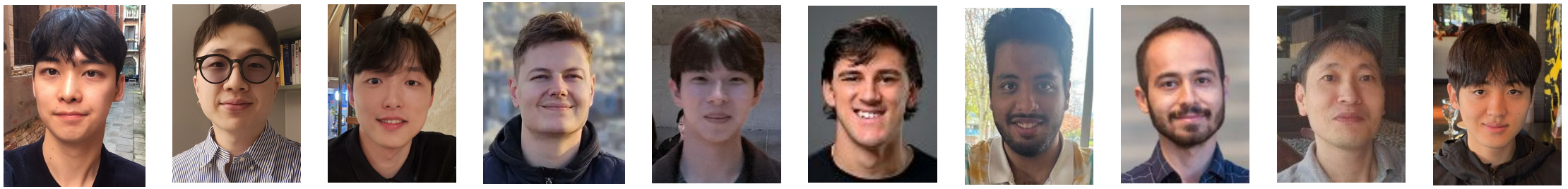
EfficientRollout: System-Aware Self-Speculative Decoding for RL Rollouts



Minseo Kim[1,2*†], Minjae Lee[1*], Seunghyuk Oh[1], Kevin Galim[1], Donghoon Kim[1], Coleman Hooper[2], Harman Singh[2], Amir Gholami[2], Hyung Il Koo[1], Wonjun Kang[1]

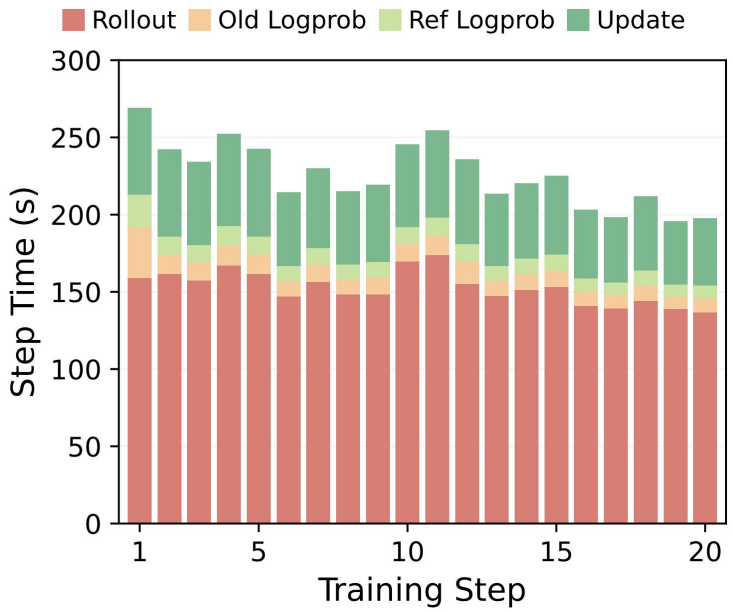
[1]FuriosaAI, [2]UC Berkeley

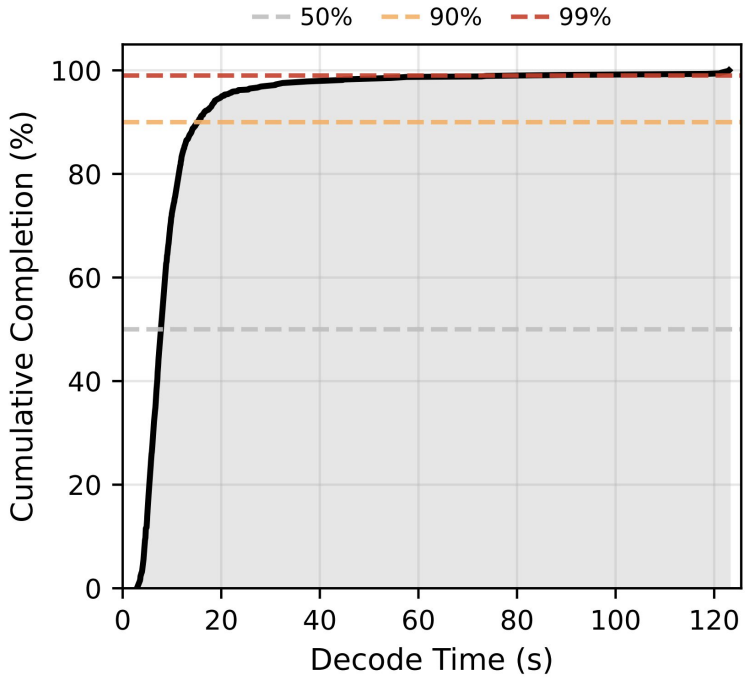
* Equal contribution
† Work done during an internship at FuriosaAI

FURIOSA

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Why RL Rollouts Are the Bottleneck





- RL rollouts account for ~70% of total training time
- Long reasoning trajectories + memory-bound tail decoding
- Most responses finish early; a few determine completion time

Speculative Decoding ?

Why Speculative Decoding Is Not Plug-and-Play for RL

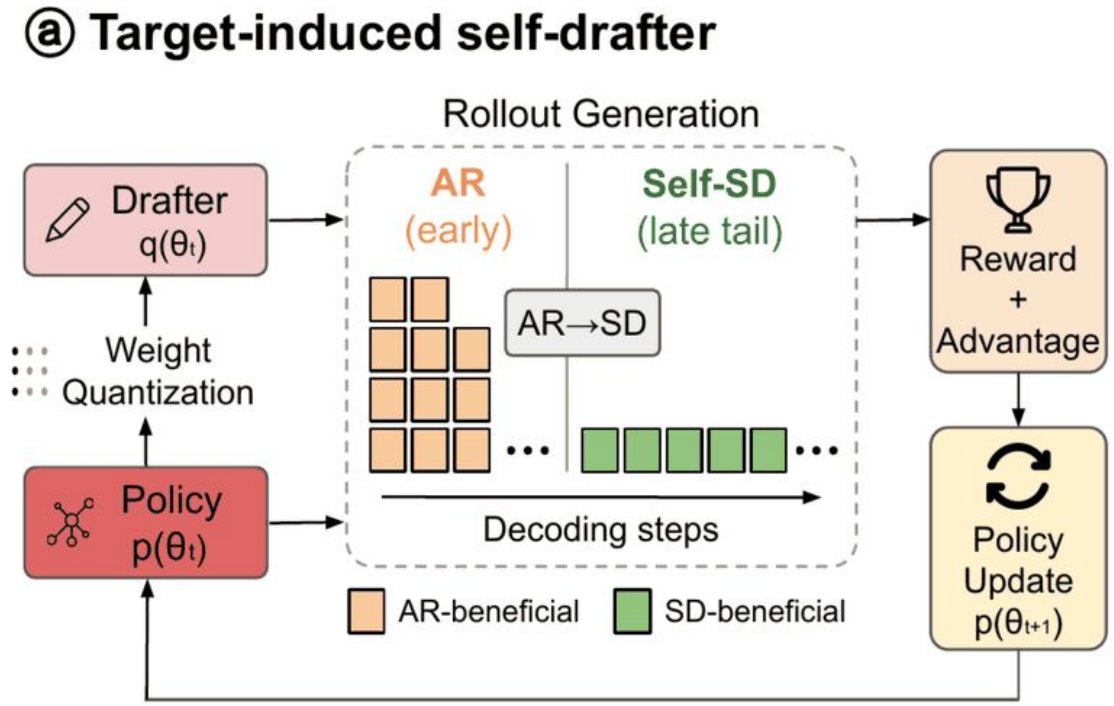
1) Evolving target policy → static drafters become stale
2) Large-batch early phase (compute-bound) → Long-tail phase (memory-bound)

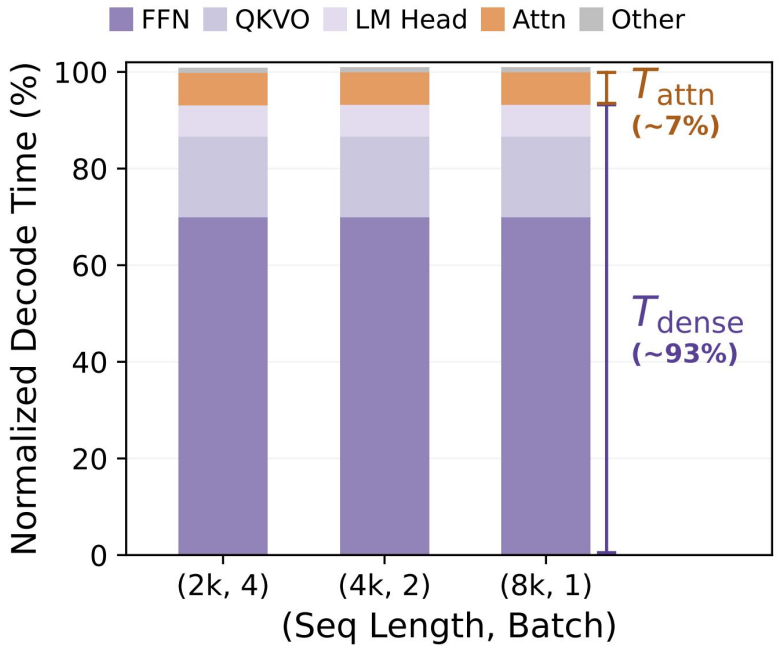


Drafter category	Parameterized	Online-adaptation free	Warm-up free
(1) History-based	No	Yes	No
(2) Learned auxiliary	Yes	No	Conditional
(3) Target-induced	**Yes**	**Yes**	**Yes**

- History-based: limited block efficiency from sparsely covered trajectories
- Learned auxiliary: separate drafter pretraining + continual online adaptation
- **EfficientRollout**: high block efficiency without any additional drafter training

Weight-Quantized Self-Drafter



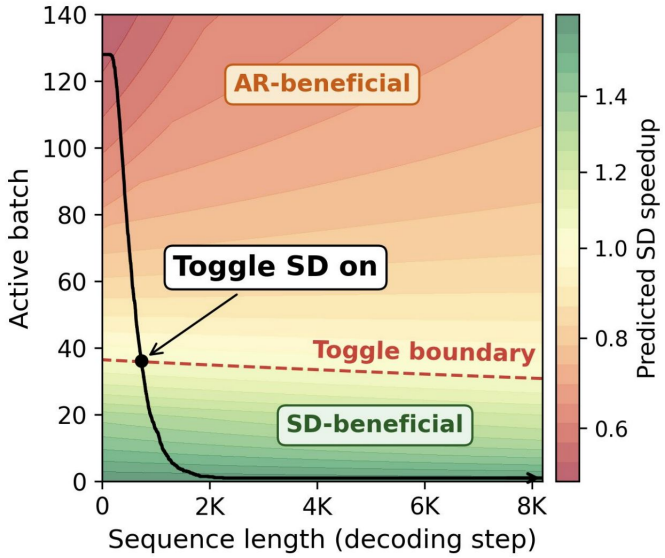


- Derive the drafter **from the current policy** → naturally tracks the evolving policy
- Dense projections ~90% of rollout-tail decoding → **weight-quantized** drafter is effective
- W4 drafting + original-target verification preserves the policy distribution

System-Aware Speculation Control

(Within Each Rollout)

ⓑ Regime-aware SD toggle



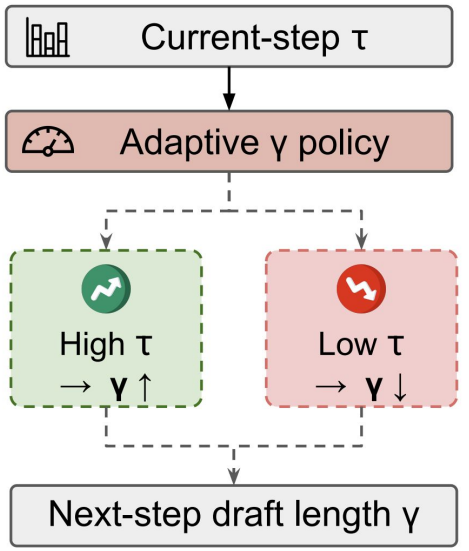
$$T_{\mathrm{SD}}(B,S,\gamma) = \gamma T_q(B,S) + T_V(B,S,\gamma),$$

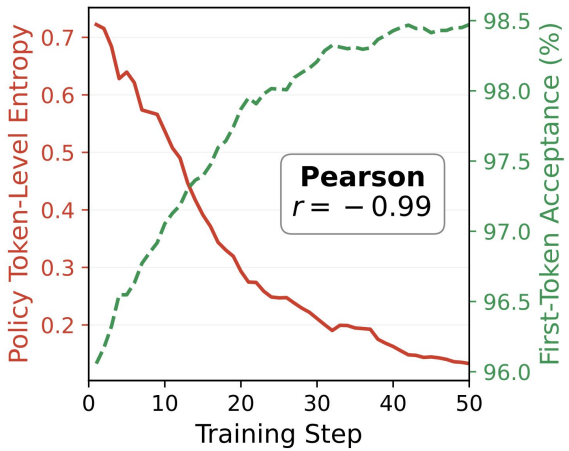
$$\mathrm{Speedup}_{\mathrm{SD}}(B,S,\gamma) = \tau \frac{T_p(B,S)}{T_{\mathrm{SD}}(B,S,\gamma)}$$

- **Regime-aware SD toggle policy** via roofline modeling
- Early rollout: AR → Tail rollout: Self-SD

(Across Training Steps)

ⓒ Adaptive γ control

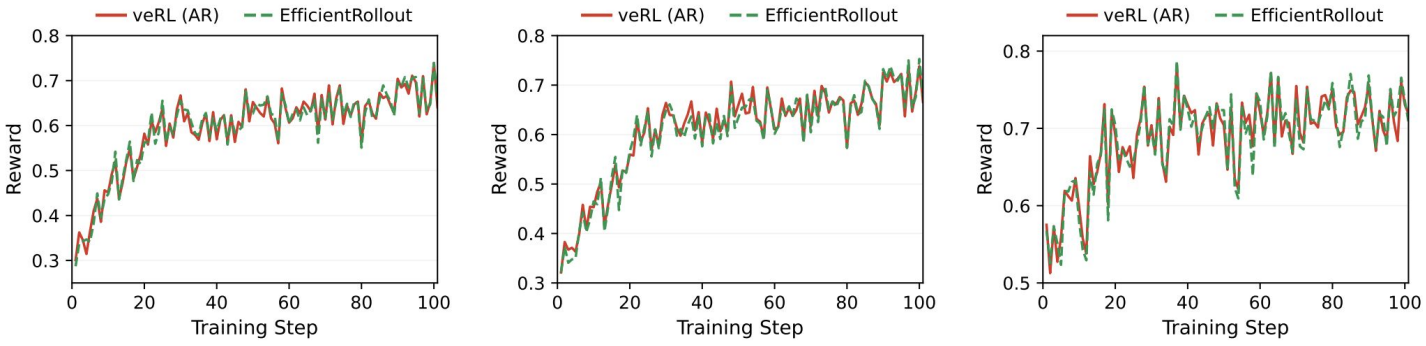




- Quantized drafter becomes more effective over training
- **Adaptive draft-length policy**

Main Results

Model	Drafting Method	Prep. (s) ↓	τ / $\bar{\gamma}$	α (%) ↑	Rollout Gen. (s) ↓	Step Time (s) ↓
Qwen2.5-7B	veRL (AR)	–	–	–	82.4	132.6
	History-based†	**1.1**	N/A	N/A	86.5 (+4.9%)	133.8 (+0.9%)
	Learned auxiliary	2.2	2.1 / 3.0	57.3	81.9 (-0.6%)	132.1 (-0.4%)
	Quantized self-SD	1.3	7.5 / 7.0	**98.1**	66.1 (-19.7%)	115.7 (-12.8%)
	EfficientRollout	1.3	8.6 / 8.3	**98.1**	**62.4 (-24.2%)**	**111.9 (-15.6%)**
Qwen2.5-14B	veRL (AR)	–	–	–	126.6	221.1
	History-based†	**1.1**	N/A	N/A	129.5 (+2.3%)	218.9 (-1.0%)
	Learned auxiliary	2.1	2.2 / 3.0	60.3	115.4 (-8.9%)	213.8 (-3.3%)
	Quantized self-SD	2.6	5.6 / 5.0	97.3	131.4 (+3.8%)	223.2 (+0.9%)
	EfficientRollout	2.6	7.4 / 7.0	**97.6**	**104.9 (-17.1%)**	**197.7 (-10.6%)**
Llama3.1-8B	veRL (AR)	–	–	–	126.4	186.9
	History-based†	**1.1**	N/A	N/A	131.1 (+3.7%)	187.4 (+0.2%)
	Learned auxiliary‡	2.6	2.0 / 3.0	54.3	172.8 (+36.7%)	234.8 (+25.6%)
	Quantized self-SD	1.4	5.5 / 5.0	**96.4**	116.7 (-7.6%)	178.1 (-4.7%)
	EfficientRollout	1.4	5.3 / 5.0	95.1	**112.2 (-11.2%)**	**173.3 (-7.3%)**



(a) Qwen2.5-7B (b) Qwen2.5-14B (c) Llama3.1-8B

- Up to 24.2% lower rollout latency
- Up to 15.6% lower training-step latency
- Consistently high acceptance rate + block efficiency

- Closely tracks veRL (AR) reward trajectories

Thank you for listening!

Paper

Code

minseo.kim@berkeley.edu

FURIOSA

BAIR

BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH