SpecDiff-2: Scaling Diffusion Drafter Alignment For Faster Speculative Decoding

Jameson Sandler, Jacob Christopher, Tom Hartvigsen, Ferdinando Fioretto

Contents

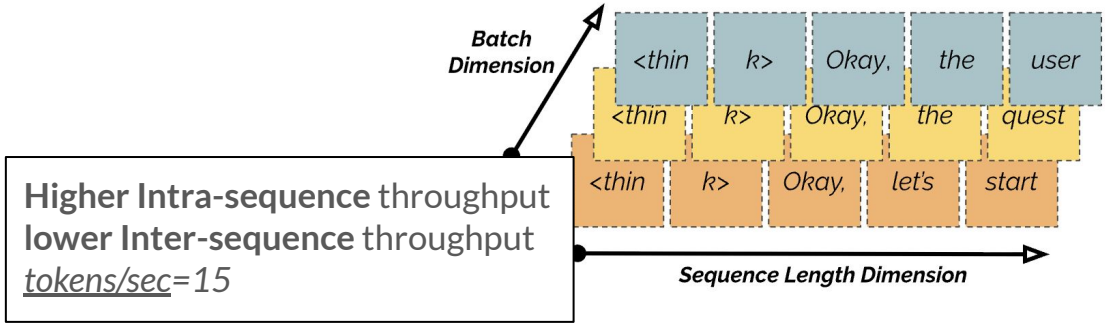
1. Convince you that Speculative-Decoding R&D is important

2. Show how '*Diffusion-Drafters*' can be effective language-generation accelerators

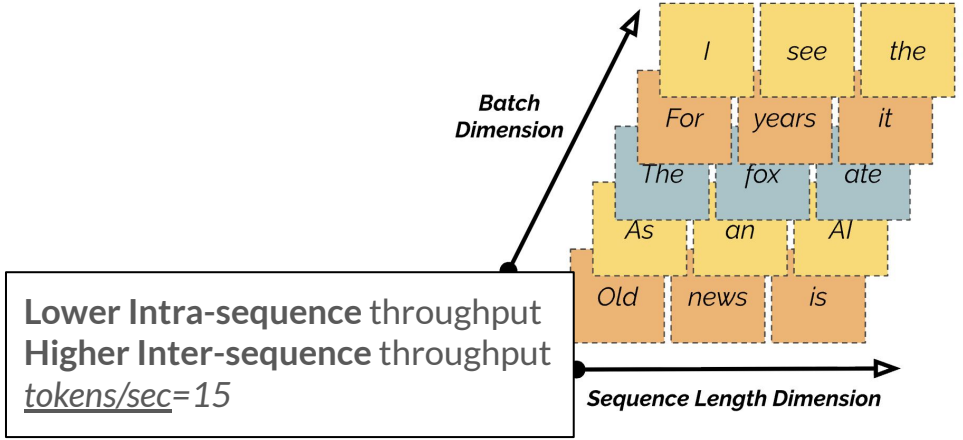
(1) What Does Speculative-Decoding Offer Us?

We can achieve *higher throughput* (*tokens/sec*) by,

1. Scaling up the batch dimension ***(inter-sequence)***
2. Scaling throughput along the sequence length dimension ***(intra-sequence)***



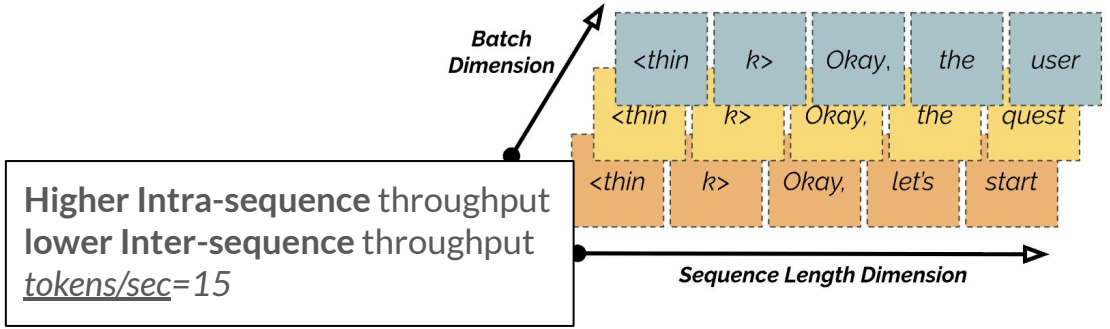
Higher Intra-sequence throughput
lower Inter-sequence throughput
tokens/sec=15



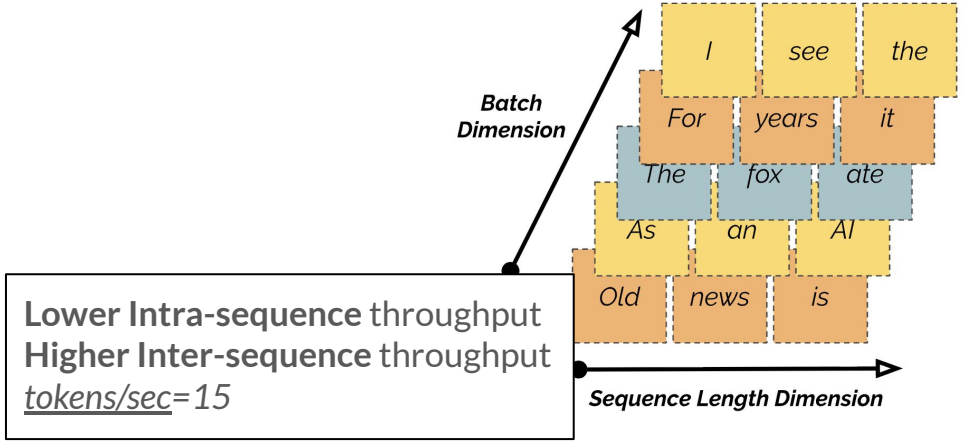
Lower Intra-sequence throughput
Higher Inter-sequence throughput
tokens/sec=15

(1) What Does Speculative-Decoding Offer Us?

SpecDec is the predominant framework for enhancing *'intra-sequence'* throughput **losslessly**.



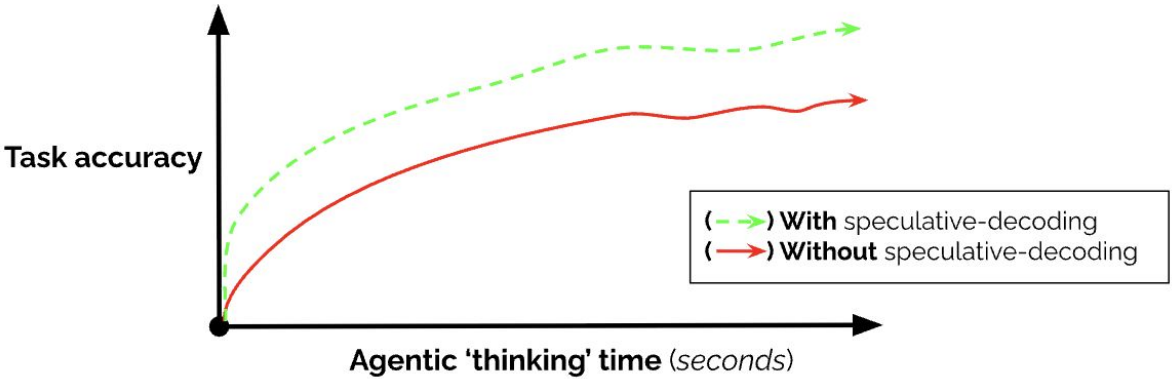
Higher Intra-sequence throughput
lower Inter-sequence throughput
tokens/sec=15



Lower Intra-sequence throughput
Higher Inter-sequence throughput
tokens/sec=15

(1) Why Speculative-Decoding? *Test-time Scaling*

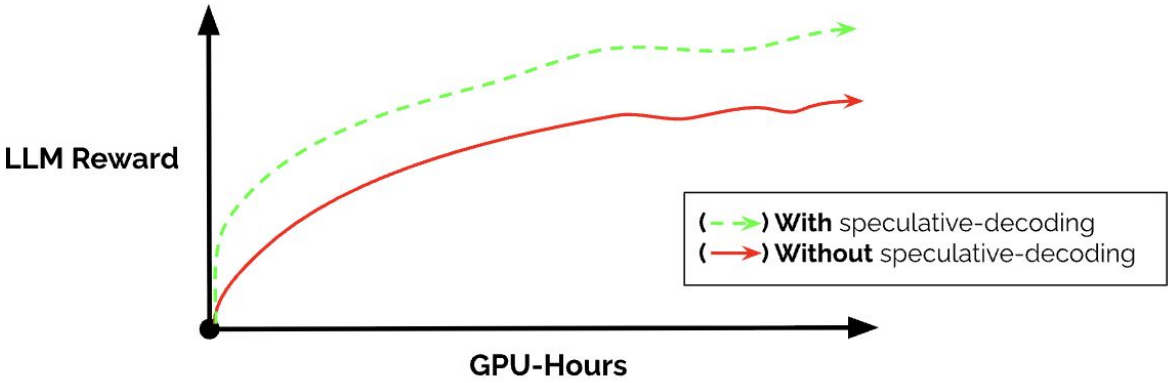
SpecDec enhances test-time scaling by scaling reasoning throughput **losslessly**.



(1) Why Speculative-Decoding? *Reinforcement Learning*

SpecDec enhances **rollouts per gpu-hour** (given smaller batch/group size e.g 16-32 rollouts per prompt)

unlocking faster reinforcement learning.



(2) Improving Speculative-Decoding: *SpecDiff-2*

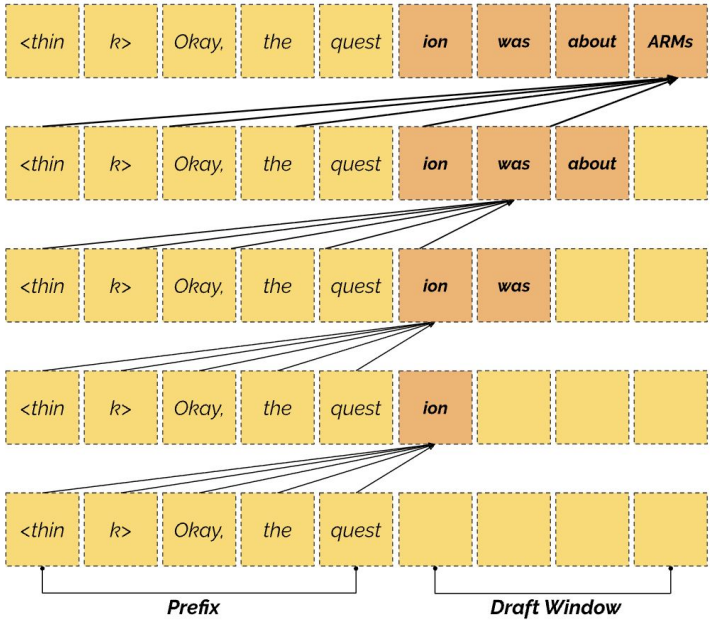
(Contributions)

i. Introduce novel training objective for learning ‘diffusion drafters’ (**steak-distillation**).

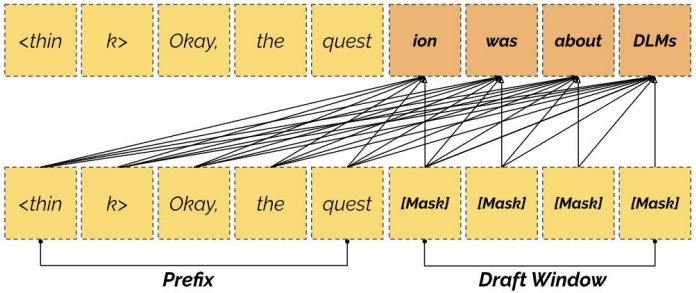
ii. Showcase **SoTA throughput** levels with ‘diffusion drafters’.

(2) Improving Speculative-Decoding: *Diffusion Drafters*

Autoregressive Models (**ARMs**) generate tokens serially, one at a time. Resulting in poor *GPU-utilization*.



Diffusion Language Models (**DLMs**) *parallelize* token generation for higher throughput potential.

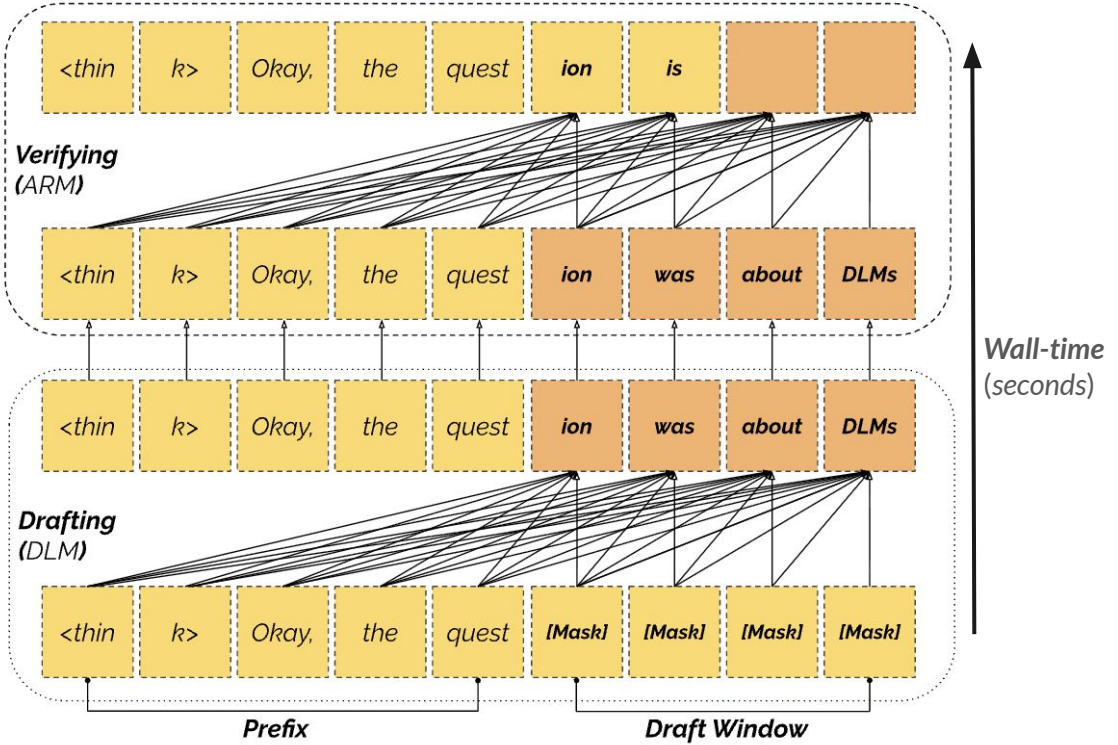


(2) Improving Speculative-Decoding: *Diffusion Drafters*

Diffusion Language Models make for very fast drafters, unlocking high sequence throughput and GPU-utilization:

Speculative-Diffusion (*SpecDiff*)

Speculative-Diffusion unlocks end-to-end *parallel language generation*, with **no quality loss**.



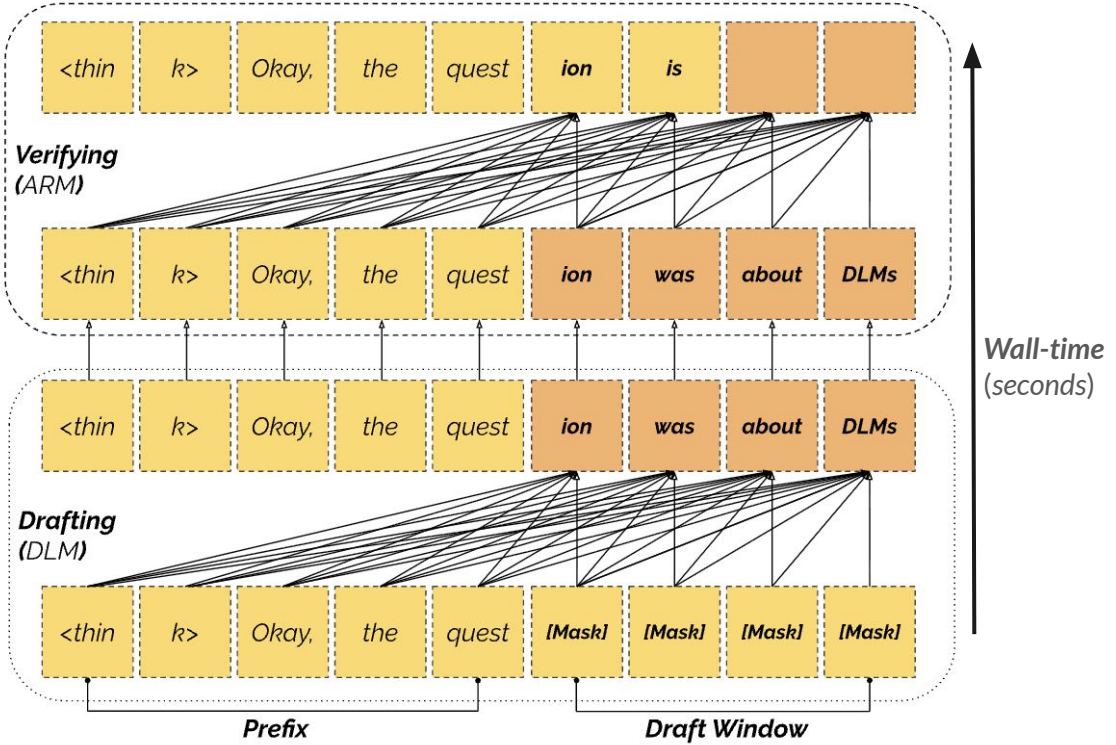
(2) Improving Speculative-Decoding: *Diffusion Drafters*

Diffusion Language Models make for very fast drafters, unlocking high sequence throughput and GPU-utilization:

Speculative-Diffusion (*SpecDiff*)

Speculative-Diffusion unlocks end-to-end *parallel language generation*, with **no quality loss**.

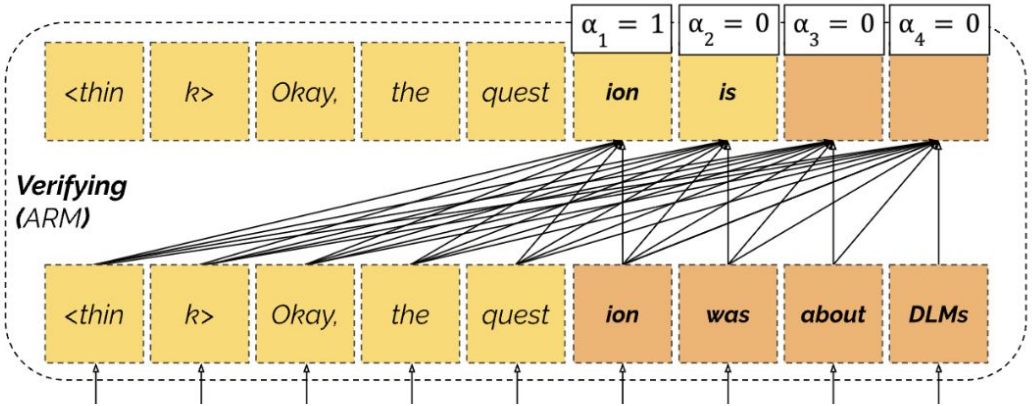
But how do we train this system?



(2) SpecDiff-2: *Introducing Steak-distillation*

Drafted tokens that are ‘approved’ by the verifier are ‘**accepted**’.

The *acceptance probability* for a token in draft position *i* is denoted α_i

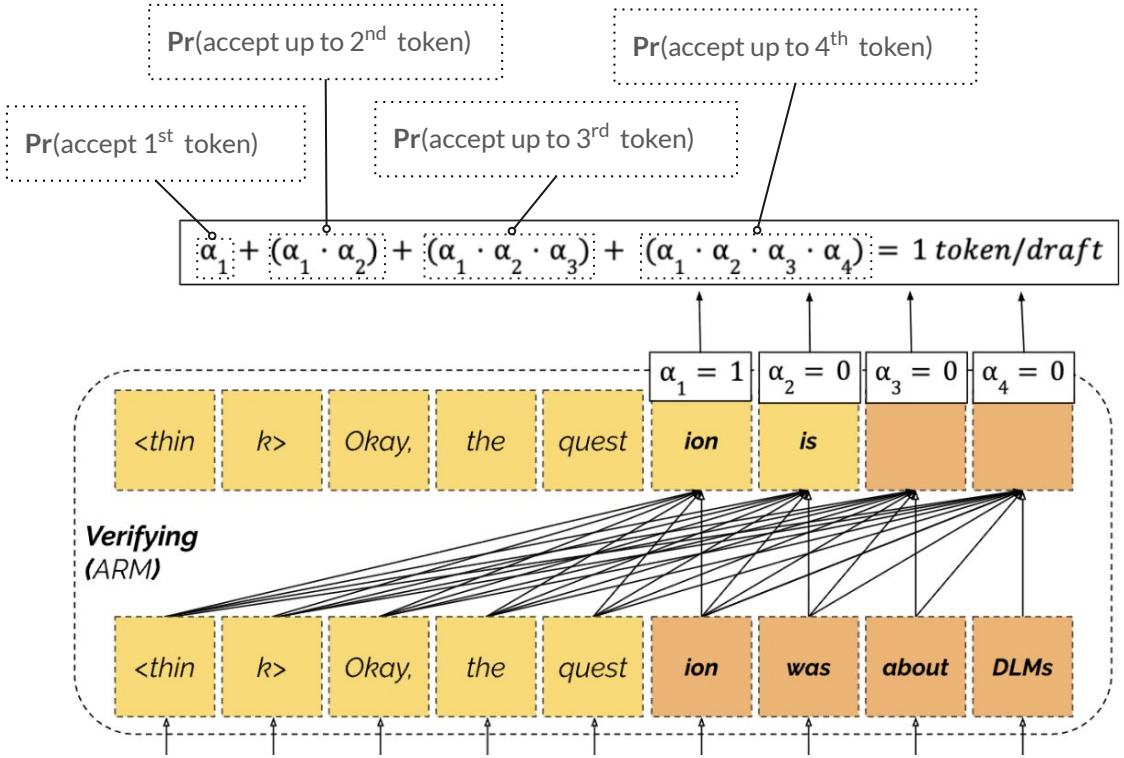


(2) SpecDiff-2: *Introducing Steak-distillation*

Drafted tokens that are 'approved' by the verifier are '**accepted**'.

The *acceptance probability* for a token in draft position *i* is denoted α_i

We delete all subsequent tokens after the first rejection point, thus the expected-accepted tokens (*tokens/draft*) is computable via *a sum-product*.

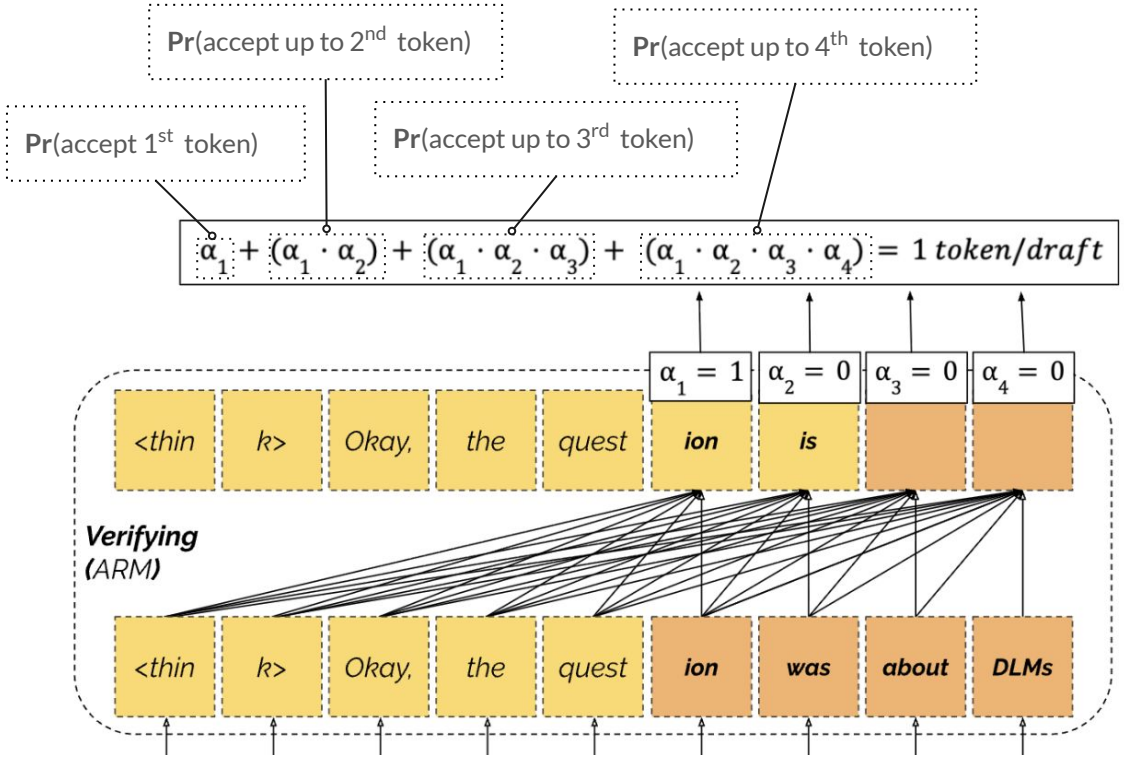


(2) SpecDiff-2: *Introducing Steak-distillation*

Generalizing this (**for draft window γ**) yields the *streak-equation*:

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}[\alpha_i]$$

Where **throughput** scales exactly with *tokens/step*.



(2) SpecDiff-2: *Computing Throughput*

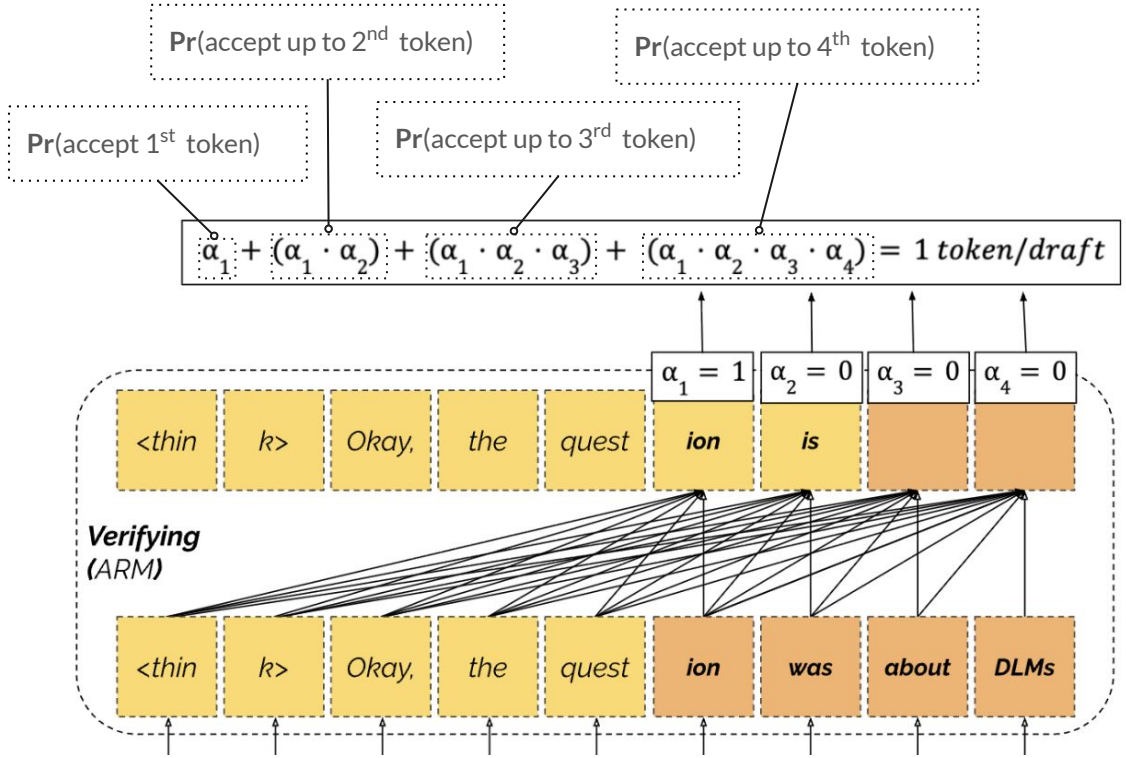
Generalizing this (**for draft window γ**) yields the *streak-equation*:

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}[\alpha_i]$$

$$\alpha_i = \frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|$$

***Diff**usion Posterior*

***A**uto**R**egressive Posterior*



(2) SpecDiff-2: *Deriving Steak-distillation*

Generalizing this (**for draft window** γ) yields the *streak-equation*:

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}[\alpha_i]$$

$$\alpha_i = \frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|$$

***Diff**usion Posterior*

***A**uto**R**egressive Posterior*

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|\right]$$

L1 Error is differentiable *w.r.t* **DLM** (almost) everywhere.

(2) SpecDiff-2: *Deriving Steak-distillation*

Generalizing this (**for draft window γ**) yields the *streak-equation*:

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}[\alpha_i]$$

$$\alpha_i = \frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|$$

***Diff**usion Posterior*

*Auto**R**egressive Posterior*

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|\right]$$

L1 Error is differentiable *w.r.t* **DLM** (almost) everywhere.

$$\max_{\theta}\left[tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM_\theta) - Pr(x_i \sim ARM)|\right]\right]$$

(2) SpecDiff-2: *Deriving Steak-distillation*

Generalizing this (**for draft window γ**) yields the *streak-equation*:

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}[\alpha_i]$$

$$\alpha_i = \frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|$$

***Diff**usion Posterior*

*Auto**R**egressive Posterior*

$$tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM) - Pr(x_i \sim ARM)|\right]$$

Allows us to perform gradient-ascent on expected throughput!

$$\max_{\theta}\left[tokens/step = \sum_{j=1}^{j=\gamma} \prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM_\theta) - Pr(x_i \sim ARM)|\right]\right]$$

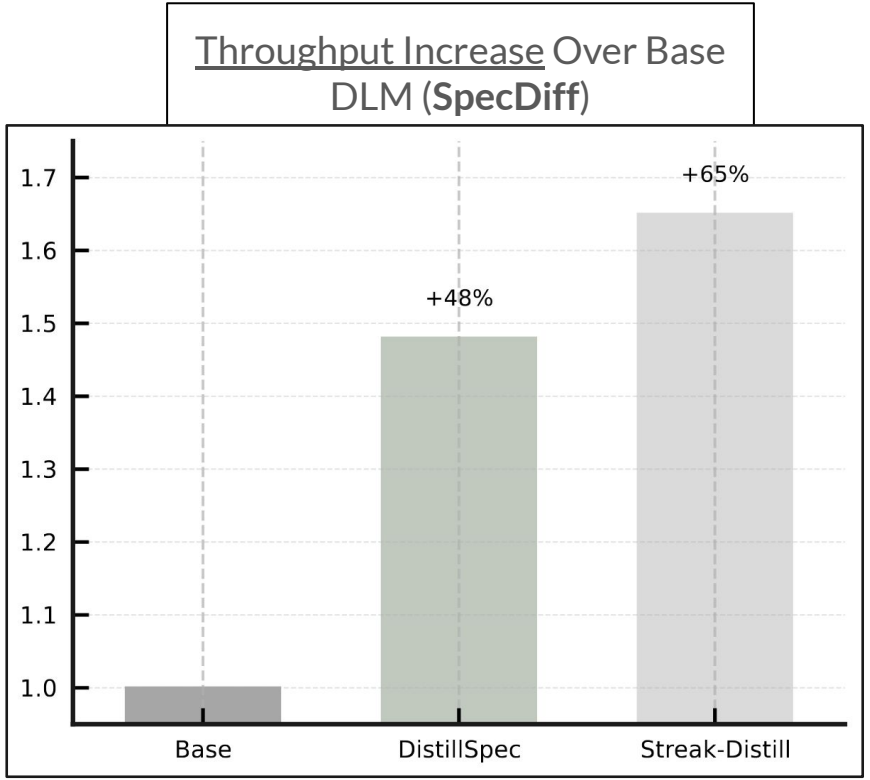
(2) SpecDiff-2: *Streak-Distillation*

Streak-distillation enables gradient ascent on expected throughput.

(i) Outperforms ordinary *knowledge-distillation* due to theoretically motivated position-wise weighting.

(ii) Results in scalable throughput improvement, besting *AR-SpecDec approaches*.

$$\max_{\theta}\left[tokens/step = \sum_{j=1}^{j=\gamma}\prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM_\theta) - Pr(x_i \sim ARM)|\right]\right]$$



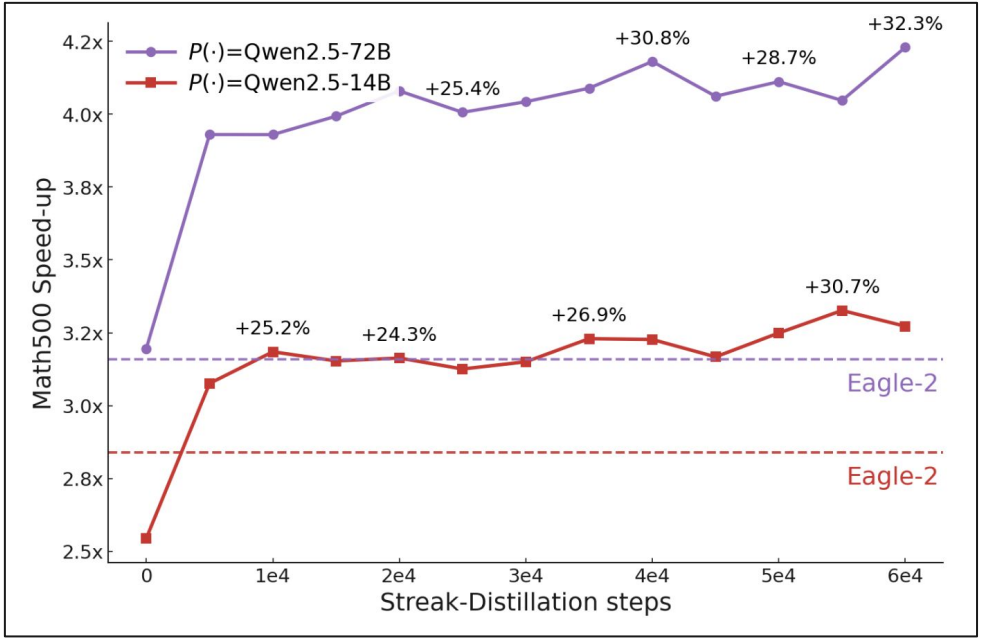
(2) SpecDiff-2: *Streak-Distillation*

Streak-distillation enables gradient ascent on expected throughput.

(i) Outperforms ordinary *knowledge-distillation* due to theoretically motivated position-wise weighting.

(ii) Results in scalable throughput improvement, besting *AR SpecDec approaches.*

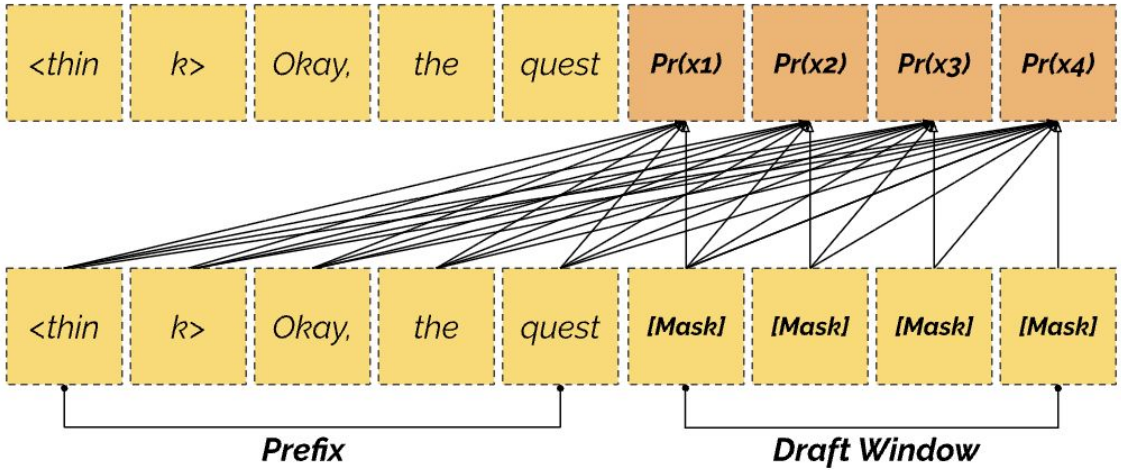
$$\max_{\theta}\left[tokens/step = \sum_{j=1}^{j=\gamma}\prod_{i=1}^{i=j} E_{prefix}\left[\frac{1}{2}|Pr(x_i \sim DLM_\theta) - Pr(x_i \sim ARM)|\right]\right]$$



(2) SpecDiff-2: *Self-selection*

Can also use diffusion-drafters to construct *draft trees* efficiently!

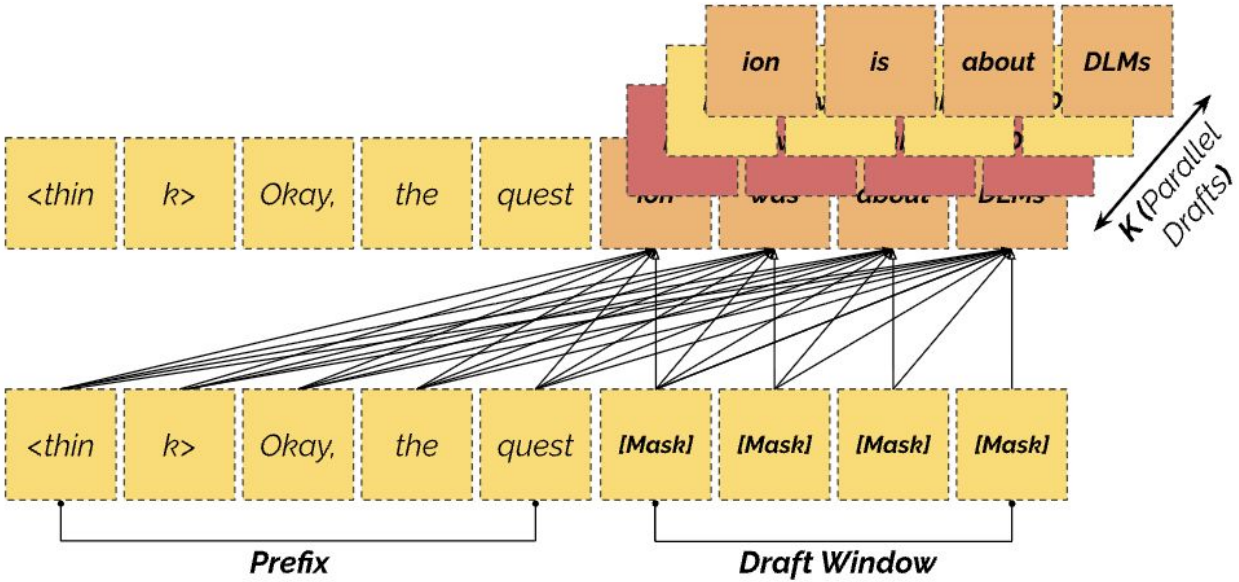
- DLMs actually generate *probabilities*, not tokens.



(2) SpecDiff-2: *Self-selection*

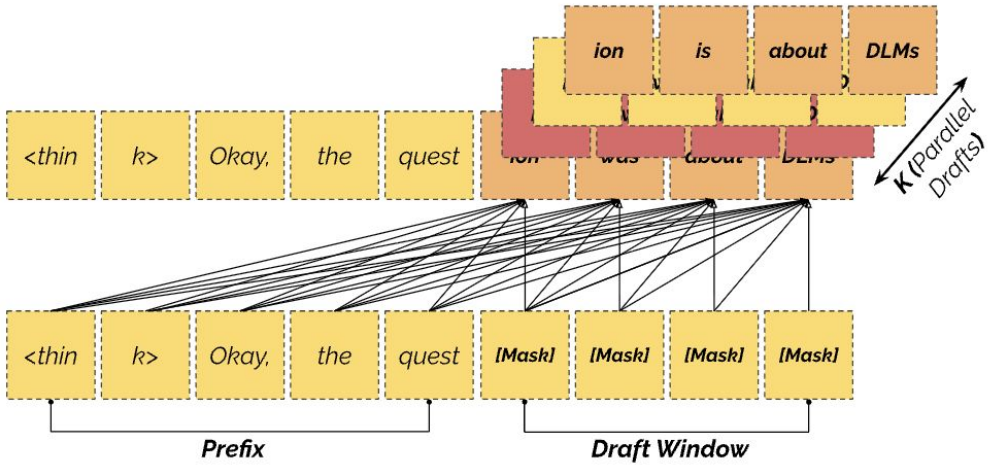
Can also use diffusion-drafters to construct *draft trees* efficiently!

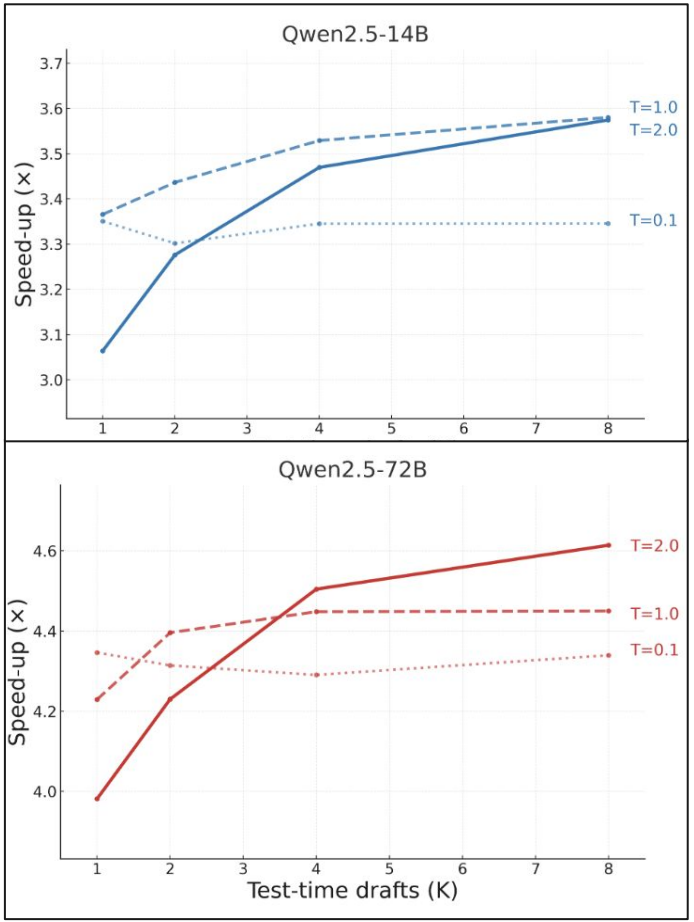
- DLMs actually generate *probabilities*, not tokens.
- Can sample ***K*** times from those probabilities at ~0 cost, then verify in parallel.



(2) SpecDiff-2: *Self-selection*

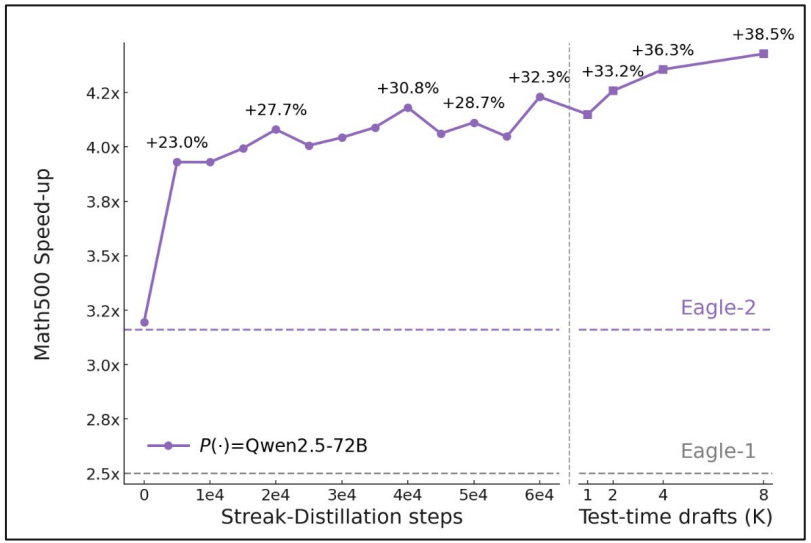
Unlocks higher throughput scaling (speed-up) w.r.t ***K***.





(2) SpecDiff-2: *All Together*

SpecDiff-2 unlocks scalable throughput improvement.



Model	Accelerator	Math-500 Speed-up	Math-500 Tokens/Draft	HumanEval Speed-up	HumanEval Tokens/Draft	GPQA Speed-up	GPQA Tokens/Draft	Mean Speed-up	Mean Tokens/Draft
Temperature = 0									
Qwen-2.5-72B	SpS	1.87×	1.77 toks	1.75×	1.77 toks	1.47×	1.78 toks	1.70×	1.77 toks
	EAGLE	2.50×	3.78 toks	2.28×	3.34 toks	1.93×	2.79 toks	2.24×	3.30 toks
	EAGLE-2	3.16×	4.69 toks	3.16×	4.87 toks	2.50×	3.67 toks	2.94×	4.41 toks
	SpecDiff-2	**4.62×**	**6.47 toks**	**4.98×**	**6.98 toks**	**3.28×**	**4.59 toks**	**4.29×**	**5.98 toks**
LLaMA-2-70B	SpS	1.32×	1.72 toks	1.39×	1.69 toks	1.15×	1.70 toks	1.29×	1.70 toks
	EAGLE	3.00×	4.03 toks	3.11×	4.28 toks	2.47×	3.34 toks	2.86×	3.88 toks
	EAGLE-2	3.48×	4.69 toks	3.87×	5.29 toks	3.08×	4.18 toks	3.48×	4.72 toks
	SpecDiff-2	**3.61×**	**5.04 toks**	**4.69×**	**6.57 toks**	**3.48×**	**4.87 toks**	**3.93×**	**5.49 toks**
Temperature = 1									
Qwen-2.5-72B	SpS	1.70×	1.87 toks	1.78×	1.96 toks	1.49×	1.98 toks	1.66×	1.94 toks
	EAGLE	2.19×	3.38 toks	2.16×	3.20 toks	1.69×	2.50 toks	2.01×	3.03 toks
	EAGLE-2	3.12×	4.63 toks	3.15×	4.77 toks	2.52×	3.67 toks	2.93×	4.36 toks
	SpecDiff-2	**5.01×**	**7.00 toks**	**5.51×**	**7.71 toks**	**2.65×**	**3.71 toks**	**4.39×**	**6.14 toks**
LLaMA-2-70B	SpS	1.39×	1.79 toks	1.39×	1.76 toks	1.24×	1.76 toks	1.34×	1.77 toks
	EAGLE	2.81×	3.80 toks	3.22×	4.38 toks	2.55×	3.38 toks	2.86×	3.85 toks
	EAGLE-2	3.56×	4.75 toks	3.91×	5.33 toks	3.08×	4.20 toks	3.52×	4.76 toks
	SpecDiff-2	**3.99×**	**5.58 toks**	**5.28×**	**7.40 toks**	**3.54×**	**4.95 toks**	**4.27×**	**5.98 toks**

(2) SpecDiff-2: *Discussion*

Where is Speculative-Diffusion Research going?

1. **Better defined drafting processes**
   a. Do we need large diffusion models, can we define more efficient diffusion drafters? (***In progress***)
2. **More unified architectures**
   a. Why must drafting + verification be separate processes?
   b. How do we unlock more unified language modelling approaches?
3. **Application to RLVR**
   a. How can we apply Speculative-diffusion to accelerate RLVR rollouts

(2) SpecDiff-2: *Discussion*

Please reach out with any other questions,
Feel free to connect with me!

Read The Full Paper

Connect with/contact me