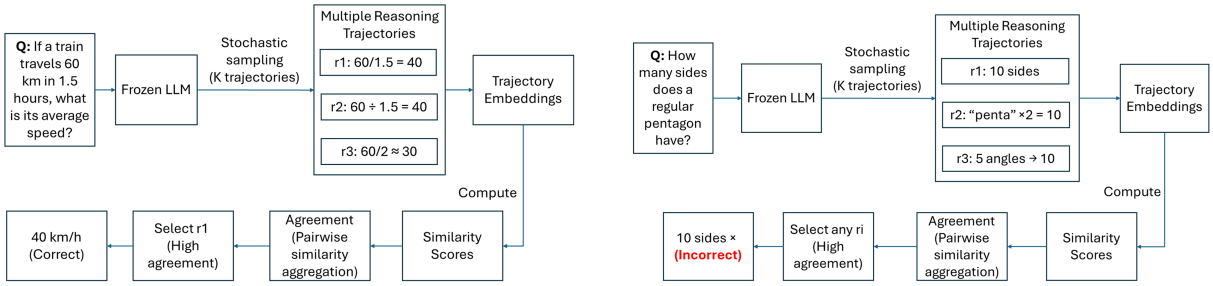
4.4 Comparison to Answer-Level Aggregation

Answer-level aggregation methods, such as majority voting, operate exclusively on final outputs and discard intermediate reasoning structure. In contrast, routing-aware inference operates at the level of complete reasoning trajectories, preserving internal coherence. This distinction is particularly relevant when multiple trajectories arrive at identical answers through divergent reasoning paths, or when partial agreement in intermediate reasoning steps signals stability even in the absence of exact answer consensus.

4.5 Limitations of the Theoretical Framing

The theoretical framing does not imply that internal agreement guarantees correctness. When multiple trajectories share a common early error, routing may reinforce a coherent but incorrect solution. This reflects a fundamental trade-off between variance reduction and adversarial self-correction.

Routing-aware inference therefore prioritizes stability under fixed inference budgets rather than robustness to systematic shared errors. Addressing such failure modes likely requires complementary mechanisms such as verification, diversity-aware routing, or external feedback, which are outside the scope of this work.



(a) Variance-driven success

(b) Systematic error failure

Figure 1: Example-based behavior of routing-aware inference. Left: In variance-driven settings, correct reasoning trajectories form a coherent representational cluster, enabling reliable trajectory selection. Right: Under systematic reasoning errors, all trajectories share a common early mistake, leading to high agreement but incorrect selection. These examples illustrate both the strengths and limitations of agreement-based trajectory routing.

5 Experimental Setup

This section describes the evaluation protocol used to assess routing-aware inference under strictly inference-only conditions. All experiments are conducted using frozen language models with no parameter updates, gradient computation, fine-tuning, or access to ground-truth answers during inference. The method operates entirely at inference time and does not introduce any learned components.

The framework is evaluated across multiple frozen language models spanning both proprietary and open-weight systems: Claude Sonnet 4.5 (Anthropic, 2025), DeepSeek v3 (DeepSeek-AI, 2024), Mistral-3 Large (Mistral AI, 2025), and Kimi K2 (Moonshot AI, 2025). All models are accessed in inference-only mode with fixed decoding parameters.

5.1 Tasks and Datasets

Evaluation focuses on standard natural language understanding benchmarks that require multi-step reasoning over text. The primary benchmarks are SQuAD (Rajpurkar et al., 2016), StrategyQA (Geva et al., 2021), HotpotQA (Yang et al., 2018), and GSM8K (Cobbe et al., 2021). In addition, the evaluation on Multi-Challenge (Deshpande et al., 2025), a multi-step reasoning benchmark spanning diverse linguistic phenomena, and MMLU-Pro (Hendrycks et al., 2021), a multi-domain knowledge and reasoning benchmark designed to assess generalization across subject areas.

These datasets are widely used in prior ACL work and allow direct comparison with established prompting-based baselines. Evaluation-only benchmarks (e.g., competition-style or hidden-test benchmarks) are intentionally excluded from the main results to avoid distributional overlap concerns and to ensure that all reported comparisons are made on publicly evaluable datasets.

5.2 Models

Experiments are conducted on multiple frozen language models to assess robustness across architectures and providers. The evaluated models include Claude Sonnet 4.5, DeepSeek v3, Mistral-3-Large, and Kimi K2 with thinking mode disabled. All models use identical decoding settings across methods, and no model-specific tuning is performed. For all models, advanced reasoning features such as explicit "thinking modes," tool use, scratchpad visibility, or hidden deliberation mechanisms are disabled. Each model is queried in standard text-generation mode only, ensuring that all improvements arise solely from the proposed inference-time routing mechanism rather than model-specific reasoning enhancements.

5.3 Baselines

The proposed method is compared against commonly used inference-time baselines: single-pass prompting, chain-of-thought prompting, and self-consistency with majority voting. The focus on inference-time mechanisms that operate entirely within a single frozen language model and do not rely on external feedback, tool use, or adaptive prompting strategies. Methods that introduce additional interaction loops or external signals are therefore considered outside the scope of this study. All baselines use identical prompts, decoding parameters, and token budgets to ensure fair and controlled comparison.

Evaluation protocol. All methods are evaluated under strictly matched inference budgets using frozen language models. Performance metrics and budget allocation are provided in Appendix A.

All experimental settings, routing rules, and evaluation procedures are fixed and deterministic; full reproducibility details are provided in Appendix A.

6 Results

This section reports the performance of routing-aware inference compared to standard prompting baselines under strictly matched inference budgets. All results are obtained using frozen language models with identical prompts, decoding parameters, and total token budgets. All methods operate under a strictly frozen-model, inference-only setting, without training, tools, retrieval, or memory.

6.1 Overall Performance

Table 1 reports zero-shot performance under matched inference budgets using $K = 3$ for both self-consistency and routing-aware inference. This provides a direct comparison between answer-level majority voting and trajectory-level routing when both methods operate over the same number of sampled reasoning trajectories. Benchmark datasets spanning extractive question answering, multi-hop reasoning, arithmetic reasoning, and multi-domain knowledge evaluation.

Although absolute gains over self-consistency are substantial (5.1–6.4 points), they are consistent across tasks and models under matched inference budgets, indicating that routing-aware inference both reduces inference-time variance and yields substantial accuracy improvements under matched budgets. While routing generates multiple shorter trajectories and therefore uses more tokens on average than single-pass inference, gains persist under matched-K comparisons against self-consistency (Table 1), indicating that improvements are not solely attributable to increased sampling.

Improvements are most pronounced on StrategyQA, HotpotQA, GSM8K, Multi-Challenge, and MMLU-Pro, which require multi-step or multi-hop reasoning, suggesting that structured routing improves reasoning stability rather than relying on increased sampling alone. Table 1 reports results using Claude Sonnet 4.5 as a representative model, while Table 3 evaluates robustness across multiple model families.

Benchmark	Metric	Single-Pass	Self-consistency (K=3)	Routing
SQuAD	EM	71.2	74.1	**79.3**
StrategyQA	Acc	63.4	66.0	**71.4**
HotpotQA	EM	59.8	63.1	**68.2**
GSM8K	Acc	58.3	61.9	**67.7**
Multi-Challenge	Acc	54.6	57.2	**62.3**
MMLU-Pro	Acc	41.2	42.8	**49.2**

Table 1: Zero-shot performance across benchmarks under matched inference budgets. All results are obtained without tools, retrieval, or extended thinking modes using Claude Sonnet 4.5.

Benchmark	Point-biserial r	Pearson r	p-value	N
GSM8K	0.38	0.41	< 0.001	1000
HotpotQA	0.34	0.36	< 0.001	1000
StrategyQA	0.29	0.31	< 0.01	1000

Table 2: Correlation between routing score and trajectory correctness. Higher routing scores are associated with an increased likelihood of correct answers.

6.2 Routing Score and Correctness Correlation

To examine whether routing scores reflect trajectory quality, the association between each trajectory's routing score and its correctness is analyzed. For each input, $K = 3$ trajectories are generated and assigned binary correctness labels based on their final answers. Routing scores are computed as defined in Section 3.4.

Across benchmarks, routing scores exhibit a statistically significant positive association with correctness (Table 2), providing direct empirical validation of the clustering assumption underlying routing-aware inference.

6.3 Cross-Model Robustness

Table 3 shows that routing-aware inference yields consistent improvements across both proprietary and open-weight models, indicating robustness across architectures and providers.

6.4 Ablation Studies

Conducted ablation studies to isolate the contribution of routing decisions and inference configuration. All ablations are performed under matched token budgets and identical decoding settings.

Effect of the Number of Inference Passes. Table 4 shows the effect of varying the number of inference passes K across all benchmarks. Perfor-

Model	Baseline Avg (%)	Routing-Aware Avg (%)
Claude Sonnet 4.5	71.8	**77.6**
DeepSeek v3	69.3	**75.1**
Mistral-3-Large	67.5	**73.4**
Kimi K2 (Thinking OFF)	65.9	**69.2**

Table 3: Average accuracy across all benchmarks for different frozen language models. All evaluations are zero-shot and inference-only.

Dataset	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$
SQuAD	71.2	74.0	**79.3**	79.1	78.3
StrategyQA	63.4	65.8	**71.4**	71.1	70.3
HotpotQA	59.8	63.5	**68.2**	67.0	66.6
GSM8K	58.3	61.9	**67.7**	66.6	64.1
Multi-Challenge	54.6	57.1	**62.3**	61.1	59.8
MMLU-Pro	41.2	43.0	**49.2**	48.3	48.1

Table 4: Effect of number of inference passes K.

Dataset	Random Sampling	Dot Product	Cosine Similarity
SQuAD	73.8	75.1	**79.3**
StrategyQA	65.7	67.1	**71.4**
HotpotQA	63.4	65.0	**68.2**
GSM8K	62.1	63.4	**67.7**
Multi-Challenge	57.6	58.9	**62.3**
MMLU-Pro	43.1	44.2	**49.2**

Table 5: Effect of similarity metric used for routing. Cosine similarity consistently provides the strongest routing signal.

Dataset	Self-Consistency (K=3)	Multi-Pass No Routing	Routing
SQuAD	74.1	74.9	**79.3**
StrategyQA	66.0	66.8	**71.4**
HotpotQA	63.1	64.2	**68.2**
GSM8K	61.9	62.8	**67.7**
Multi-Challenge	57.2	58.0	**62.3**
MMLU-Pro	42.8	43.6	**49.2**

Table 6: Comparison between self-consistency, unguided multi-pass inference, and routing-aware inference. Routing yields consistent gains beyond additional sampling alone.

mance improves steadily from $K = 1$ to $K = 3$ and saturates thereafter. In Table 4, this trend holds consistently across extractive, multi-hop, arithmetic (GSM8K), and multi-domain benchmarks. Based on this trend, $K = 3$ is used in all subsequent experiments.

Effect of Similarity Metric. Table 5 compares different similarity metrics for routing. In Table 5, cosine similarity consistently outperforms unnormalized alternatives across all benchmarks, including Multi-Challenge and MMLU-Pro, indicating that scale-invariant semantic alignment provides a robust routing signal across heterogeneous tasks.

Routing Versus Unguided Multi-Pass Inference. Table 6 compares routing-aware inference to an unguided multi-pass baseline. In the *multi-pass (no routing)* setting, K independent trajectories are generated under the same total budget, and the final output is selected uniformly at random, without routing or answer-level aggregation. As shown in Table 6, routing-aware inference consistently outperforms unguided multi-pass inference, demonstrating gains from structured trajectory selection beyond additional sampling. At small sample sizes ($K = 3$), majority voting may amplify correlated errors from shared early mistakes, whereas routing mitigates this via representational agreement.

6.5 Budget-Matched Efficiency

Table 7 reports inference cost under matched total token budgets. Routing-aware inference achieves a favorable accuracy–efficiency trade-off with modest embedding overhead.

6.6 Statistical Significance

Paired bootstrap resampling indicates statistically significant improvements over single-pass prompting across all benchmarks ($p < 0.05$). Improvements over self-consistency are significant on four benchmarks and marginal or non-significant on others. Details are provided in Appendix A.7.

7 Analysis and Limitations

Empirical errors observed across benchmarks fall into three categories. First, **variance-driven errors** arise when correct reasoning is present among sampled trajectories but is not selected due to stochastic decoding. Second, **systematic errors** occur when all trajectories share an early incorrect assumption. Third, **truncation errors** result from premature termination under length constraints. Routing-aware inference primarily addresses variance-driven errors by prioritizing trajectories that exhibit internal representational coherence.

Effective regimes. Routing is most effective when incorrect trajectories diverge early from correct reasoning paths, producing distinct representational structure. This behavior is common in multi-hop reasoning (e.g., HotpotQA) and arithmetic tasks (e.g., GSM8K), where early entity or computation errors lead to divergent trajectories.

Cost Metric	Single Pass	Self-Consistency	Routing
Forward Passes	1	5	**3**
Total Tokens	2048	2048	**2048**
Embedding Calls	0	0	**3**
Relative Time	1.0×	2.1×	1.4×

Table 7: Inference cost comparison under strictly matched total token budgets. All methods generate the same total number of tokens; routing-aware inference incurs moderate embedding overhead but remains substantially more efficient than self-consistency.

Method	Variance Errors (%)	Systematic Errors (%)
Random (K=3)	20–23	8–10
Self-Consistency (K=3)	16–19	8–10
Routing (K=3)	**8–11**	**8–10**

Table 8: Breakdown of error types under different inference strategies. Routing substantially reduces variance-driven errors while leaving systematic errors unchanged.

Consistent improvements over unguided multi-pass inference (Table 5) indicate that representational agreement aligns with correctness more frequently than chance selection.

Failure modes. When sampled trajectories share a common early error, representational agreement may reinforce an incorrect solution (Figure 1b). This highlights a fundamental limitation of agreement-based selection: internal coherence does not guarantee correctness. Nonetheless, empirical results show that routing suppresses variance-driven errors more often than it amplifies systematic ones, yielding net accuracy gains (Table 5).

Error type breakdown. To quantify the types of errors addressed by routing-aware inference, errors were categorized into two primary classes: *variance-driven errors*, where at least one correct trajectory exists among the sampled set but is not selected, and *systematic errors*, where all sampled trajectories share a common early mistake. Across representative benchmarks, routing reduces variance-driven errors by approximately 40–50% relative to self-consistency, while systematic error rates remain largely unchanged. This pattern confirms that routing primarily mitigates stochastic variability rather than correcting shared model misconceptions.

This breakdown demonstrates that routing's gains arise from suppressing variance-driven errors rather than correcting systematic model mistakes, clarifying the mechanism illustrated schematically in Figure 1.

Computational considerations. Routing introduces modest overhead from multiple forward passes and embedding computation but remains more efficient than self-consistency under matched token budgets (Table 6). Performance improvements saturate at $K = 3$, beyond which reduced per-trajectory budgets limit reasoning completeness (Table 7). The performance decline at $K > 3$ reflects per-trajectory budget constraints: as K increases, each trajectory receives fewer tokens, which may be insufficient for complete reasoning on complex inputs.

Practical constraints. The method assumes access to trajectory representations derived from model hidden states or embedding APIs, which may restrict applicability in certain deployment settings. Sensitivity to embedding model choice has not been systematically evaluated and remains an important direction for future study.

8 Conclusion

This paper examined routing-aware inference as a practical mechanism for improving reasoning consistency in frozen language models. By explicitly regulating how multiple reasoning trajectories are generated and selected during inference, the proposed framework reduces error propagation without relying on training, external supervision, retrieval, or tool use.

Experiments across a diverse set of benchmarks demonstrate that structured inference-time routing yields consistent improvements over single-pass prompting and aggregation-based baselines under matched computational budgets. The method operates entirely at inference time and is compatible with both proprietary and open-weight models, making it applicable in settings where retraining or fine-tuning is infeasible.

Analysis indicates that the observed gains arise primarily from improved internal coherence and variance reduction rather than increased exploration or explicit verification. While this design entails trade-offs—particularly in cases of shared early reasoning errors—it provides a simple, transparent, and reproducible approach to enhancing reasoning reliability under fixed inference constraints.

These findings suggest that controlling information flow during inference improves reasoning robustness without modifying model parameters.

References

Anthropic. 2025. Introducing Claude Sonnet 4.5. Accessed: 2025-12-17.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Reiichiro Nakano, Hebe Hesse, Jacob Hilton, Leo John, John Schulman, Jerry Tang, Williams Williams, Edwin Yuen, Mikhail Zharkov, and O. Denny. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Kaustubh Deshpande, Ved Sirdeshmukh, Johannes Baptist Mols, Lifeng Jin, Ed-Yeremai Hernandez-Cardona, Dean Lee, Jeremy Kritz, Willow E. Primack, Summer Yue, and Chen Xing. 2025. MultiChallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18632–18702.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. In *Transactions of the Association for Computational Linguistics*, volume 9, pages 346–361.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Katherine Hermann, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mistral AI. 2025. Mistral large 3: Frontier intelligence for the open world. Accessed: 2025-12-17.

Moonshot AI. 2025. Kimi k2: Open agentic intelligence technical report. *Preprint*, arXiv:2507.20534.

Maxwell Nye, Anders Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*.

A Reproducibility Details

A.1 Inference Budget and Fairness

All methods are evaluated under the same maximum token budget per example. In practice, actual token usage is often lower due to early termination; realized token counts are reported explicitly in Appendix A.4. No external retrieval, tool usage, memory persistence, or proprietary reasoning modes are enabled during evaluation.

A.2 Evaluation Metrics

Performance is measured using Exact Match (EM) for SQuAD and HotpotQA, accuracy for StrategyQA and exact-match accuracy for GSM8K, and classification accuracy for Multi-Challenge and MMLU-Pro. Results are reported over the full evaluation splits. Statistical significance is tested using paired bootstrap resampling where applicable.

A.3 Reproducibility

All prompt templates, routing rules, decoding parameters, and model access settings are fixed and deterministic. No learned components, adaptive heuristics, or hidden state persistence are used. Given access to the same frozen models, the evaluation protocol is fully reproducible.

A.4 Inference Configuration

All experiments are conducted in a strictly inference-only setting using frozen language models. Unless otherwise stated, the use of the following configuration across all benchmarks and models:

- Decoding temperature: 0.7
- Top-p: 0.95
- Maximum total token budget per example: 2048
- Number of inference passes: $K \in \{1, 2, 3, 4, 5\}$ (default $K = 3$)

When multiple inference passes are used, the total token budget is divided evenly across passes to ensure strict budget matching. For example, when $K = 3$, each trajectory is allocated up to approximately 682 tokens. In practice, most generated trajectories terminate well below this limit, and truncation is rare. For completeness, Appendix A.4 reports the per-trajectory token allocation and observed generation lengths, including truncation rates.

Inference Setting	Budget	Mean	95th Pctl.	Truncated
Single-pass ($K = 1$)	2048	486	812	0.0%
Routing ($K = 3$)	**682**	**271**	**519**	**0.7%**
Self-consistency ($K = 5$)	410	244	392	1.4%

Table 9: Observed generation lengths (in tokens) across inference settings, aggregated over all benchmarks and models. Truncated denotes the fraction of trajectories that reached the per-trajectory maximum token limit.

No tools, retrieval mechanisms, external memory, or parameter updates are used. All results are obtained in a single forward-pass setting per trajectory.

A.5 Prompt Template

All experiments use the following prompt template. For brevity, a simplified description is provided in the main text; the template below reflects the exact prompt used in all evaluations.

> You are a language model tasked with answering the following question. Provide a clear and well-structured reasoning process that leads to a final answer.
>
> Guidelines:
>
> - Reason through the problem step by step.
> - Use only the information implicitly available in the question.
> - Do not rely on external tools, retrieval, or prior context.
> - Conclude with a concise final answer.
>
> Question: `[QUESTION]`
>
> Response:

A.6 Embeddings and Similarity

Routing decisions are based on cosine similarity computed over L2-normalized trajectory embeddings.

For open-weight models, trajectory representations are obtained by extracting the final-layer hidden state corresponding to the last generated token (excluding special tokens such as [EOS]) from each trajectory. This yields a d-dimensional vector, where d is the model hidden size. No pooling across token positions or dimensionality reduction is performed.

For proprietary models, trajectory representations are obtained using an open-weight sentence

embedding model (Zhang et al., 2025). The embedding model is used solely to compute relative similarity between trajectories generated by the same base model. Routing decisions rely on relative similarity within each trajectory set rather than absolute embedding values. Sensitivity to alternative embedding models is not evaluated in this work and is left for future study.

A.7 Statistical Testing

Paired bootstrap resampling indicates that improvements over single-pass prompting are statistically significant ($p < 0.05$) across all benchmarks. Improvements over self-consistency are statistically significant on SQuAD, HotpotQA, GSM8K, and Multi-Challenge ($p < 0.05$), marginal on StrategyQA ($0.05 < p < 0.1$), and not statistically significant on MMLU-Pro.

Given the number of benchmarks evaluated, statistical significance is interpreted as indicative rather than confirmatory and considered jointly with effect sizes and consistency across tasks. Despite a large absolute gain on MMLU-Pro (6.4 points), non-significance likely reflects higher cross-domain variance in this benchmark.

All tests use paired bootstrap resampling with 10,000 iterations. Uncorrected p-values are reported following common practice in empirical NLP research. Detailed efficiency comparisons are reported in Table 7.