LEAP: Zone-Aware MCTS for LLM Self-Speculative Decoding

Leiquan Zheng, Yuan Liu

LEAP: Zone-Aware MCTS for LLM Self-Speculative Decoding

South China University of Technology

Why self-SD?

Speculative decoding drafts tokens first, then verifies them in parallel with the target model.

Self-speculative decoding constructs the drafter from a subset of the target model layers: no extra module and no training.

The key questions are:

- Which layers should be executed, skipped or repeated?
- How to optimize them online with SD feedback?

Limitations of existing self-SD

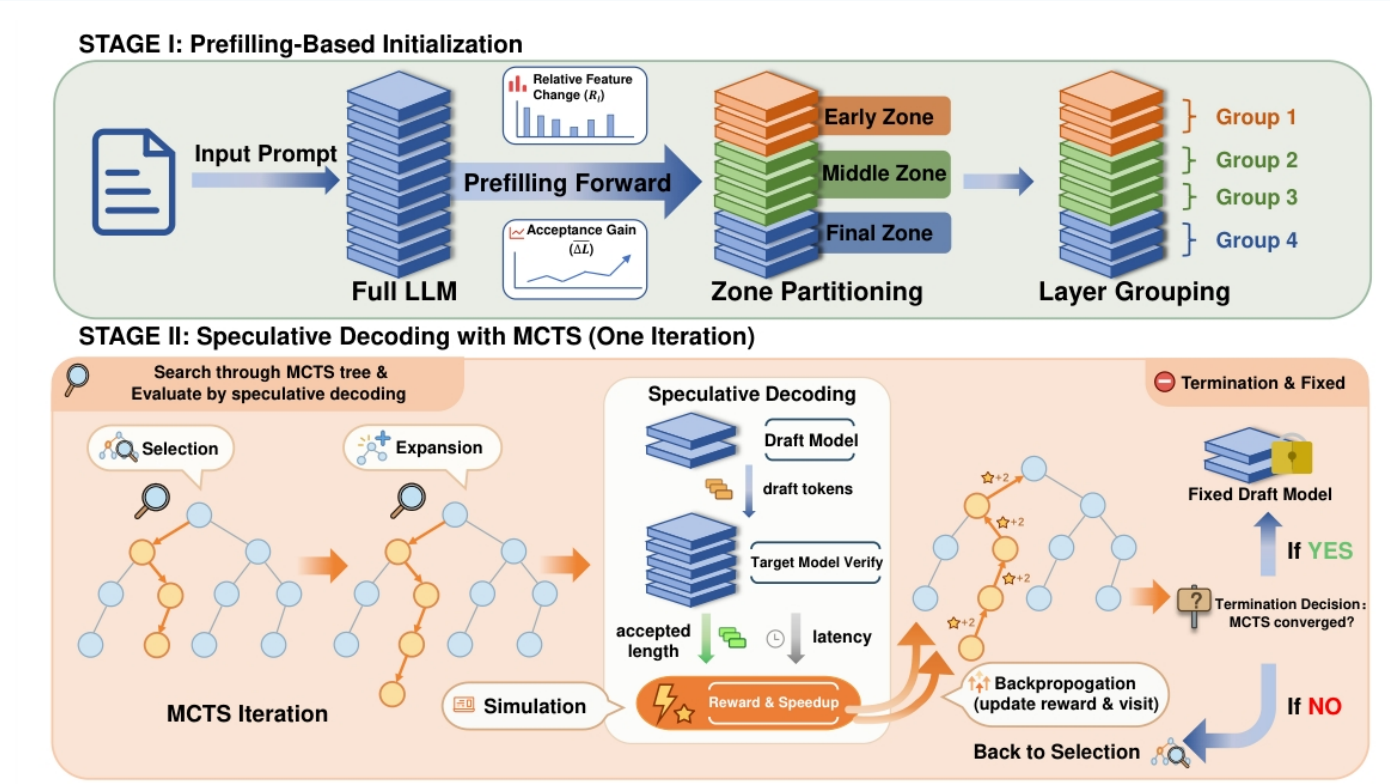
Fixed skip ratios limit adaptability across tasks and prompts.

Coarse search treats the whole configuration as a black box.

Delayed optimization across instances weakens early-stage speedup.

LEAP in one sentence:

Use prefilling-derived redundancy to structure the search space, then use MCTS to optimize layer-group actions with real-time speedup feedback.

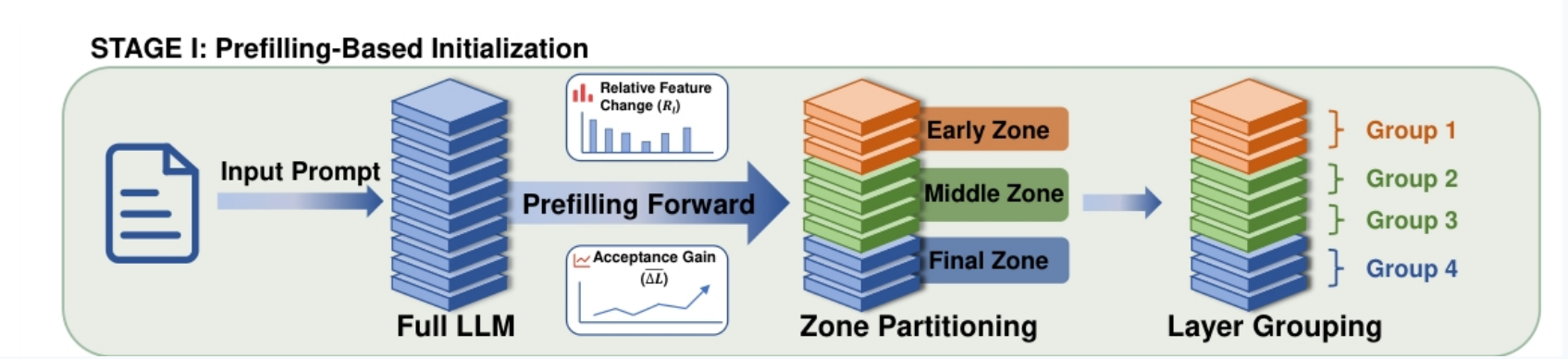


Two-stage workflow: prefilling-based initialization → online MCTS search during speculative decoding

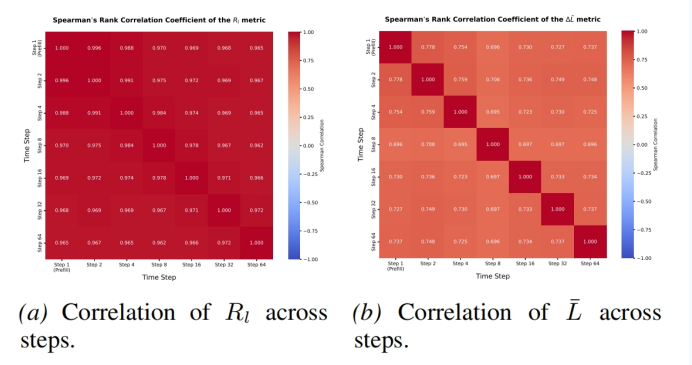
Stage I: Prefilling-Based Initialization

Compute once, partition layers into zones, and group layers as MCTS decision units.

South China University of Technology



Observation 1: Prefilling redundancy stays informative during decoding



(a) Correlation of R_l across steps. *(b)* Correlation of $\bar{L}$ across steps.

Observation 2: Two complementary redundancy metrics

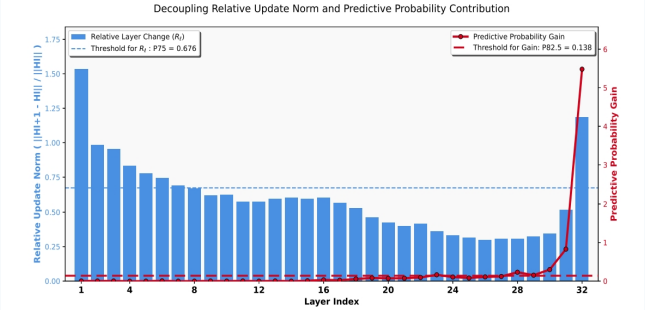
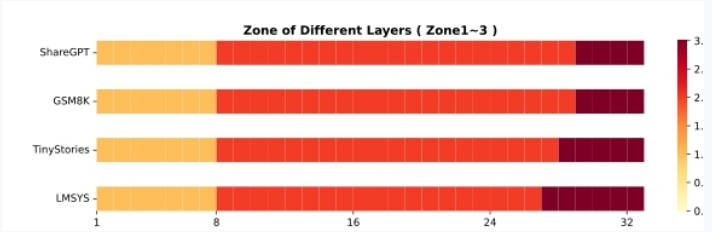


Figure 2. Two redundancy metrics of all layers.

Observation 3: Layer redundancy is zone-wise



Initialization outcome

- Early zone: preserve foundational feature transformation.
- Middle zone: likely redundant, good skip candidates.
- Final zone: important for acceptance and alignment.

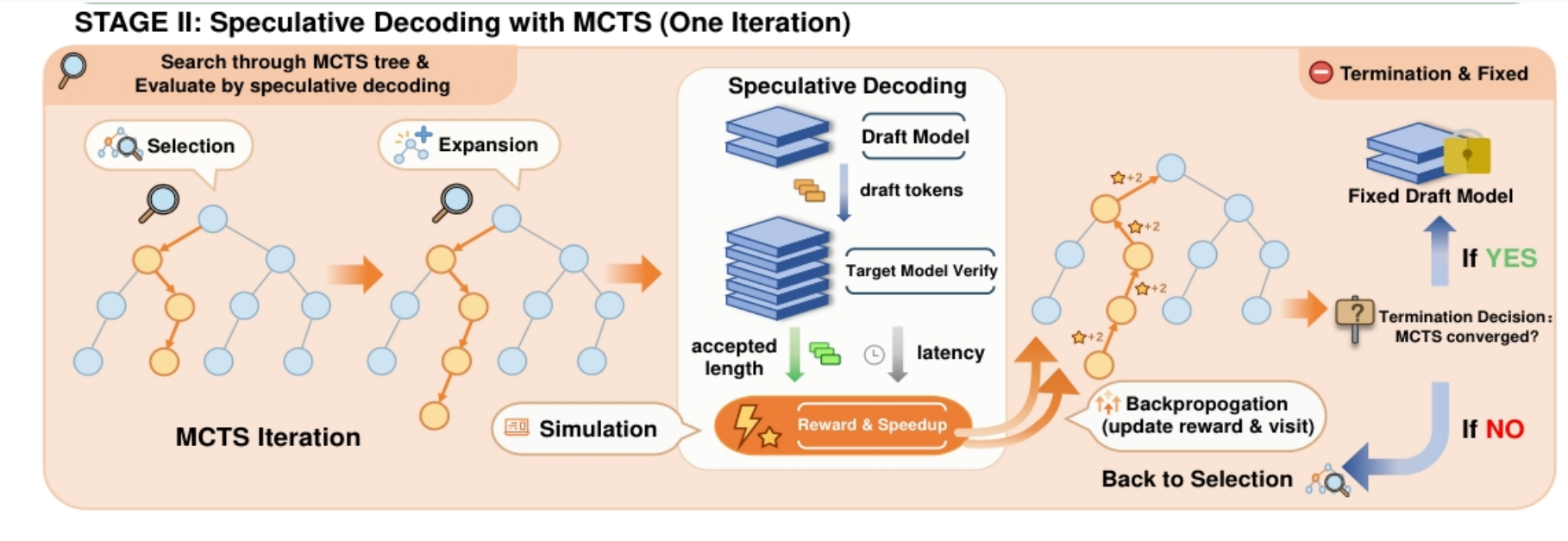
Purpose of initialization

- Structured and reduced search space for MCTS.
- Guides MCTS toward zone-specific actions for each group.
- Final actions are still selected online with SD feedback.

Stage II: MCTS-Guided Decoding

Search complete layer configurations using real-time speedup as reward.

South China University of Technology



One MCTS iteration

1. **Selection:** choose the best leaf node by UCB score.
2. **Expansion:** expand the next group with a zone-specific action.
3. **Simulation:** complete the layer configuration by rollout and run one SD iteration.
4. **Backpropagation:** update reward and visit counts along the selected path.
5. **Termination:** fix the best configuration if search converges or budget ends.

Zone-specific action constraints

Zone	Actions	Purpose
Early	execute	preserve feature transformation
Middle	execute / skip	remove redundancy
Final	execute / repeat	compensate for aggresive skipping

Key point: MCTS does not optimize a proxy metric. It directly uses the speedup feedback of speculative decoding to adjust the search direction.

Experimental Results

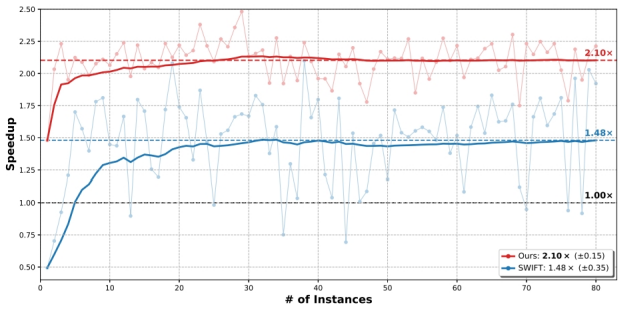
South China University of Technology

Table 1. Comparison between LEAP and other plug-and-play self-speculative decoding methods. We report the mean generation length M and wall-time speedup ratio on different datasets and models when Temperature=0 and Temperature=1.

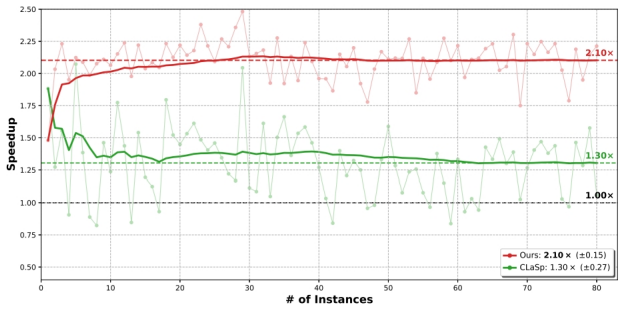
Models	Methods	GSM8K		MT-Bench		CNN/DM		NQ		WMT14		Overall Speedup
		M	Speedup	M	Speedup	M	Speedup	M	Speedup	M	Speedup	
Temperature = 0												
LLaMA-3-8B	Vanilla	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00×
	SWIFT	2.56	0.86×	5.70	1.48×	5.40	1.34×	2.61	1.00×	2.53	0.96×	1.13×
	CLaSP	1.86	1.20×	2.10	1.30×	1.91	1.22×	2.04	1.25×	2.25	1.36×	1.27×
	LEAP	4.66	**2.12**×	4.63	**2.07**×	4.44	**1.87**×	4.71	**2.09**×	4.71	**2.11**×	**2.05**×
LLaMA-3-70B	Vanilla	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00×
	SWIFT	2.24	1.40×	2.74	1.55×	2.14	1.31×	2.20	1.43×	1.92	1.35×	1.41×
	CLaSP	3.06	1.54×	3.17	1.56×	3.15	1.39×	3.08	1.49×	3.28	1.62×	1.52×
	LEAP	4.95	**2.00**×	4.81	**2.00**×	4.93	**1.81**×	4.52	**1.86**×	4.84	**2.03**×	**1.94**×
Temperature = 1												
LLaMA-3-8B	Vanilla	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00×
	SWIFT	2.34	0.78×	2.04	0.79×	1.70	0.64×	1.88	0.83×	1.91	0.78×	0.76×
	CLaSP	1.71	1.06×	1.80	1.02×	1.83	1.09×	1.80	1.03×	1.73	1.09×	1.06×
	LEAP	4.41	**1.86**×	3.97	**1.58**×	4.00	**1.69**×	3.99	**1.60**×	3.85	**1.66**×	**1.68**×
LLaMA-3-70B	Vanilla	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00	1.00×	1.00×
	SWIFT	2.48	1.08×	2.52	1.04×	2.62	1.02×	2.84	1.26×	2.58	1.10×	1.10×
	CLaSP	2.79	1.26×	2.66	1.19×	2.88	1.16×	2.52	1.09×	2.56	1.11×	1.16×
	LEAP	4.58	**1.82**×	4.41	**1.81**×	4.66	**1.65**×	3.89	**1.59**×	4.35	**1.80**×	**1.73**×

Key Analyses

- **Speedup:** LEAP achieves about 1.7×~2.0× overall speedup, consistently outperforming SWIFT and CLaSP.
- **Efficiency:** Sample-specific optimization keeps high per-instance speedup from the beginning.
- **Ablation:** Zone partitioning & layer grouping provide a structured search space; MCTS uses SD feedback to efficiently explore layer configurations



(a) Comparison with SWIFT on MT-Bench.



(b) Comparison with CLaSp on MT-Bench.

Table 2. Ablation study on different components in LEAP.

Methods	GSM8K	MT-Bench	WMT14	Overall Speedup
LEAP	2.12×	2.07×	2.11×	2.10×
w/o zone	1.78×	1.64×	1.61×	1.67×
w/o group	1.89×	1.48×	1.53×	1.63×
w/o MCTS	1.62×	1.55×	1.51×	1.56×