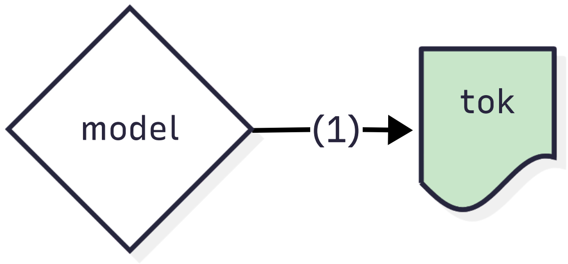
Accelerating LLM Inference with Lossless Speculative Decoding Algorithms for Heterogeneous Vocabularies

Nadav Timor [w], Jonathan Mamou[i], Daniel Korat[i], Moshe Berchansky[i], Gaurav Jain[d], Oren Pereg[i], Moshe Wasserblat[i], David Harel[w]

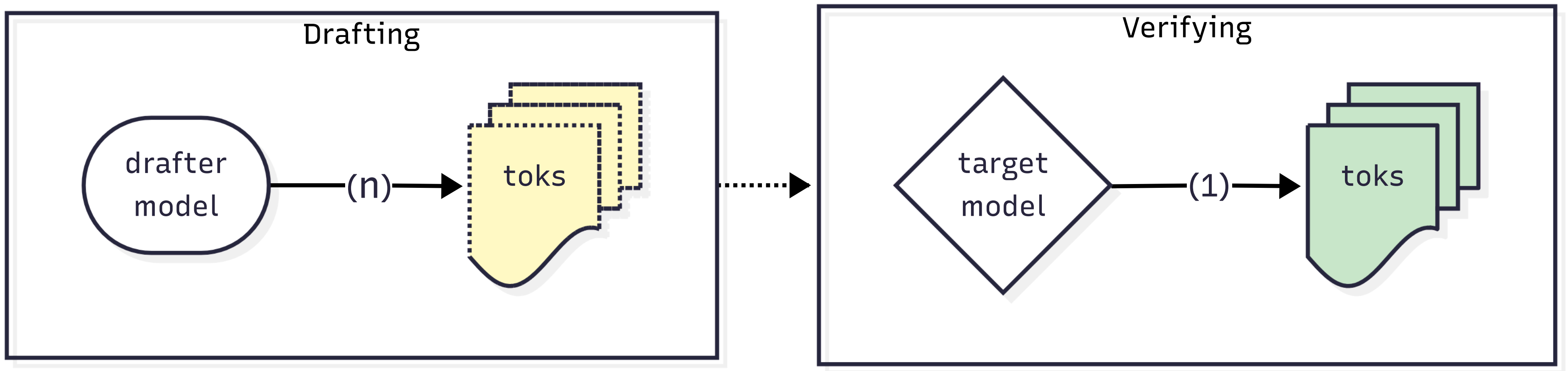
[w] Weizmann Institute of Science, [i] Intel Labs, [d] d-Matrix

Speculative decoding [Leviathan et al., 2023; Chen et al.,

- Autoregressive decoding



- Speculative decoding



    - up to 3x faster ($\downarrow$ latency, $\uparrow$ throughput)
    - lossless

Contribution: Removing the shared-vocab constraint

Current limitation: drafter must share the same vocab as the target

- 💸 Training drafters from scratch:
  - No family (e.g., `DeepSeek-R1`, `phi-4`, `Mixtral-8x22B`, `CodeLlama`)
  - In-family is too slow (e.g., `DeepSeek-R1-Distill-*`, `Llama 3.1`, `gemma-2`)
  - No reuse

Our contribution: removing this limitation (& remaining lossless)

- 🆓 No training:
  - Any off-the-shelf drafter
  - Reuse
- Up to 2.8x faster (than autoregressive)
- Default in 🤗 Transformers (since Oct '24 + Feb '25)

Usage example

```diff
from transformers import pipeline

pipe = pipeline(
    "text-generation",
    model="google/gemma-2-9b-it",
-   assistant_model="google/gemma-2-2b-it"
+   assistant_model="double7/vicuna-68m"        # 1.5x lossless speedup!
)
out = pipe("Summarize this article...")
```

Our 3 algos

- Speculative decoding is undefined for heterogeneous vocabs

1. TLI, Token-level intersection 🤗
    - vocab pruning
2. SLEM, String-level exact match 🤗
    - back-and-forth tokenization + heuristic
3. SLRS, String-level rejection sampling
    - probs on strings

- How to choose?

Theoretical guarantees

- Lossless
- Acceptance rate (expected)
- Acceptance rate is higher than baseline

Empirical speedups

- **Up to 2.8$\times$ toks/sec**
- Various hardware
- Tasks:
  - summarization
  - coding
  - long-context understanding
- Independent evaluation by Hugging Face

Summary: free-lunch for everyone

- Speculative decoding with any off-the-shelf drafter
- Unlocks **lossless** speedups that previously required training
- Default in 🤗 (388k repos + 6k libs)

Qs?

Poster session, **4:30-7:00 pm** (📍East Exhibition Hall A-B #E-2810)

> **Summary:** free-lunch for everyone
>
> 1. Speculative decoding with any off-the-shelf drafter
> 2. Unlocks lossless speedups that previously required training
> 3. Default in 🤗 (388k repos + 6k libs)

Thank you!

Poster session, **4:30-7:00 pm** (📍East Exhibition Hall A-B #E-2810)

> **Summary:** free-lunch for everyone
>
> 1. Speculative decoding with any off-the-shelf drafter
> 2. Unlocks lossless speedups that previously required training
> 3. Default in 🤗 (388k repos + 6k libs)

Appendix

- Speculative decoding recap
- Heterogeneous vocabulary challenge
- SLEM
- SLRS
- TLI
- Detailed empirical results

Speculative decoding recap

- **Algo (informal):** Repeat until generated N tokens:
  - k drafter fwds
  - 1 target fwd (batching)
  - lossless rejection sampling (accept / reject & resample)
- **Intuition:**

$$1 \text{ target fwd} \quad \to \quad > 1 \text{ new tokens}$$

Heterogeneous vocabulary challenge

- Tokens outside the intersection have **no “natural” 1-to-1 mapping**

$$\begin{aligned}\text{`rainbow'} &\rightarrow \text{`rain'} \oplus \text{`bow'} \\ &\rightarrow \text{`r'} \oplus \text{`a'} \oplus \text{`i'} \oplus \text{`n'} \oplus \text{`b'} \oplus \text{`o'} \oplus \text{`w'}\end{aligned}$$

- **Observation:** natural 1-to-n exists

String-Level Exact Match (SLEM)

- **Algo (informal):**
  i. Drafter outputs k draft tokens $\rightarrow$ untokenize to string S
  ii. Tokenize S w.r.t. target $\rightarrow$ target token sequence $(t_1, \ldots, t_m)$
  iii. Target outputs $(t'_1, \ldots, t'_m, t'_{m+1})$ via batching
  iv. Accept $(t_1, \ldots, t_{j-1}, t'_j)$
  i.e., longest matched prefix + first unmatched target token

- **Primary challenge:** non-injective tokenizers

$$s \neq \text{untokenize}(\text{tokenize}(s))$$

  - Our heuristic (covers common cases):

$$c_2 \oplus s = \text{untokenize}(\text{tokenize}(c_1 \oplus c_2 \oplus s))$$

String-Level Rejection Sampling (SLRS)

- **Algo (informal):**
  i. Drafter outputs k draft tokens $\rightarrow$ untokenize to string S
  ii. Tokenize S w.r.t. target $\rightarrow$ target token sequence $(t_1, \ldots, t_m)$
  iii. SD verification between target $p(t_1)$ and generalized drafter $\psi(t_1)$:
     - Accept t_1 if $p(t_1) > \psi(t_1)$ else w.p. $p(t_1)/\psi(t_1)$
- Example:

$$\psi(\text{`hey'}) = q(\text{`hey'}) + q(\text{`he'}, \text{`y'}) + q(\text{`h'}, \text{`ey'}) + q(\text{`h'}, \text{`e'}, \text{`y'})$$

- **Thm:**
  - lossless
  - higher acceptance rate than SLEM

Token-Level Intersection (TLI)

- A sampler for the drafter
- **Algo (informal):**
  i. Zero-out probabilities outside the intersection
  ii. Normalize
- **Thm:** higher acceptance rate (than naive "union" approach)

Detailed empirical results

- **Tasks:** summarization (CNN-DailyMail), code generation (HumanEval), long-context understanding (SCROLLS) on various hardware setups
- **Speedups:**
  - up to $1.7\times$ (TLI) and $2.8\times$ (SLEM) tokens per second
  - SLEM & TLI outperform homogeneous SD (e.g., DeepSeek-R1, Gemma-2)
  - independent evaluation by Hugging Face