ALGOVERSE

NEURAL INFORMATION PROCESSING SYSTEMS

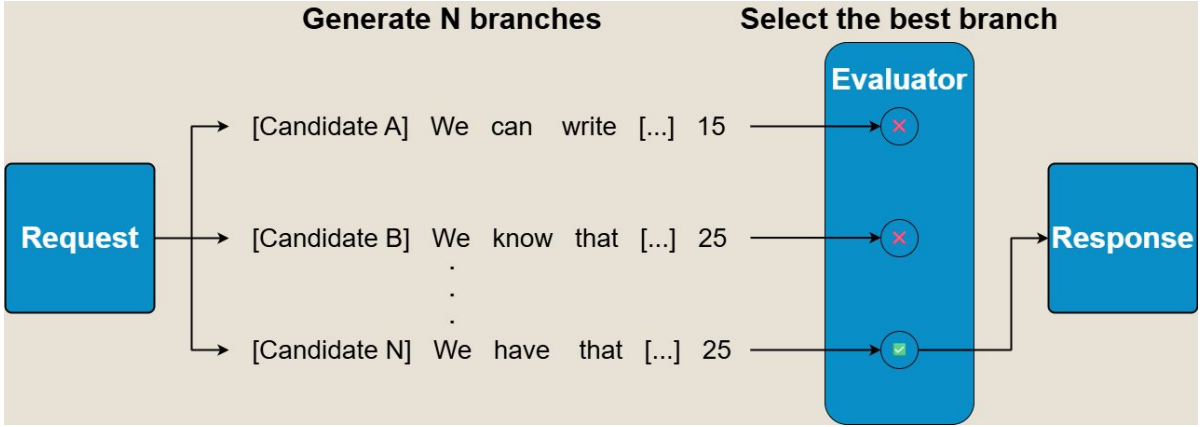
Inference-Time Chain-of-Thought Pruning with Latent Informativeness Signals

NeurIPS 2025 Poster Presentation

Sophie Li[1]*, Nicholas Huang[2]*, Nayan Saxena*, Nina Luo[3], Vincent Lin[4], Kevin Zhu[5], Sunishchal Dev[5]

[1]Columbia University, [2]University of British Columbia, [3]Harvey Mudd College, [4]University of Florida, [5]Algoverse AI Research

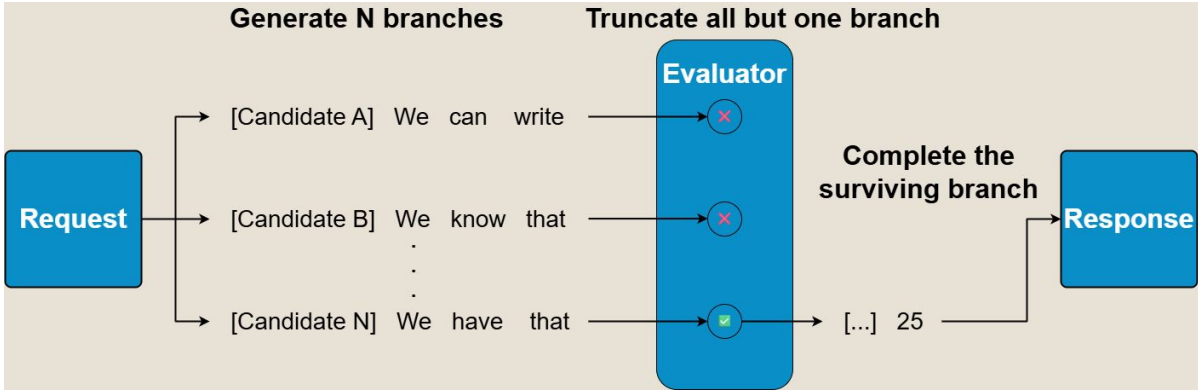
Inference-Time Scaling



Best-of-N (BoN) sampling

- N sequences are sampled and scored post-hoc.
- The best is typically chosen using self-consistency or reward models.
- Expensive because all N branches must be fully generated.

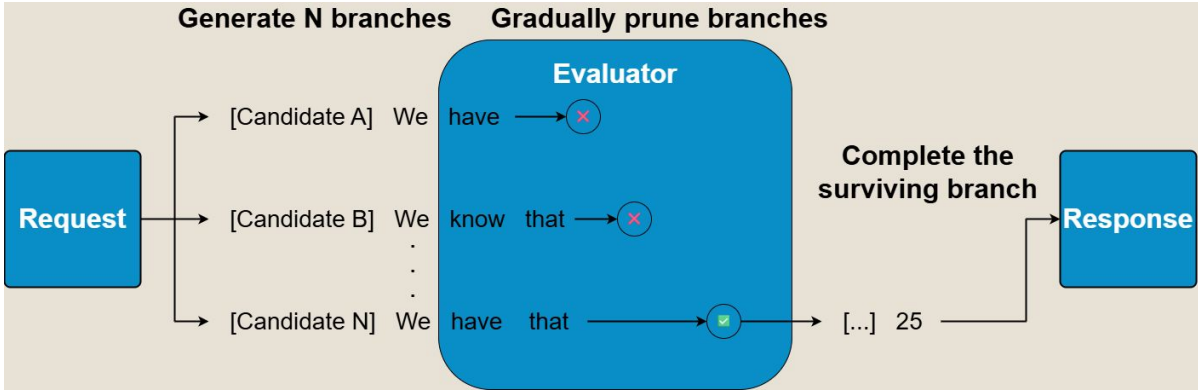
Inference-Time Scaling



Self-Truncation-BoN

- Truncates all but one branch early based on consistency.
- Significantly reduces cost.
- Does not directly estimate branch quality.

Inference-Time Scaling



Information-theoretic signals

- Kullback-Leibler divergence, confidence, and entropy
- Principled scoring
- Progressive pruning

Motivation

1. Reduce BoN's large token and memory costs without sacrificing accuracy.
2. Move beyond consistency-based pruning and develop a principled, training-free scoring mechanism.
3. Combine inference-time exploration with efficient branch elimination to stabilize reasoning in smaller models and scale compute more effectively.
4. Address the absence of branch-quality evaluation in ST-BoN

Key Contributions

We introduce **KL-Adjusted Pruned Path Algorithm (KAPPA)**, a novel sampling algorithm whose key features are as follows:

1. **Exploration vs. efficiency:** Diversity is encouraged during the draft phase.
2. **Uncertainty as a self-supervised signal:** KL divergence enables training-free principled scoring.
3. **Pruning schedule:** Progressively pruning branches eliminates unpromising branches earlier.

KL-Adjusted Pruned Path Algorithm

Draft Phase: Generate N branches until the earliest estimation time c when all branches are pairwise inconsistent.

Scoring Phase: At each token step t after c and for each branch i:

Compute KL divergence to an unconditional reference distribution: $D_t^i = D_{KL}(p_t^i \,\|\, q)$ and estimate information change: $\Delta I_t^i = D_t^i - D_{t-1}^i$

Median-of-Means: Partition ΔI over the last w steps into m equal-size buckets to obtain $\Delta \hat{I}_t^i$

Apply bias-corrected EMA smoothing with rate α: EMA_t^i

Compute uncertainty signals: confidence $C_t^i = \max_v p_t^i(v)$ and entropy H_t^i

Normalize each signal across alive branches at time t (z-score, clamp to [-3, 3]).

Form instantaneous weighted score: $s_t^i = w_{KL} \cdot EMA_t^i \cdot w_C \cdot \hat{C}_t^i + w_H \cdot \hat{H}_t^i$

Update scores by assigning greater weight to recent tokens.

Sample one-step continuation $y_{t+1}^i \sim p_\theta(\cdot \mid x, y_{1:t}^i)$ for the next round.

Gating Phase: After computing the scores of all alive branches at each time step t, prune the lowest-scoring branch every τ / N steps.

Continuation Phase: The final surviving branch is decoded until EOS.

Experimental Results

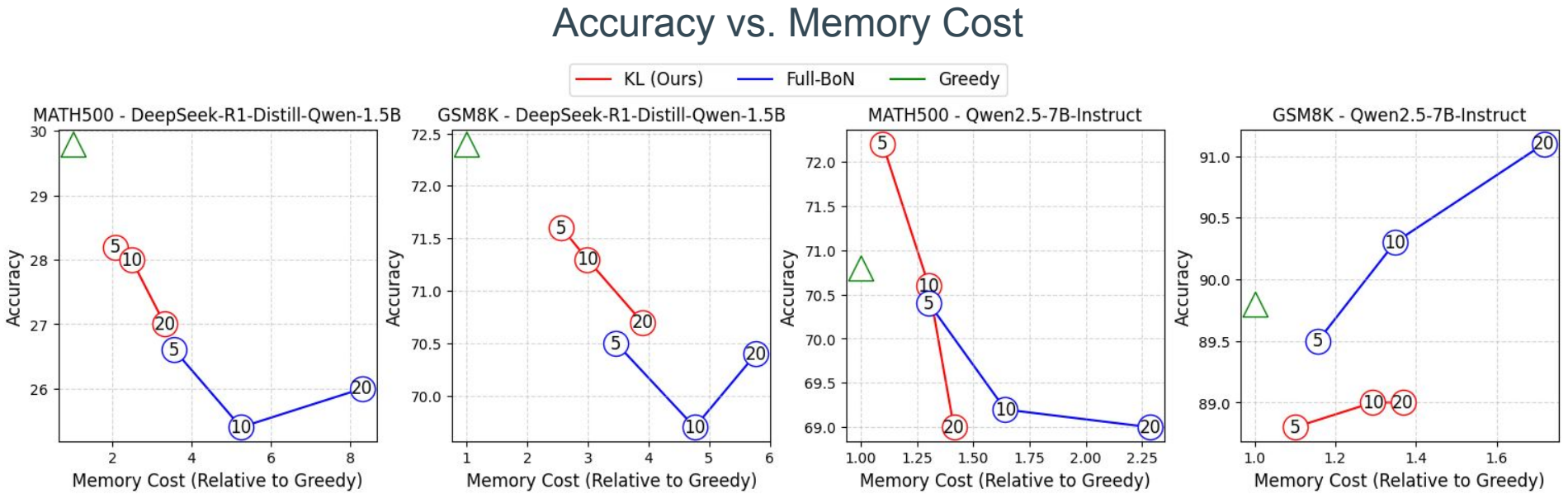


Figure 1: The computational cost and accuracy results in two LLMs across two mathematical and reasoning datasets as labeled. Each point on each polyline represents different sampling sizes N = 5, 10, 20 from left to right.

Peak Memory Reduction Ratio vs. Sampling Size

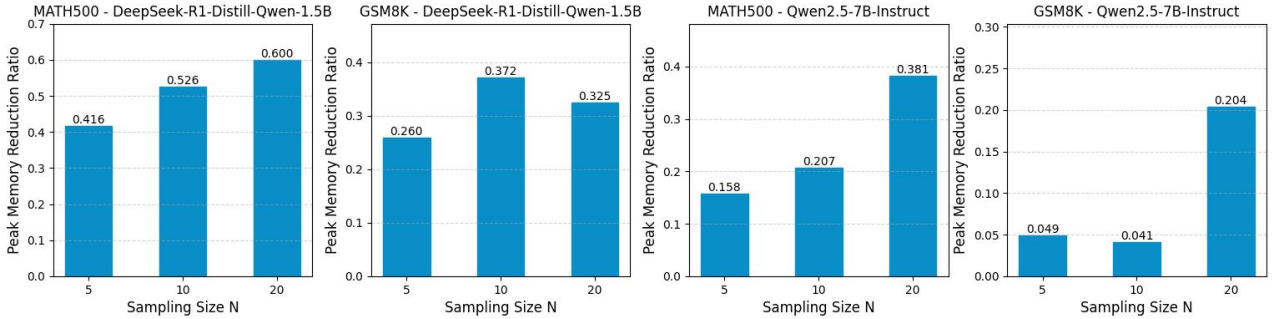


Figure 2: The computed peak memory reduction ratio under different sampling sizes N.

Token Reduction Ratio vs. Sampling Size

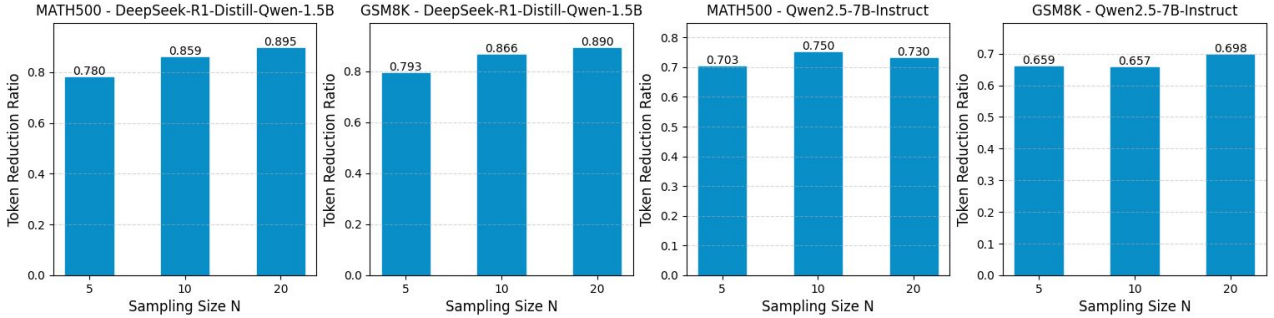


Figure 3: The computed token reduction ratio under different sampling sizes N.

Future Work

1. Experiment with less aggressive pruning schedules, such as a cosine schedule

2. Explore a dynamic pruning horizon τ that is based on problem complexity.

3. Conduct further experiments with other models and datasets, such as commonsense reasoning and theorem proving datasets.

4. More extensive hyperparameter tuning due to large number of variables

Thank You