Fast Inference from Transformers via Speculative Decoding

Leviathan, Yaniv et al. “Fast Inference from Transformers via Speculative Decoding.” International Conference on Machine Learning (2022)

Presenter: Ao Li, Bogdan Pikula

UNIVERSITY OF TORONTO

Agenda

- Background
- Motivation & Design
- Analysis
- Experiments
- Discussion

Background

- With great parameters comes long inference time

- Decoding happens sequentially
  - Decode token k requires token k - 1
  - Can we improve the latency?

UNIVERSITY OF TORONTO

Related Work

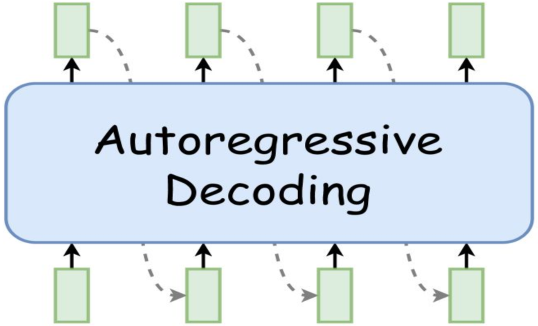
- Make the model smaller
  - Distiliation, Transfer Learning
  - Requires a re-training
- Modify the design of model
  - Adaptive Transformers

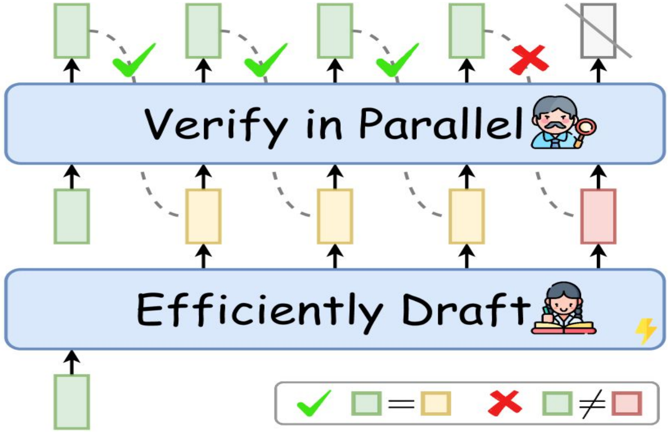
Observations

- Observation 1: Inference bottlenecked on memory bandwidth and communcation
  - Majority of time is not spent on computation but on moving parameters and weights
- Observation 2: Some inference are easy to do even with a small model.
  - e.g., My name ?

Speculative Decoding

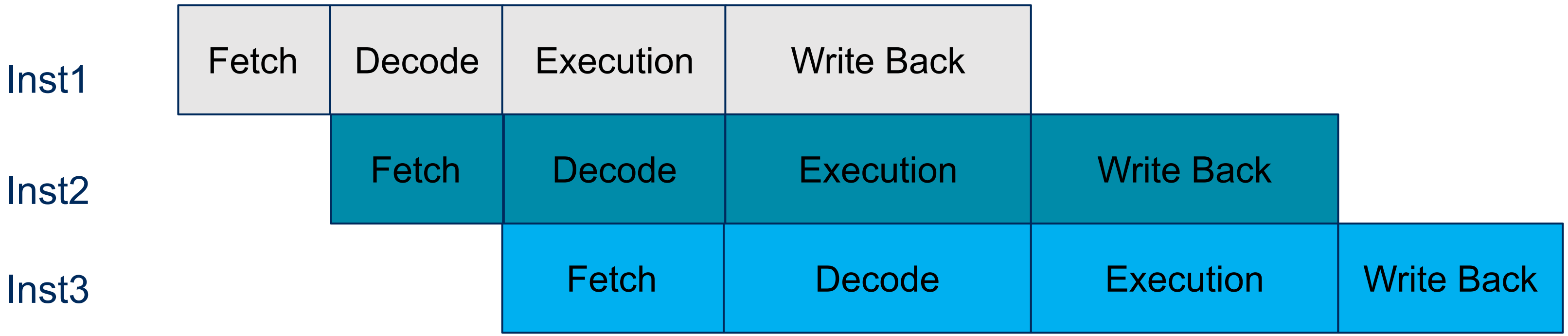
- Utilize smaller models as draft models to generate tokens in less amount of time
- Target model is in charged of verifcation of drafts in parallel





Speculative Execution - Instruction Pipeline

- There are 4 stages for an instruction:
    - Fetch, Decode, Execution, and Write Back
- Each instruction can enter the pipeline once previous instruction shifts to a different stage



Pipeline - Branching

- Branching can slow down the performance
- if (cond) then A
- The instruction has to wait the result of cond to be available in the hardware to execute
    - This causes the stall

Fetch	Decode	Execution	Write Back		
	Fetch	Decode	Stall	Execution	Write Back

Speculative Execution - Branch Prediction

- Instead of stall, take a guess of which branch will be taken and execute it.
- If prediction is correct: saves time
- If wrong: discard results (minimal overhead)
- Hide latency in pipelined processors by keeping execution units busy

Speculative Decoding - Draft Stage

1. Draft model samples γ tokens and probabilities

▷ Sample γ guesses $x_{1,\ldots,\gamma}$ from M_q autoregressively.

for $i = 1$ **to** γ **do**

$\quad q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$

$\quad x_i \sim q_i(x)$

end for

Speculative Decoding - Verfication

2. Target Model validates the guesses and their respective probabilities in parallel

▷ Run M_p in parallel.

$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$

$\quad M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$

▷ Determine the number of accepted guesses n.

$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$

$n \leftarrow \min(\{i-1 \mid 1 \le i \le \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$

▷ Adjust the distribution from M_p if needed.

$p'(x) \leftarrow p_{n+1}(x)$

if $n < \gamma$ **then**

$\quad p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$

end if

Speculative Decoding - Verification

- The verification is measured by the probability between target model and draft model
- Accept if $P_{target}(x) \geq P_{draft}(x)$
  - Target model is more confident in this case
- Otherwise, accept with probability $\frac{P_{target}(x)}{P_{draft(x)}}$

Token	x1	x2	x3	x4	x5
	dogs	love	chasing	after	cars
Draft	0.8	0.7	0.9	0.8	0.7
Target	0.9	0.8	0.8	0.3	0.8
Accept	✅	✅	✅	❌	❌

UNIVERSITY OF TORONTO

Speculative Decoding - Additional Sampling

3. Sample an additional token to fix the first one that was rejected, or to add an additional one if they are all accepted

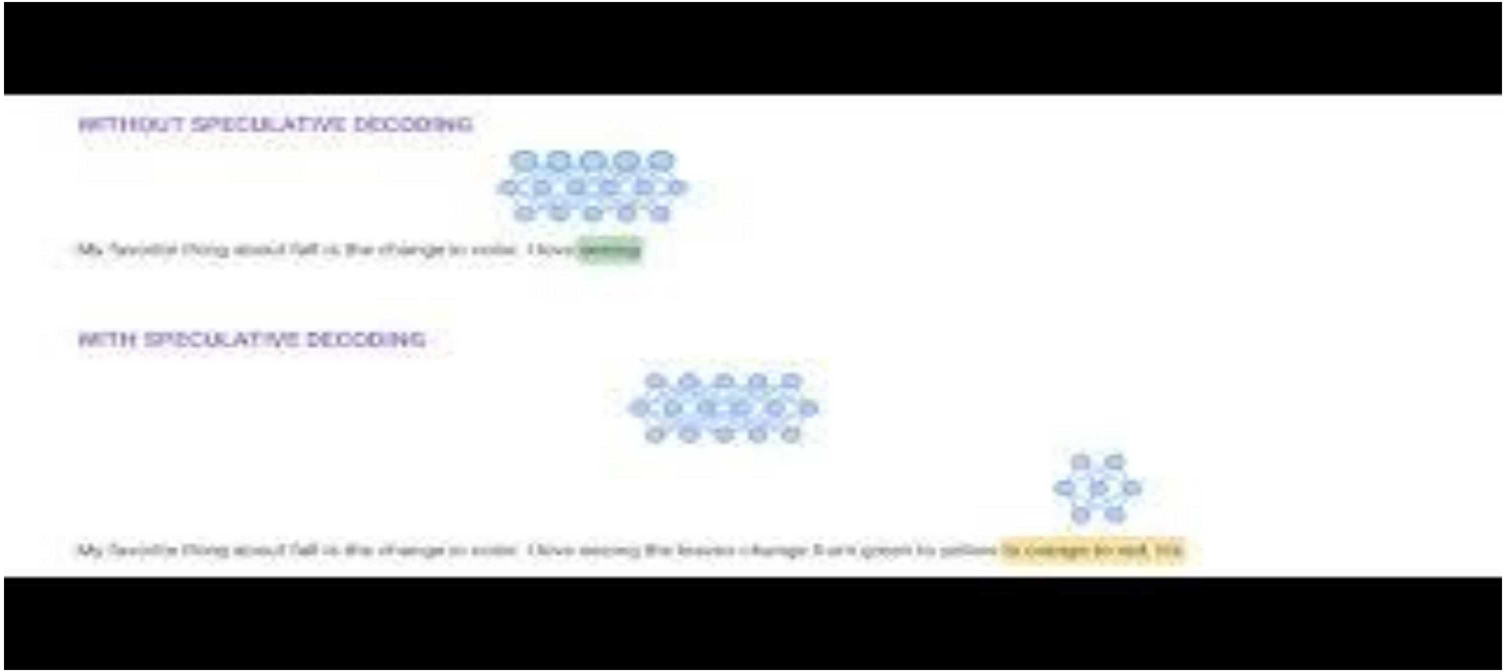
$\triangleright$ Return one token from M_p, and n tokens from M_q.

$t \sim p'(x)$

return $prefix + [x_1, \ldots, x_n, t]$

UNIVERSITY OF TORONTO

Animation



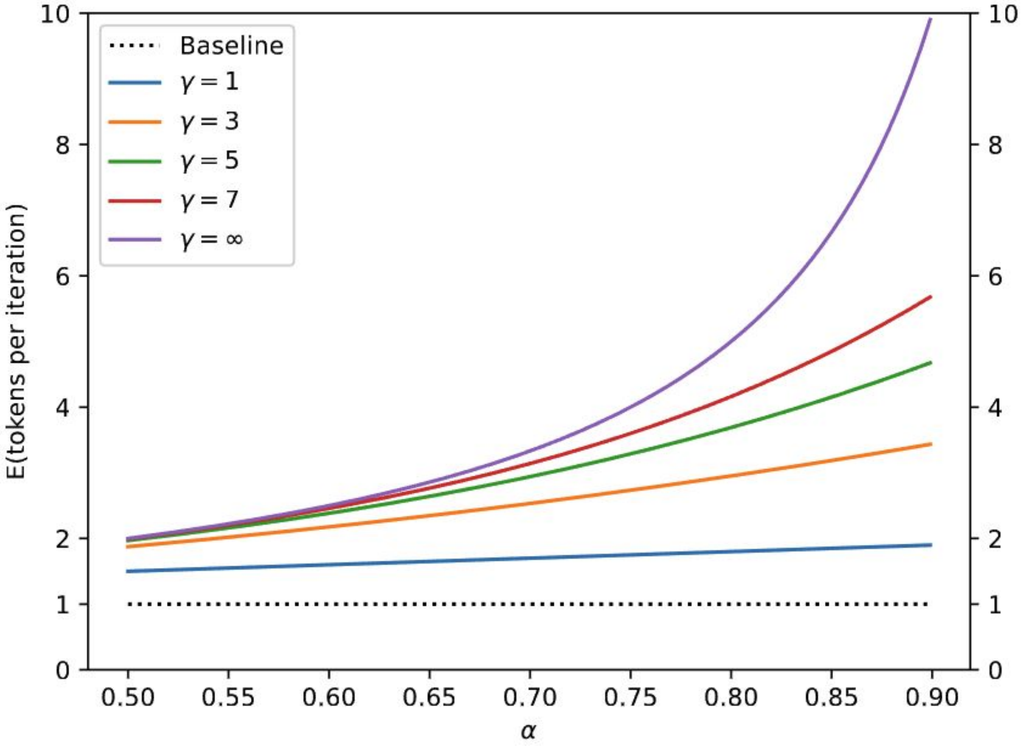
[https://research.google/blog/looking-back-at-speculative-decoding/]

Analysis - Walltime Improvement

- Key Metric: Acceptance Rate (***α***)
  - Probability of small model's tokens being accepted
- Expected tokens per iteration: $\dfrac{1-\alpha^{\gamma+1}}{1-\alpha}$
  - ***γ*** = speculation length
- Walltime Improvement factor: $\dfrac{1-\alpha^{\gamma+1}}{(1-\alpha)(\gamma c+1)}$
  - c = cost coefficient (runtime ratio of small vs. large model)
- Trade-off: More speculative tokens (***γ***) → more potential speedup but diminishing returns

Analysis - Expected Tokens Per Iteration

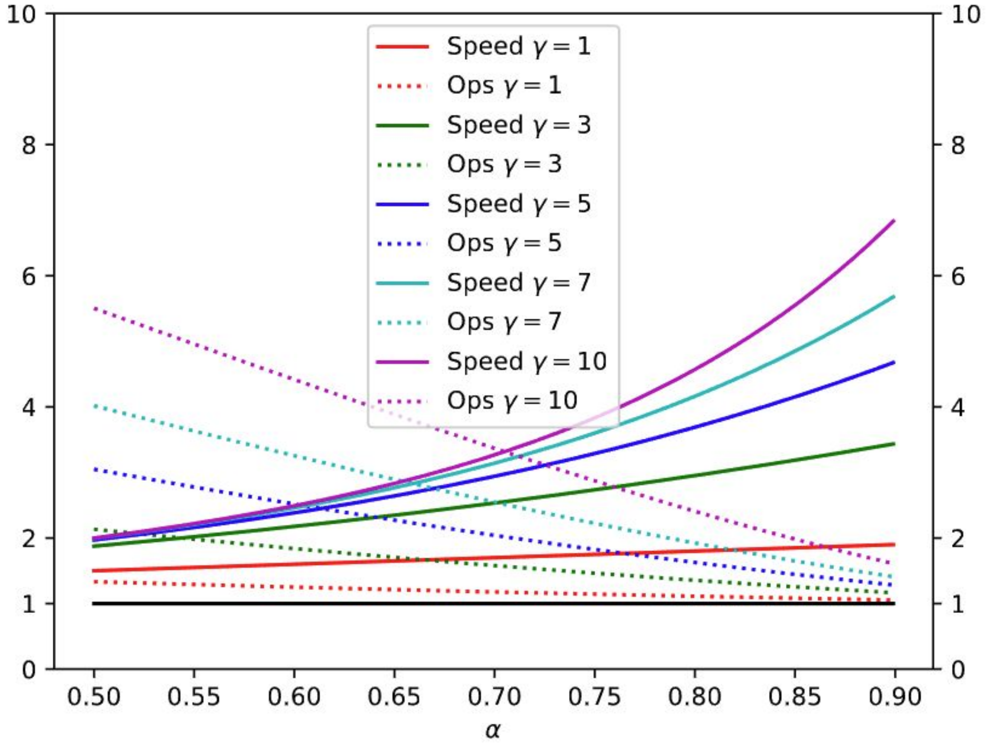
- ***Higher acceptance rates dramatically improve throughput;***
- X-axis: Acceptance rate (***α***) ϵ [0.5, 0.9];
- Y-axis: expected tokens per iteration (1 to 10);
- Multiple curves for ***γ*** = {1,3,5,7, ∞};
- Baseline (standard decoding) = 1 token per iteration.



Expected number of tokens generated by SpeculativeDecodingStep as a function of ***α*** for various values of ***γ***

Analysis - Speed vs. Computation

- ***Speed improves while computation increases, but the trade-off is favorable;***
- X-axis: Acceptance rate (***α***) ϵ [0.5, 0.9];
- Y-axis: Factor relative to baseline (1 to 10)
- Multiple curves for different ***γ*** values {3, 5, 7, 10}



Experimental Setup

- Target Models:
  - T5-XXL (11B param)
  - GPT-like model (97B param)
  - LaMDA (137B param)

- Approximation Models:
  - T5-small (77M param), T5-base (250M param), T5-large (800M param)
  - Smaller GPT-like models (6M parameters)
  - Even simple n-gram models

- Tasks:
  - English-German translation (WMT)
  - News summarization (CNN/DM)
  - Unconditional Generation (LM1B)
  - Dialog Tasks

Empirical Results - Translation & Summarization

Real-world speedups of 2-3X with identical outputs

TASK	M_q	TEMP	γ	α	SPEED
ENDE	T5-SMALL ★	0	7	0.75	**3.4X**
ENDE	T5-BASE	0	7	0.8	2.8X
ENDE	T5-LARGE	0	7	0.82	1.7X
ENDE	T5-SMALL ★	1	7	0.62	**2.6X**
ENDE	T5-BASE	1	5	0.68	2.4X
ENDE	T5-LARGE	1	3	0.71	1.4X
CNNDM	T5-SMALL ★	0	5	0.65	**3.1X**
CNNDM	T5-BASE	0	5	0.73	3.0X
CNNDM	T5-LARGE	0	3	0.74	2.2X
CNNDM	T5-SMALL ★	1	5	0.53	**2.3X**
CNNDM	T5-BASE	1	3	0.55	2.2X
CNNDM	T5-LARGE	1	3	0.56	1.7X

Empirical results for speeding up inference from a T5XXL 11B model.

Acceptance Rates (α) Across Models

Size increases

Model Pair (Mq → Mp)	Task	SMPL{Temp=0} (α)	SMPL{Temp=1} (α)
Unigram → T5-XXL	Translation(En-De)	*0.08* ⇧	*0.07* ⇩
Bigram → T5-XXL	Translation(En-De)	*0.20* ⇧	*0.19* ⇩
T5-small → T5-XXL	Translation(En-De)	*0.75* ⇧	*0.62* ⇩
T5-base → T5-XXL	Translation(En-De)	*0.80* ⇧	*0.68* ⇩
T5-large → T5-XXL	Translation(En-De)	*0.82* ⇧	*0.71* ⇩

- Higher α with argmax sampling (T=0) vs. standard sampling (T=1)
- Even simple models achieve non-zero acceptance rates
- Larger approximation models → higher acceptance rates

Theoretical vs. Empirical Results

The mathematical model generally predicts real-world performance well

Task	M_q	Temp	γ	α	c	Exp		Emp
EnDe	T5-small	0	7	0.75	0.02	3.2	⇔	3.4
EnDe	T5-base	0	7	0.8	0.04	3.3	⇔	2.8
EnDe	T5-large	0	7	0.82	0.11	2.5	⇔	1.7
EnDe	T5-small	1	7	0.62	0.02	2.3	⇔	2.6
EnDe	T5-base	1	5	0.68	0.04	2.4	⇔	2.4
EnDe	T5-large	1	3	0.71	0.11	2.0	⇔	1.4
CNNDM	T5-small	0	5	0.65	0.02	2.4	⇔	3.1
CNNDM	T5-base	0	5	0.73	0.04	2.6	⇔	3.0
CNNDM	T5-large	0	3	0.74	0.11	2.0	⇔	2.2
CNNDM	T5-small	1	5	0.53	0.02	1.9	⇔	2.3
CNNDM	T5-base	1	3	0.55	0.04	1.8	⇔	2.2
CNNDM	T5-large	1	3	0.56	0.11	1.6	⇔	1.7

Expected improvement factor (EXP) vs. empirically measured improvement factor (EMP).

Limitations and Privacy Concerns

- Privacy Concern
  - Spectre and Meltdown
  - Input fingerprinting: Network-based attackers can identify user queries with >90% accuracy
  - Wei, J., Abdulrazzag, A., Zhang, T., Muursepp, A., & Saileshwar, G. (2024). Privacy Risks of Speculative Decoding in Large Language Models. ArXiv. https://arxiv.org/abs/2411.01076
- Increased computational overhead

Key Takeaways from Experiments

- Practical speedups of **2-3X** with no change to model outputs;
- Works across model scales - from millions to billions of parameters;
- Effective for diverse tasks - translation, summarization, dialog, etc;
- Small approximation models often provide best overall speedup;
- Even simple models (like n-grams) can provide measurable benefits;

Discussion

Questions?