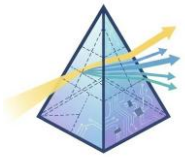
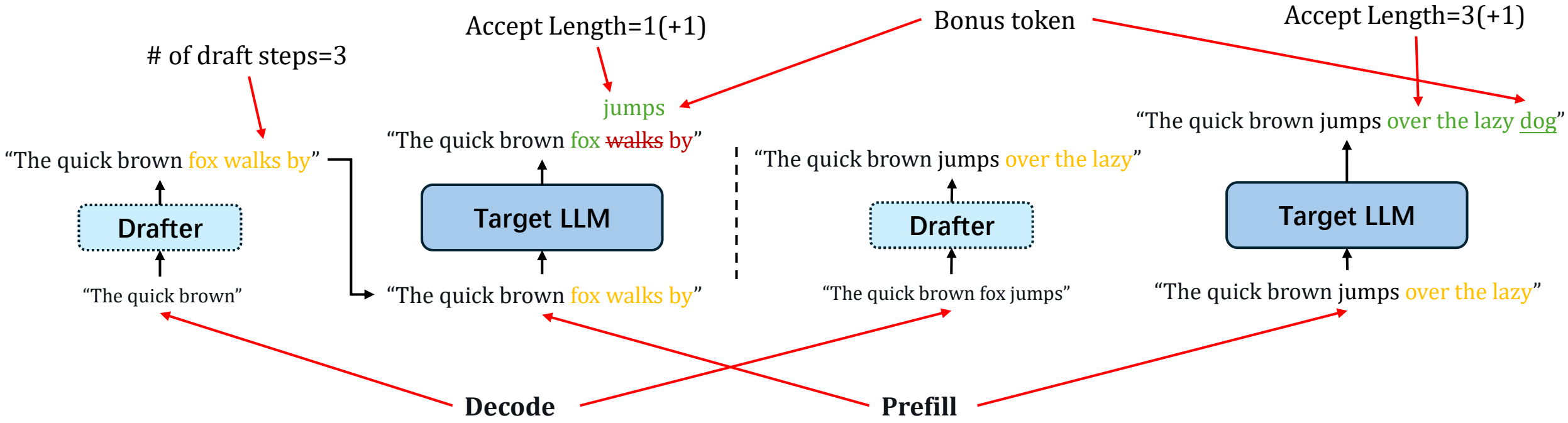


PRISM: Parametrically Refactor Inference for Speculative Decoding Draft Models

Xuliang Wang*, Yuetao Chen*, Maochan Zhen, Fang Liu, Xinzhou Zheng, Xingwu Liu, Hong Xu and Ming Li

* Equal Contribution

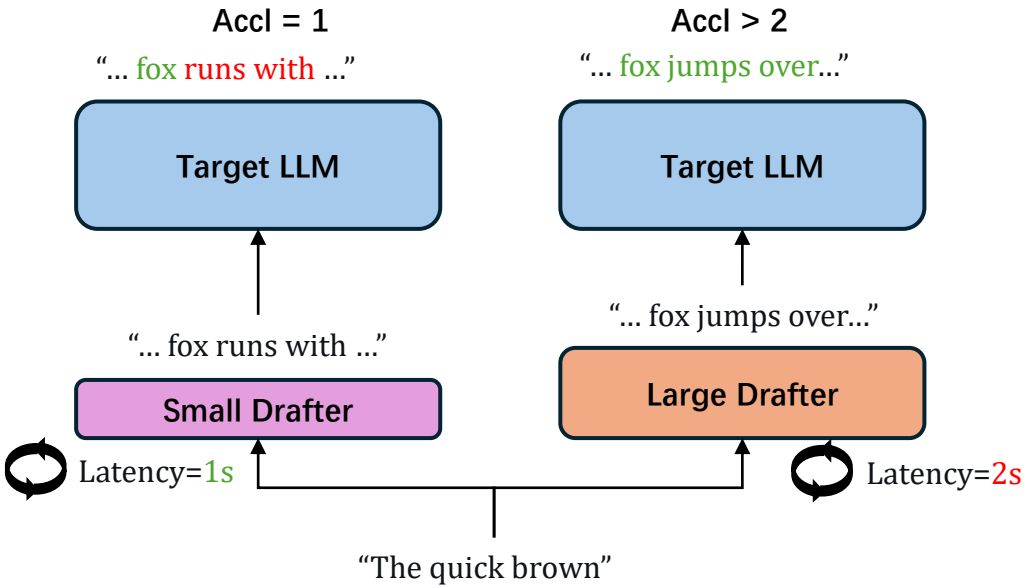
Background: Speculative Decoding



- Efficiency of the autoregressive LLM decoding is bounded by memory access
- Speculative decoding amortizes the memory access cost across multiple accepted tokens

Background: Previous Exploration

- How to get more drafted tokens accepted?
  - Cover more possible drafts → Tree Draft and Verification [Miao et al.]
  - Drafter architectural innovation → Target model feature dependent drafter [Li et al., Zhang et al.]
  - Scaling → Train with more parameters and more data [Li et al., Yan et al., Tang et al.]



- Large Drafter: Better prediction but more latency

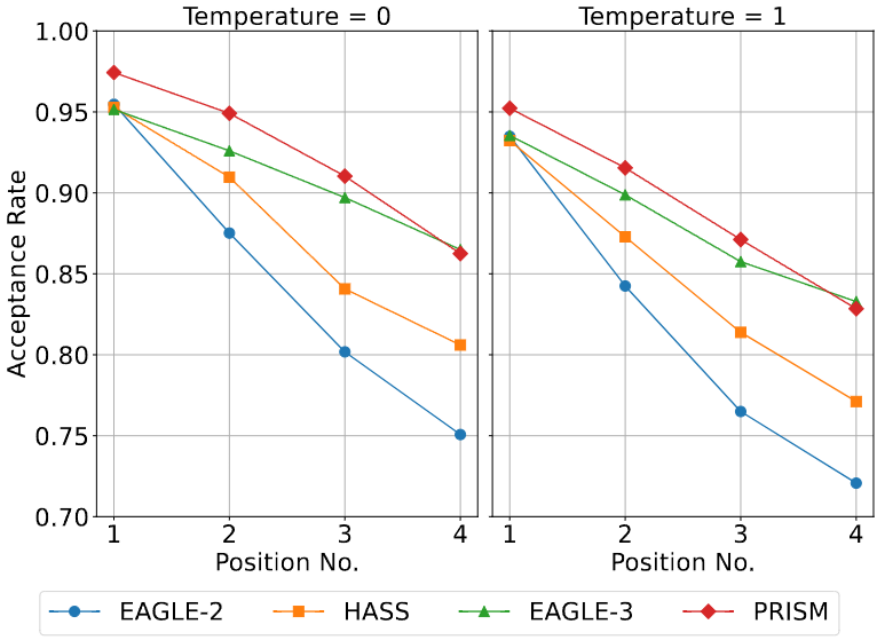
Tradeoff

- Small Drafter: Less latency but worse prediction

Is it possible to expand model capacity while keeping inference latency low?

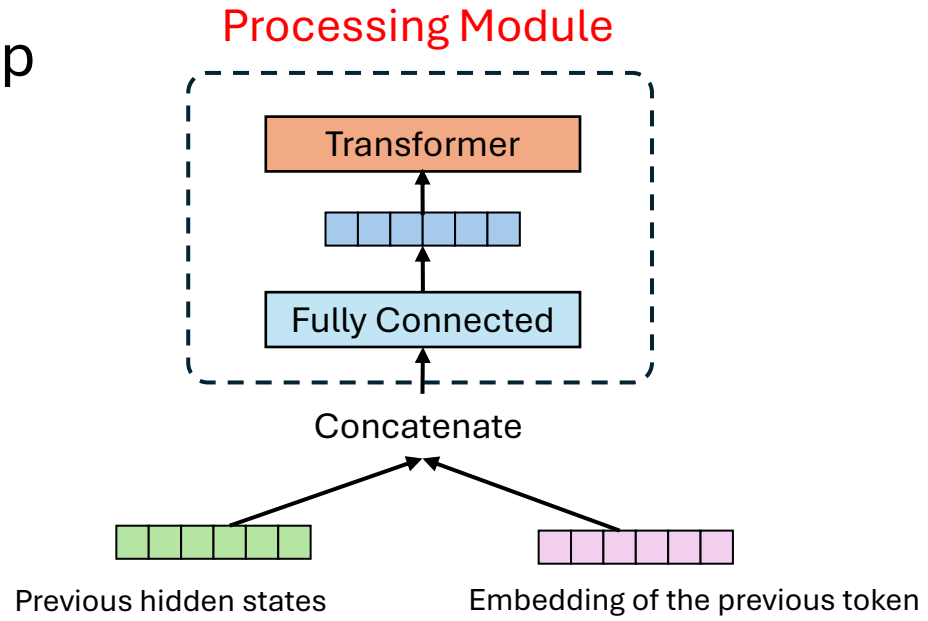
Method: Intuition

- **Observation 1:** Finite number of generation steps for draft models
- **Observation 2:** Heterogeneity of draft steps → **Design opportunity for specialization**



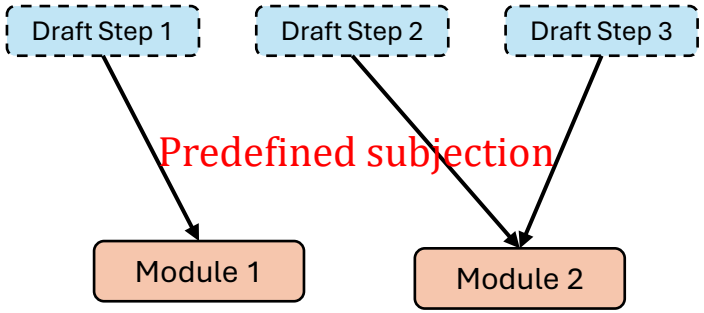
Heterogeneity of draft steps evidenced by varied acceptance rates across steps

Intuition: Activate different parameters for each draft step

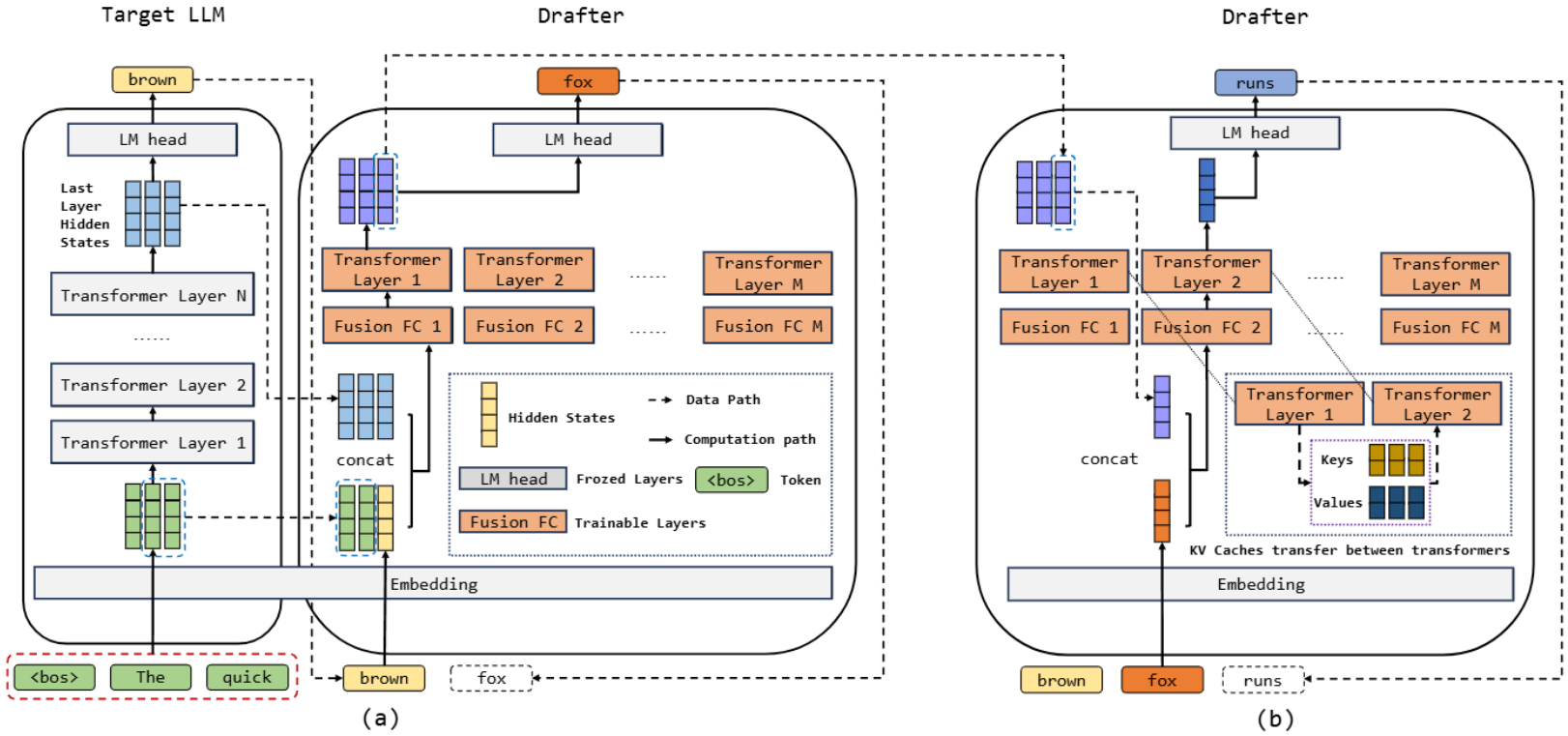


Method: Architecture

of draft steps=3



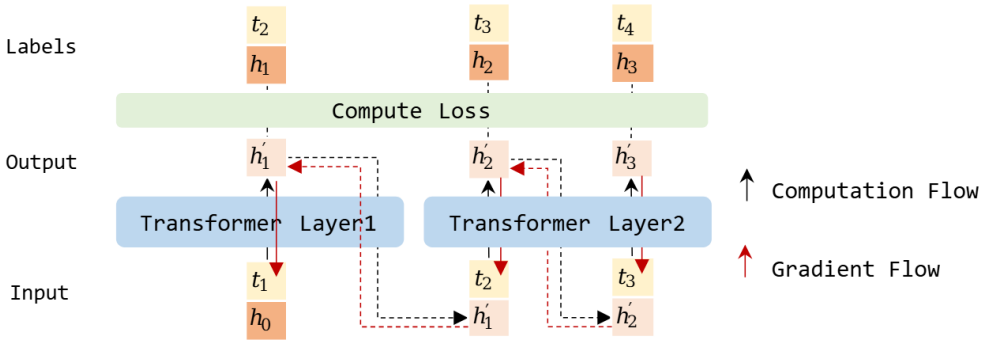
# of processing modules=2



- According to a predefined subjection, switch between processing modules as draft step proceeds
- KV caches transfer between transformer layers as processing modules switch

Method: Training

- **Context alignment:** Simulate and align multiple generation steps of the draft model with target model
- **Two stage training**: Warmup and Tuning
  - **Warmup:** Train one processing module with MSE loss on hidden states and CE loss on logits
  - **Tuning:** Replicate trained processing module then train for parameter switching



Dataset	Volume	Proportion	Topic Coverage
ShareGPT	68K	8.5%	QA/Code/Math/Logic
UltraChat	463K	57.9%	QA
OpenThoughts2	269K	33.6%	Math/Logic

- For OpenThoughts2, thinking contexts are removed
- Train the PRISM draft model with different datasets ranging from 100k to 800k samples on 8 NVIDIA A100 40G GPUs
- 25-40 epochs for warmup and 10 epochs for tuing

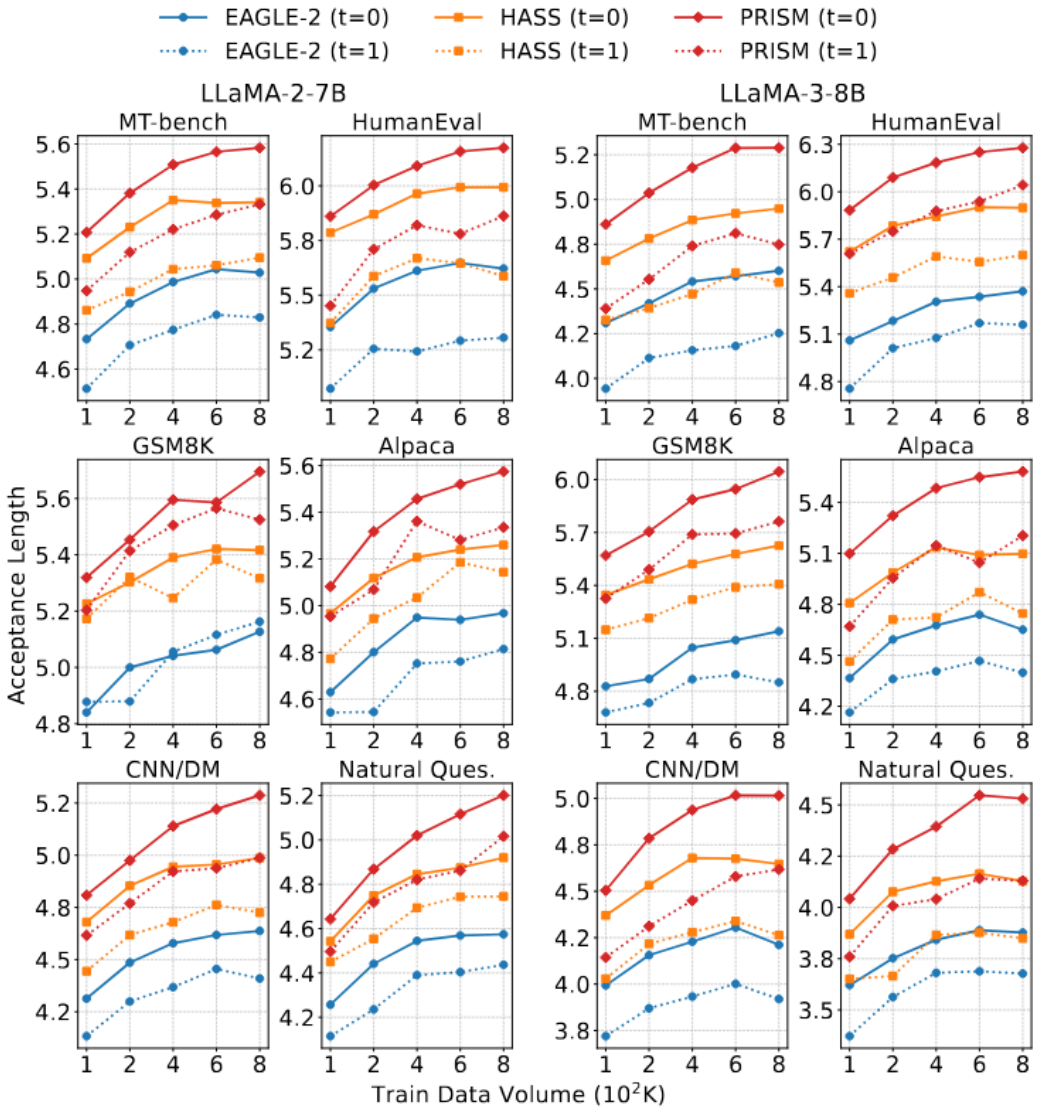
Evaluation: End2End

- Target Models: LLaMA-2-7B, LLaMA-3-8B
- Benchmarks: MT-bench, HumanEval, GSM8K, Alpaca, CNN/DM, Natural Ques.
- Baselines: Standard, EAGLE-2, HASS
- Hardware: 1 A100, 1 H800, 2 RTX 4090

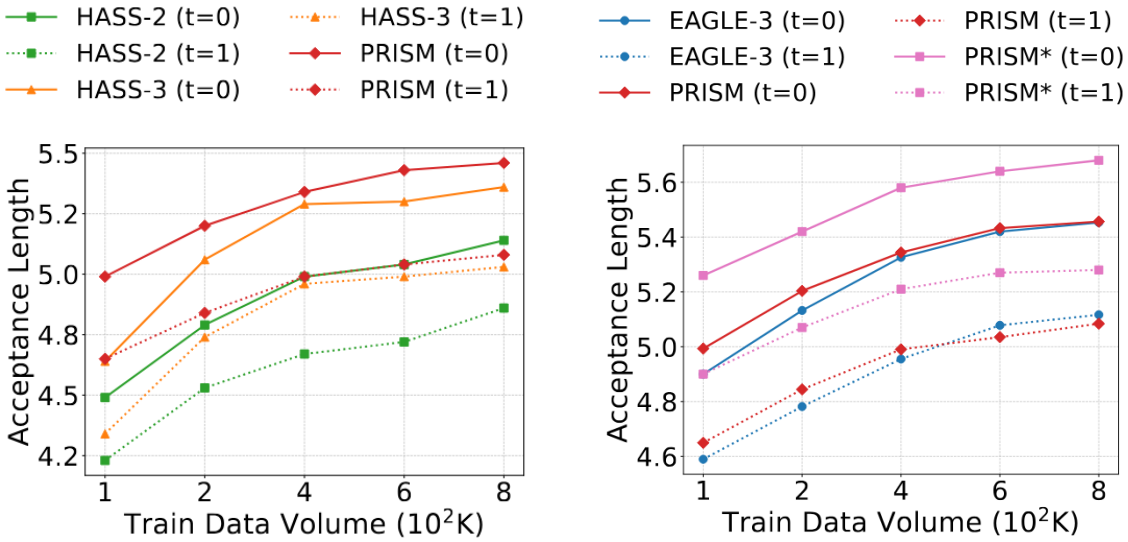
			MT-bench		HumanEval		GSM8K		Alpaca		CNN/DM		Natural Ques.	
Model	Method	Temp	AL	TPS	AL	TPS	AL	TPS	AL	TPS	AL	TPS	AL	TPS
LLaMA-2	Vanilla	T = 0	N/A	142.10	N/A	142.37	N/A	143.08	N/A	143.43	N/A	139.76	N/A	142.99
		T = 1	N/A	142.12	N/A	140.76	N/A	141.70	N/A	141.22	N/A	138.42	N/A	141.47
	Standard	T = 0	2.73	188.02	2.76	193.47	2.91	202.73	2.90	204.16	2.06	141.69	2.87	200.25
		T = 1	2.74	191.44	2.60	180.92	2.82	195.00	2.85	198.09	2.05	139.84	2.75	190.07
	EAGLE-2	T = 0	4.12	335.71	4.72	387.42	4.28	347.95	4.07	333.99	3.86	309.78	3.81	309.43
		T = 1	4.12	337.17	4.37	319.90	4.14	298.77	3.90	282.88	3.69	263.54	3.54	257.02
	HASS	T = 0	4.40	349.75	5.08	408.49	4.54	360.79	4.38	350.96	4.17	327.60	4.05	321.47
		T = 1	4.38	350.88	4.77	342.02	4.38	310.17	4.06	290.11	3.98	279.20	3.85	271.97
	PRISM (ours)	T = 0	**4.67**	**373.94**	**5.36**	**432.93**	**4.77**	**381.51**	**4.68**	**377.50**	**4.40**	**345.78**	**4.32**	**343.94**
		T = 1	**4.68**	**377.55**	**4.82**	**348.27**	**4.68**	**333.45**	**4.40**	**316.14**	**4.21**	**297.57**	**3.97**	**283.70**
LLaMA-3	Vanilla	T = 0	N/A	145.10	N/A	145.34	N/A	145.52	N/A	145.89	N/A	144.03	N/A	145.64
		T = 1	N/A	146.03	N/A	144.10	N/A	144.09	N/A	144.20	N/A	143.32	N/A	143.80
	Standard	T = 0	5.07	241.19	5.89	276.13	5.75	271.18	4.80	228.80	4.72	221.31	4.39	209.10
		T = 1	5.05	241.55	5.41	254.10	5.35	250.34	4.22	199.23	4.21	197.30	3.88	182.53
	EAGLE-2	T = 0	3.72	202.10	4.58	312.94	4.27	291.35	3.67	253.10	3.57	243.60	3.25	223.44
		T = 1	3.72	257.92	4.31	210.01	3.98	192.28	3.41	165.85	3.32	161.04	2.99	146.24
	HASS	T = 0	4.00	278.70	5.08	349.65	4.75	326.78	3.93	273.59	3.91	268.37	3.40	236.46
		T = 1	4.00	280.60	4.81	235.42	4.51	220.36	3.64	179.08	3.55	174.53	3.08	152.45
	PRISM (ours)	T = 0	**4.63**	**315.42**	**5.83**	**392.05**	**5.51**	**369.40**	**4.53**	**308.54**	**4.60**	**309.17**	**3.91**	**266.71**
		T = 1	**4.62**	**316.13**	**5.30**	**255.74**	**5.03**	**241.16**	**4.03**	**195.39**	**4.08**	**196.56**	**3.51**	**170.65**

- Accept Length:
  - +21.8% vs Standard
  - +13.8% vs EAGLE2
  - +9.8% vs HASS
- Token per Second:
  - +51% vs Standard
  - +17.2% vs EAGLE2
  - +8.8% vs HASS
- Memory footprint:
  - +0.77% vs EAGLE2, HASS

Evaluation: Scaling

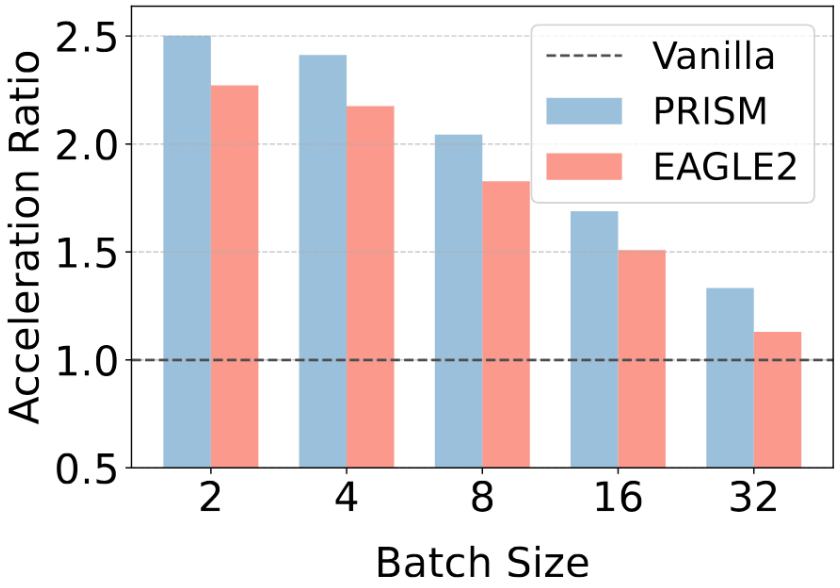


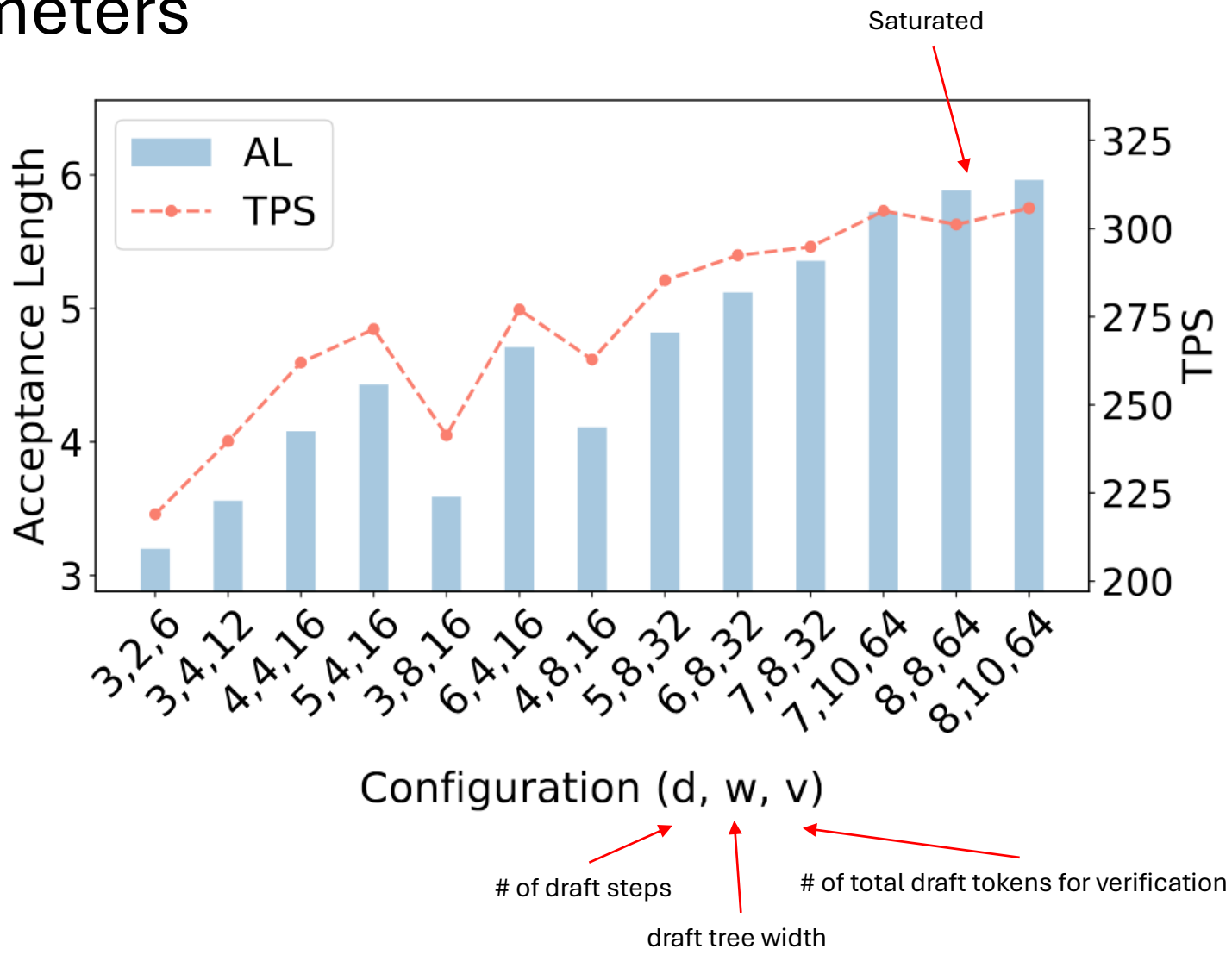
- 6-step, 10-branch tree to demonstrate prediction power upper bound
- Better Scaling curves for PRISM Compared to all baselines
- Stacking parameters helps improving scalability; PRISM is even more efficient at scaling



PRISM*: Employ multiple previous hidden states as input

Evaluation: Hyper-Parameters





- Acceleration degrades as batch size grows

Discussion

- Design opportunities for draft model: Finite and heterogenous draft steps
- Speculative drafter design as a predictor refinement task
- New drafter architectures coming out: Diffusion

Thank you!