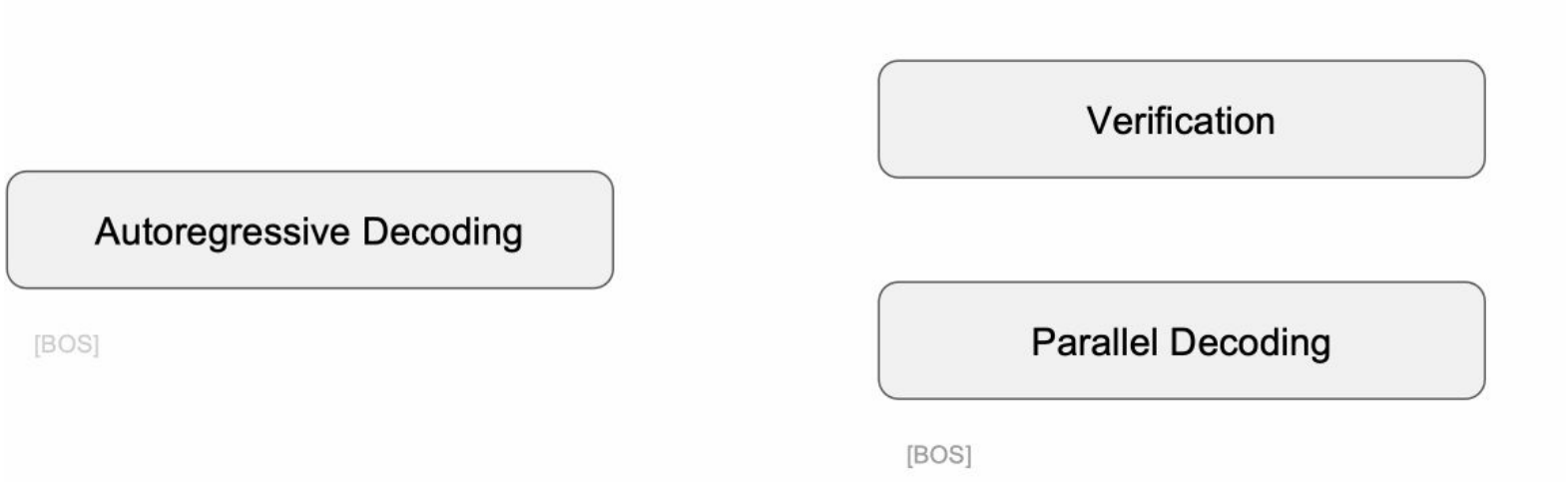
Speculative Decoding: Performance or Illusion?

Xiaoxuan Liu*, Jiaxiang Yu*, Jongseok Park, Ion Stoica, Alvin Cheung

MLSys 2026

Speculative Decoding (SD)

- Speculative decoding reduces latency by amortizing memory access



SD is Widely Deployed in Production LLM Serving Stacks

Model

Framework

Application

Code editing

Online Chatting

Reasoning

.....

What is the Performance of SD in Production?

To answer this question, we need a rigorous benchmark.

Prior work often assumes:

- Many evaluations use prototype implementations.
- Batch size is often 1.
- Few comparisons across SD variants.
- Limited analysis of why speedup appears or disappears.

We need a systematic study under production-like serving conditions.

Today’s Schedule

- End to end performance
  - Production-level system
  - Representative workloads
- Understand the performance
  - Runtime breakdown
  - Acceptance behaviour
- How far are we from optimal performance?

End-to-End Performance: Experiment Setup

Engine & Hardware: vLLM 0.10.1 / 0.11.1 on Nvidia H100

SD Methods: n-gram, EAGLE, EAGLE-3, draft model, MTP

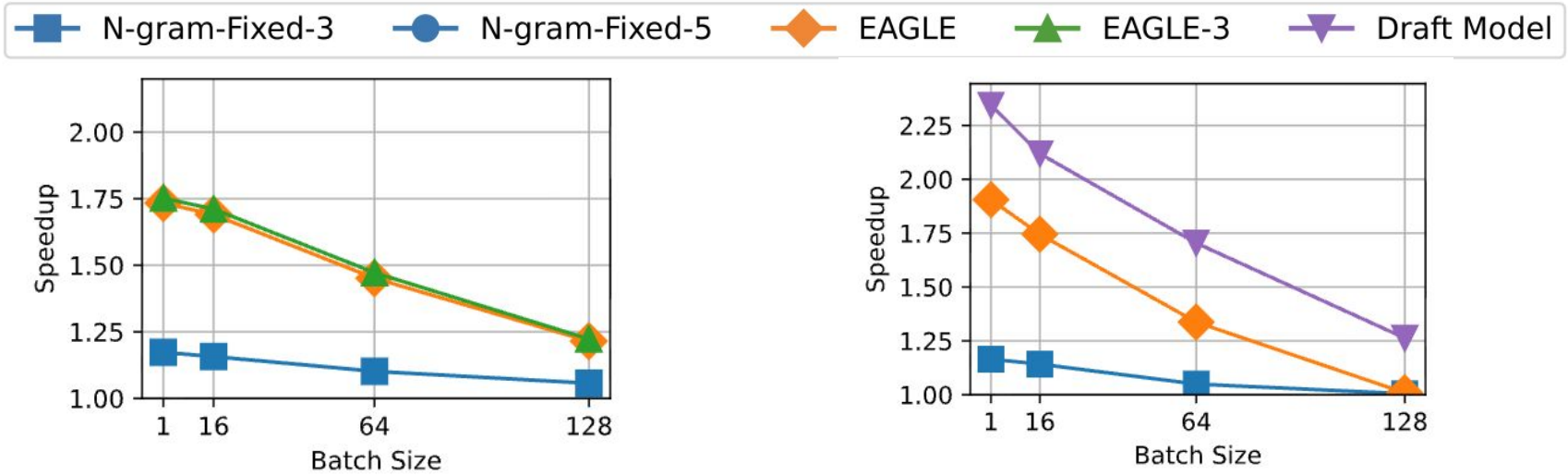
Models: Llama3.1-8B, Llama3-70B, Qwen3-8B, GLM-4.5-Air-106B

Workloads:

- Code editing (InstructCoder), Online Chatting (ShareGPT)
- Summarization (CNN/DailyMail), Math (GSM8K)
- Long-Form Reasoning (AIME, GPQA)

Metric: Generation throughput

End-to-End Performance: Batch Size

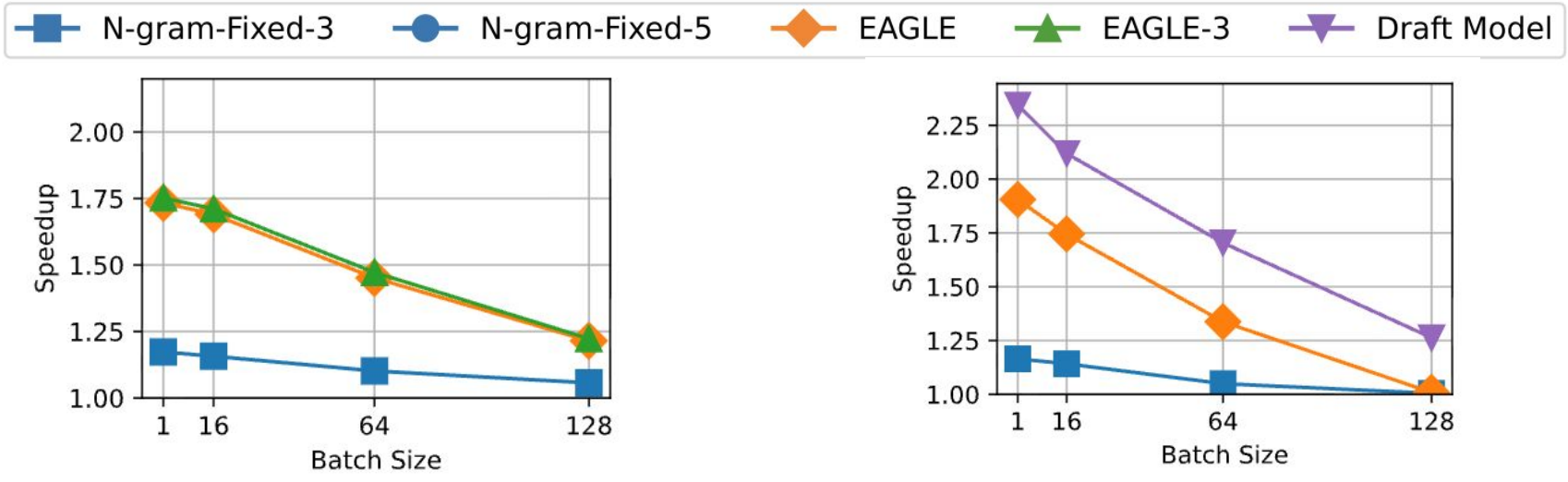


Llama3.1-8B, GSM8K

Llama3-70B, GSM8K

Larger batches increase throughput but reduce speculative decoding speedups, especially for larger models.

End-to-End Performance: SD Variants and Datasets

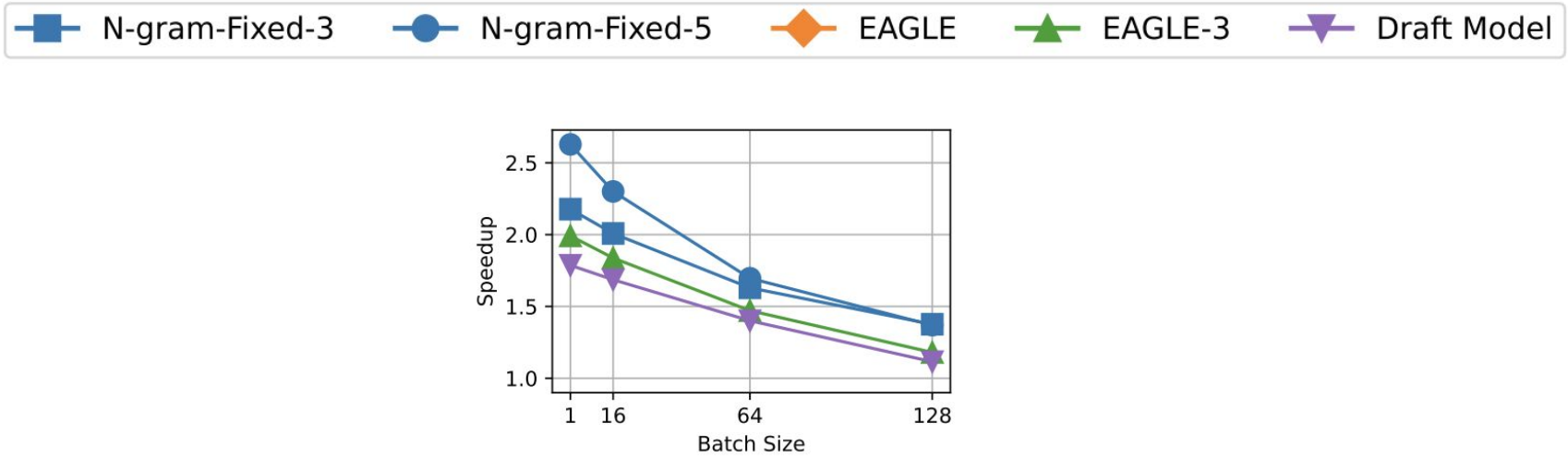


Llama3.1-8B, GSM8K

Llama3-70B, GSM8K

SD effectiveness depends on the method and workload: draft-model methods excel on 70B target model but weaken on smaller models.

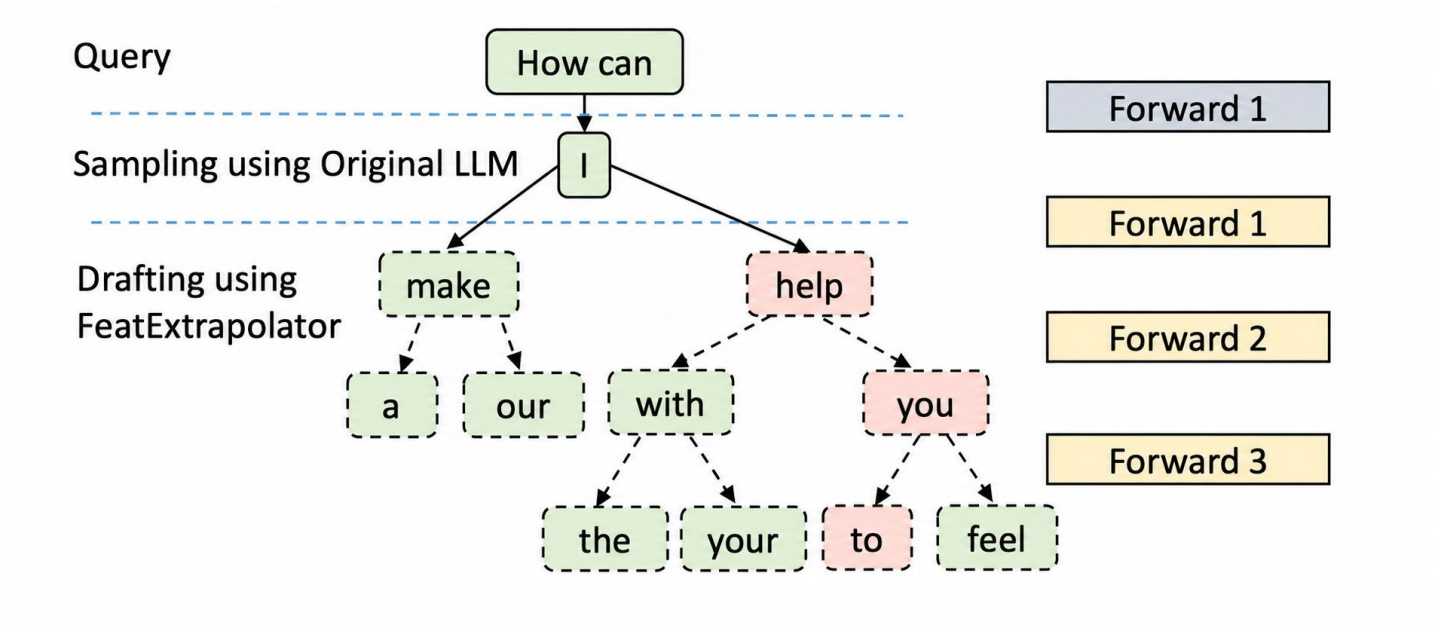
End-to-End Performance: SD Variants and Datasets



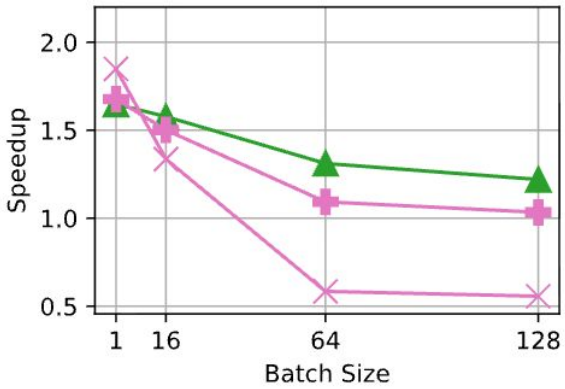
Qwen3-8B, InstructCoder

SD effectiveness depends on the method and workload:
n-gram works the best for code editing

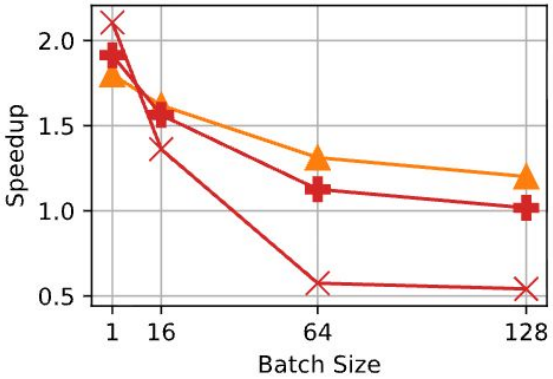
End-to-End Performance: Tree-style Verification



End-to-End Performance: Tree-style Verification



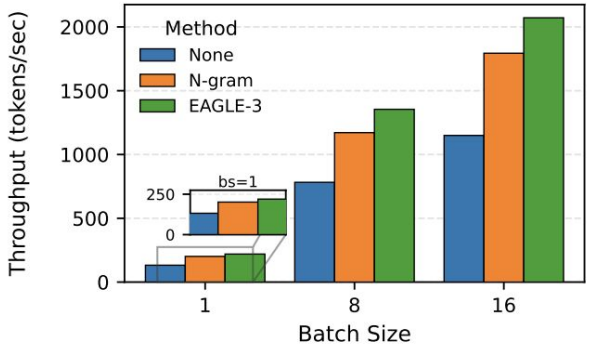
Qwen3-8B, GSM8K



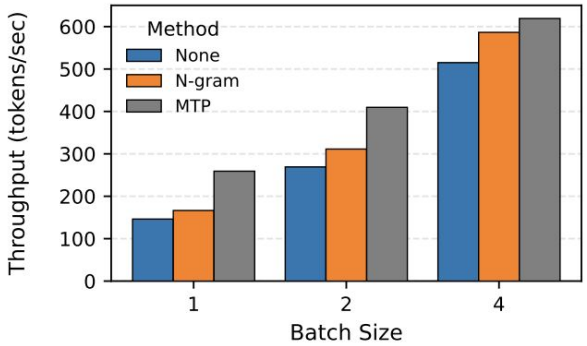
Llama3-70B, GSM8K

Tree-based methods lead slightly at batch size 1, but chain-style methods are often stronger as batch size grows.

End-to-End Performance: MTP and Reasoning Workloads



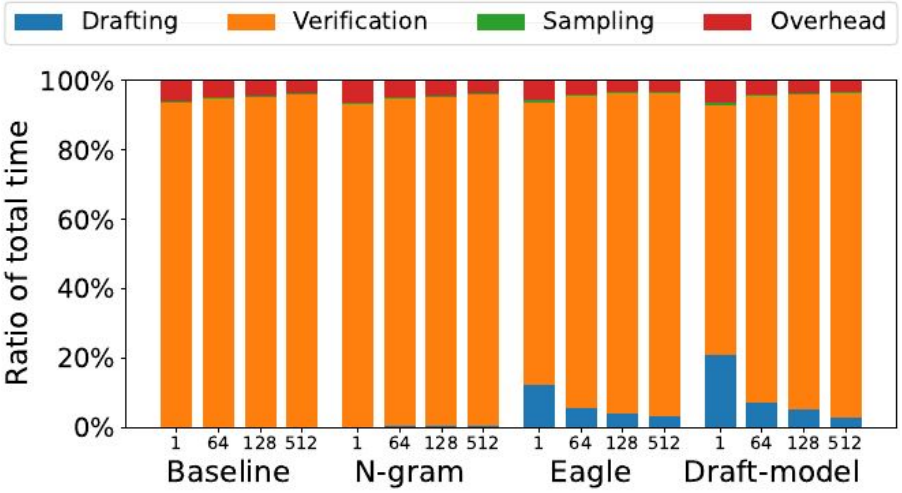
Qwen3-8B-Thinking, AIME22-24



GLM-4.5-Air, AIME22-24

Reasoning tasks benefit most from methods that maintain high long-context acceptance, while MTP is constrained by reusing a single head across drafted tokens.

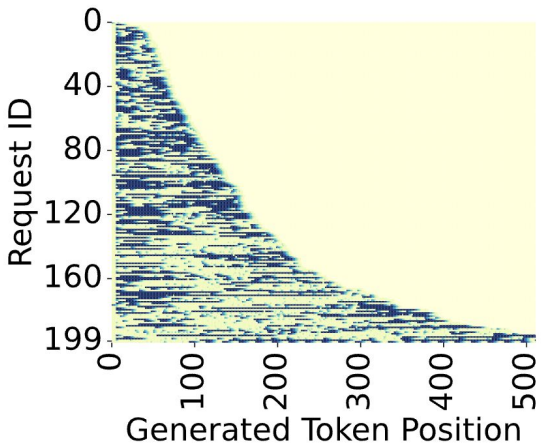
Understand the Performance: Runtime Breakdown



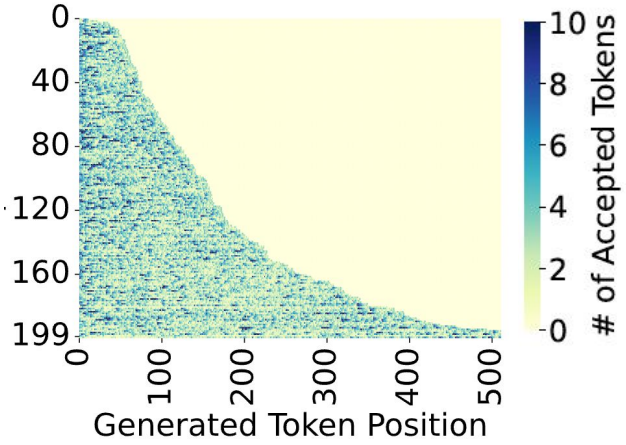
Llama3-70B, CNN/DailyMail

- **Verification time dominates**
- **Drafting time** is
  - **Small** for n-gram;
  - **Up to 20%** for EAGLE/EAGLE3;
  - **Up to 47%** for draft-model
- **Sampling time and other overheads** are relatively **small**

Understand the Performance: Acceptance Behaviour



N-gram, Llama3.1-8B, InstructCoder

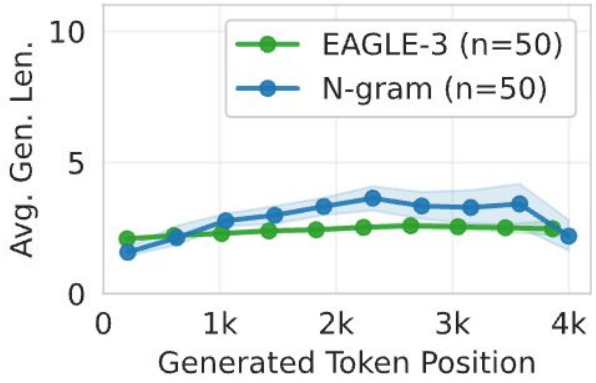


EAGLE, Llama3.1-8B, InstructCoder

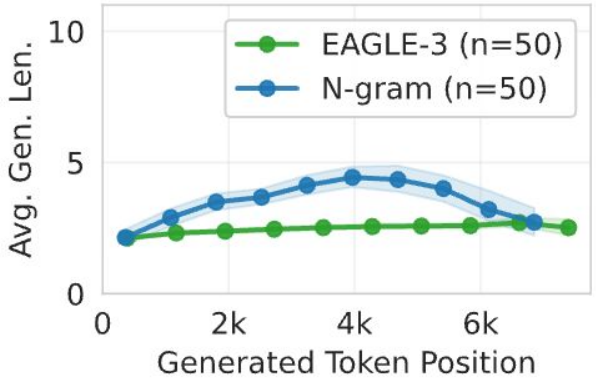
Acceptance varies **within a request** and **across requests.**

(as well as across datasets, more details in our paper)

Understand the Performance: Acceptance Behaviour



Output < 4k tokens

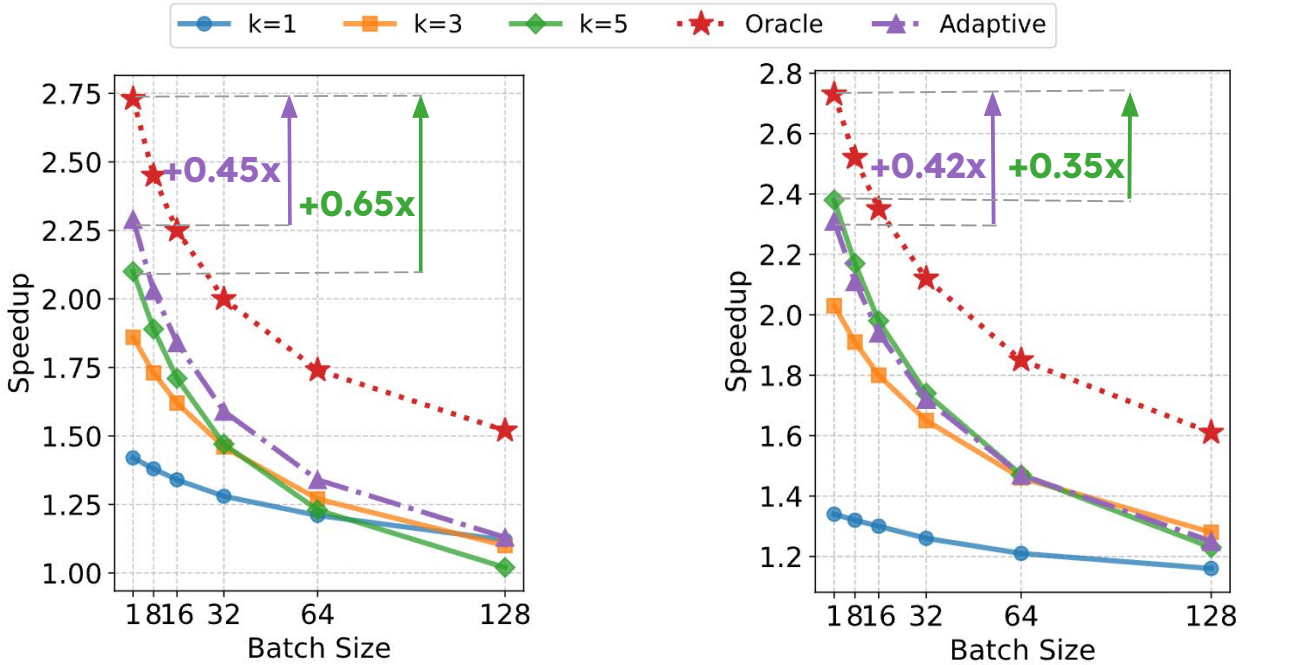


Output of 4k-8k tokens

Qwen3-8B-Thinking, GPQA-Main

N-gram **benefits from repetitive patterns during reasoning**, but **drops near the end** as the generation shifts toward conclusions.

How Far Are We From the Upper Bound?

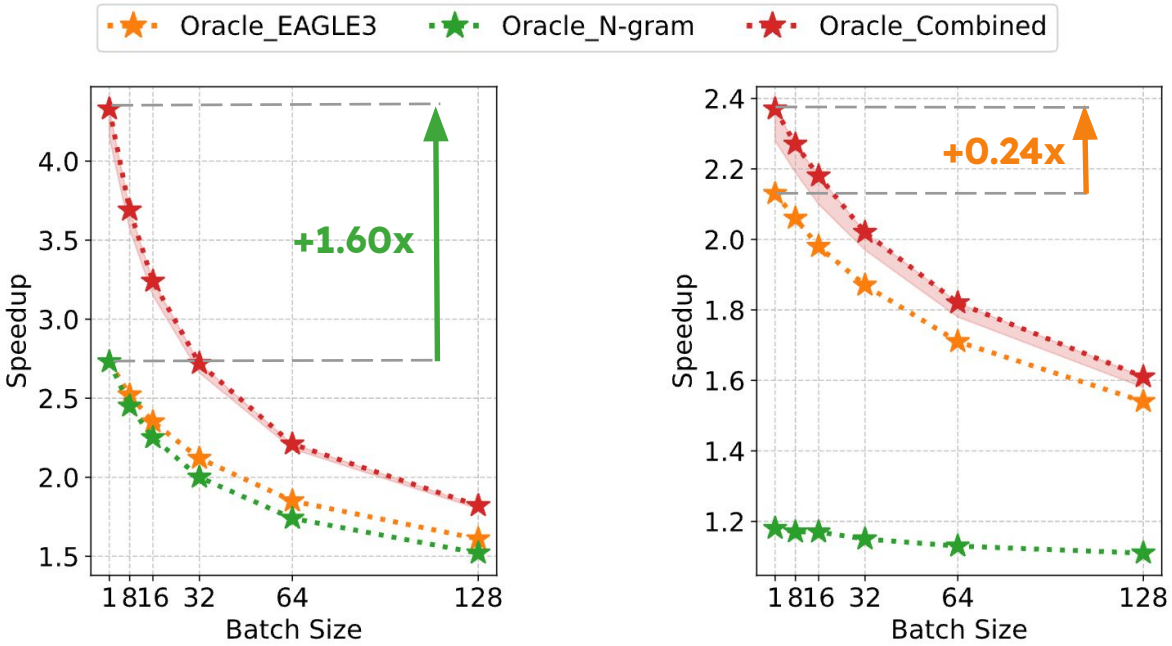


N-gram, Llama3.1-8B, InstructCoder

EAGLE-3, Llama3.1-8B, InstructCoder

Fixed-k and simple adaptive policy leave significant room for improvement, and **the gap widens at higher batch sizes.**

Combining Complementary SD Methods



Llama3.1-8B, InstructCoder

Llama3.1-8B, GSM8K

Combining complementary SD methods reveals substantial extra headroom, but **is highly workload-dependent.**

Takeaways

- Speculative decoding **provides real speedups**
- **Verification** is the bottleneck in runtime
- Acceptance varies **across token positions, requests, and datasets**
- **Adaptive proposal length and proposer selection** are promising future directions

Check out our paper, code and blog

Thank you!

Berkeley UNIVERSITY OF CALIFORNIA

sky UC Berkeley