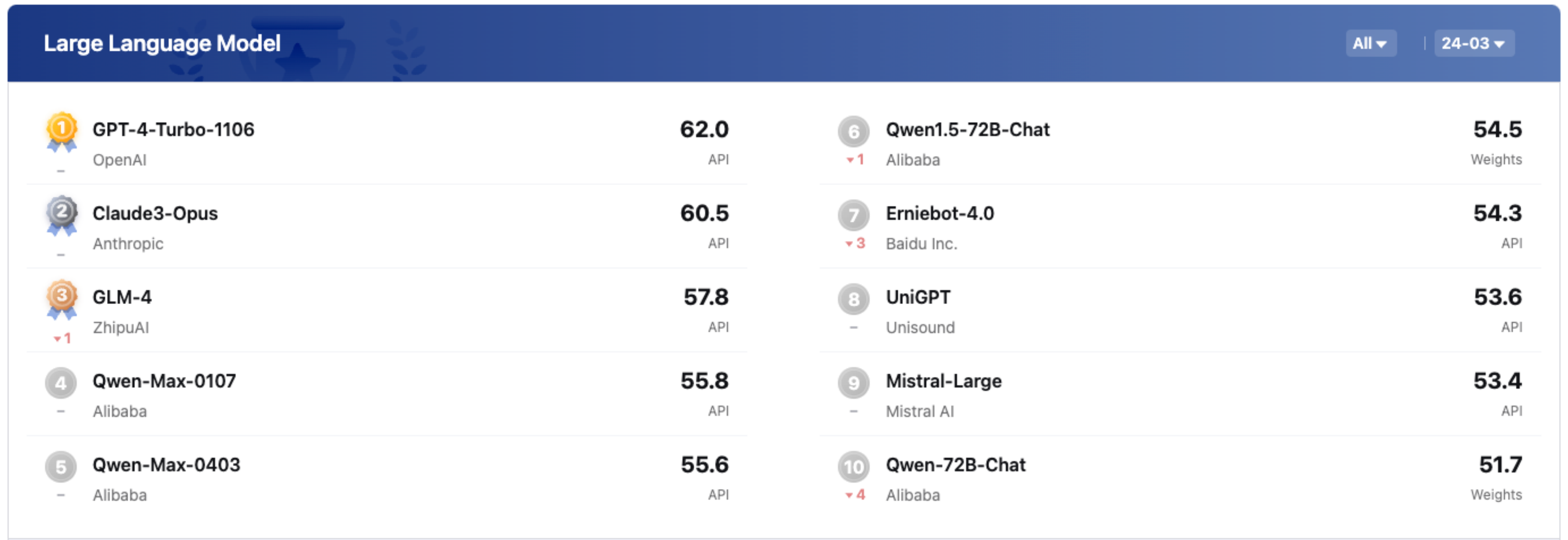
大语言模型推理与训练协同演进

—— 探索高效推理技术的新篇章

算法开发工程师 / 谢帅

LLM 训练与推理蓬勃发展

LLM 综合能力 OpenCompass Leaderboad (GPT–3.5–Turbo 46.5, rank 15)



Large Language Model All 24-03

Rank	Model	Score	Rank	Model	Score
1	GPT-4-Turbo-1106 OpenAI	62.0 API	6 (▾1)	Qwen1.5-72B-Chat Alibaba	54.5 Weights
2	Claude3-Opus Anthropic	60.5 API	7 (▾3)	Erniebot-4.0 Baidu Inc.	54.3 API
3 (▾1)	GLM-4 ZhipuAI	57.8 API	8	UniGPT Unisound	53.6 API
4	Qwen-Max-0107 Alibaba	55.8 API	9	Mistral-Large Mistral AI	53.4 API
5	Qwen-Max-0403 Alibaba	55.6 API	10 (▾4)	Qwen-72B-Chat Alibaba	51.7 Weights

LLM 中文对齐能力 Align Bench (GPT–3.5–Turbo 6.08, rank 5)

model 模型	Overall 总分	Reasoning 中文推理			Language 中文语言						
		Avg. 推理总分	Math. 数学计算	Logi. 逻辑推理	*Avg.* 语言总分	Fund. 基本任务	Chi. 中文理解	Open. 综合问答	Writ. 文本写作	Role. 角色扮演	Pro. 专业能力
gpt-4-1106-preview	**8.01**	*7.73*	**7.80**	**7.66**	**8.29**	**7.99**	7.33	**8.61**	**8.67**	**8.47**	**8.65**
gpt-4-0613	**7.53**	*7.47*	7.56	7.37	7.59	7.81	6.93	7.42	7.93	7.51	7.94
chatglm-turbo（智谱清言）	**6.24**	*5.00*	4.74	5.26	7.49	6.82	7.17	8.16	7.77	7.76	7.24
erniebot-3.5（文心一言）	**6.14**	*5.15*	5.03	5.27	7.13	6.62	**7.60**	7.26	7.56	6.83	6.90
gpt-3.5-turbo-0613	**6.08**	*5.35*	5.68	5.02	6.82	6.71	5.81	7.29	7.03	7.28	6.77
chatglm-pro（智谱清言）	**5.83**	*4.65*	4.54	4.75	7.01	6.51	6.76	7.47	7.07	7.34	6.89
spark_desk_v2（讯飞星火）	**5.74**	*4.73*	4.71	4.74	6.76	5.84	6.97	7.29	7.18	6.92	6.34
Qwen-14B-Chat	**5.72**	*4.81*	4.91	4.71	6.63	6.90	6.36	6.74	6.64	6.59	6.56
Baichuan2-13B-Chat	**5.25**	*3.92*	3.76	4.07	6.59	6.22	6.05	7.11	6.97	6.75	6.43
ChatGLM3-6B	**4.97**	*3.85*	3.55	4.14	6.10	5.75	5.29	6.71	6.83	6.28	5.73
Baichuan2-7B-Chat	**4.97**	*3.66*	3.56	3.75	6.28	5.81	5.50	7.13	6.84	6.53	5.84
InternLM-20B	**4.96**	*3.66*	3.39	3.92	6.26	5.96	5.50	7.18	6.19	6.49	6.22
Qwen-7B-Chat	**4.91**	*3.73*	3.62	3.83	6.09	6.40	5.74	6.26	6.31	6.19	5.66
ChatGLM2-6B	**4.48**	*3.39*	3.16	3.61	5.58	4.91	4.52	6.66	6.25	6.08	5.08
InternLM-Chat-7B	**3.65**	*2.56*	2.45	2.66	4.75	4.34	4.09	5.82	4.89	5.32	4.06
Chinese-LLaMA-2-7B-Chat	**3.57**	*2.68*	2.29	3.07	4.46	4.31	4.26	4.50	4.63	4.91	4.13
LLaMA-2-13B-Chinese-Chat	**3.35**	*2.47*	2.21	2.73	4.23	4.13	3.31	4.79	3.93	4.53	4.71

其他榜单

SafetyBench

AgentBench

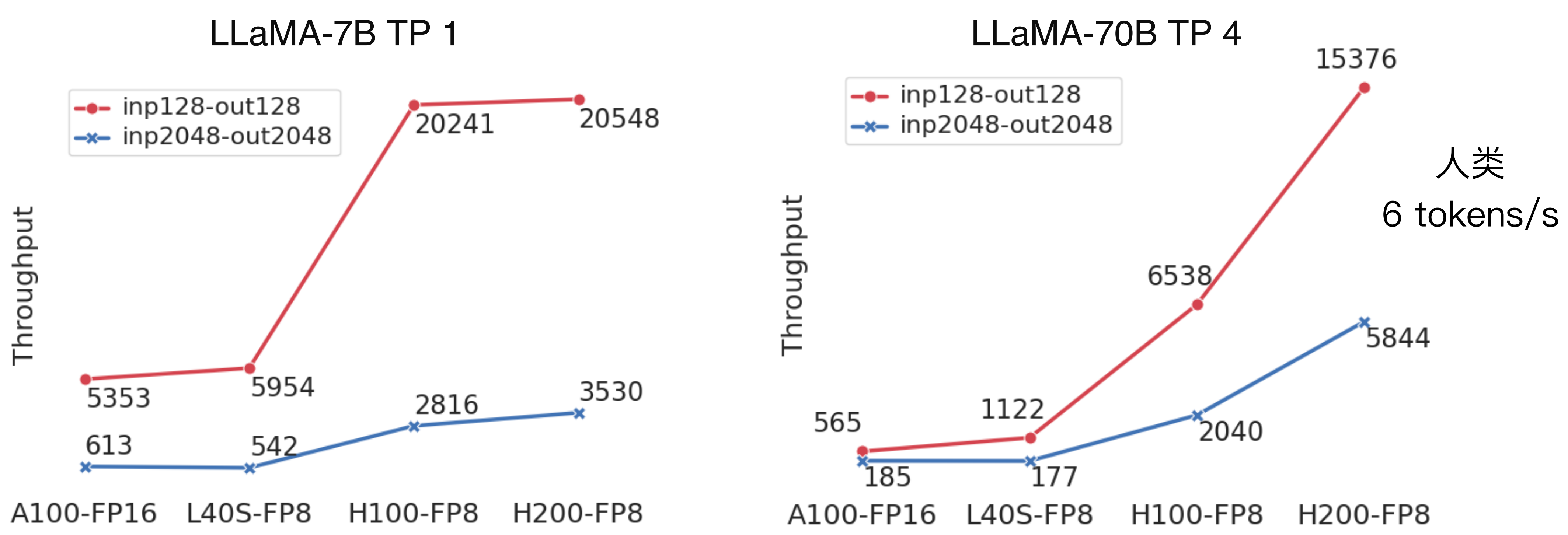
BigBench

MTBench

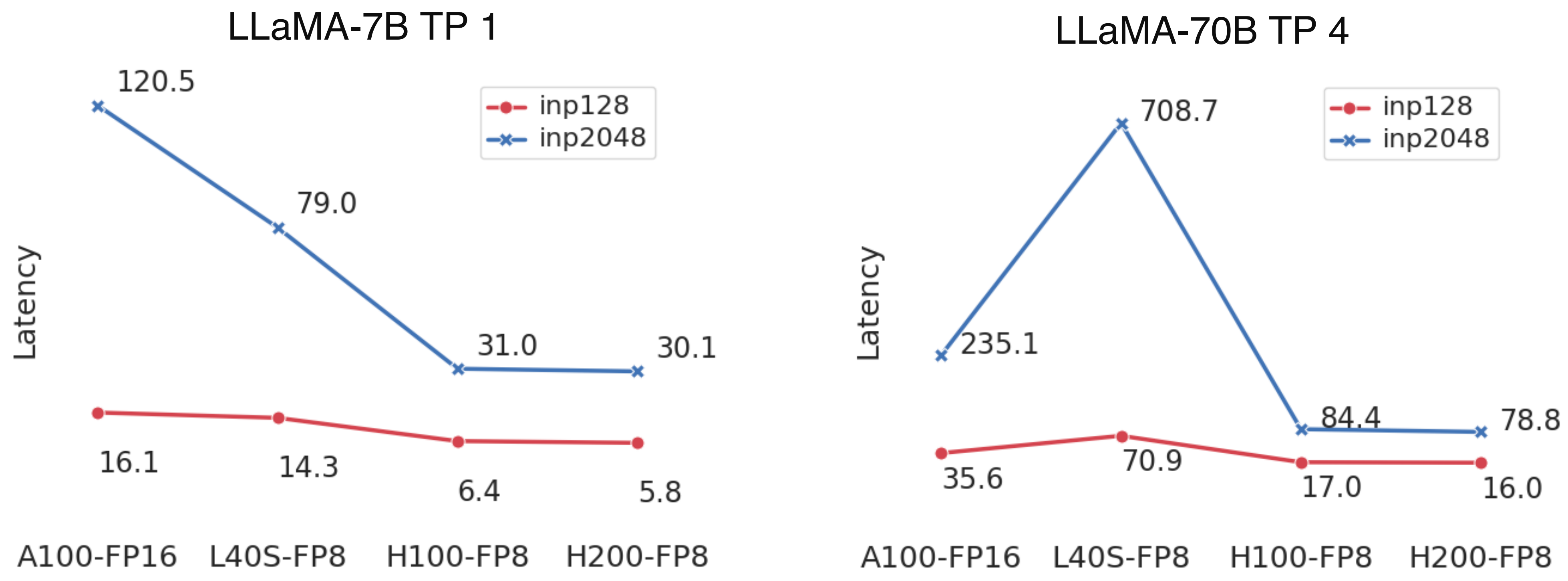
AlpacaEval

…

LLM 推理性能 Peak Throughput (TPS)



LLM 推理性能 Low Latency (TTFT ms)



数据来自 TensorRT-LLM 推理框架

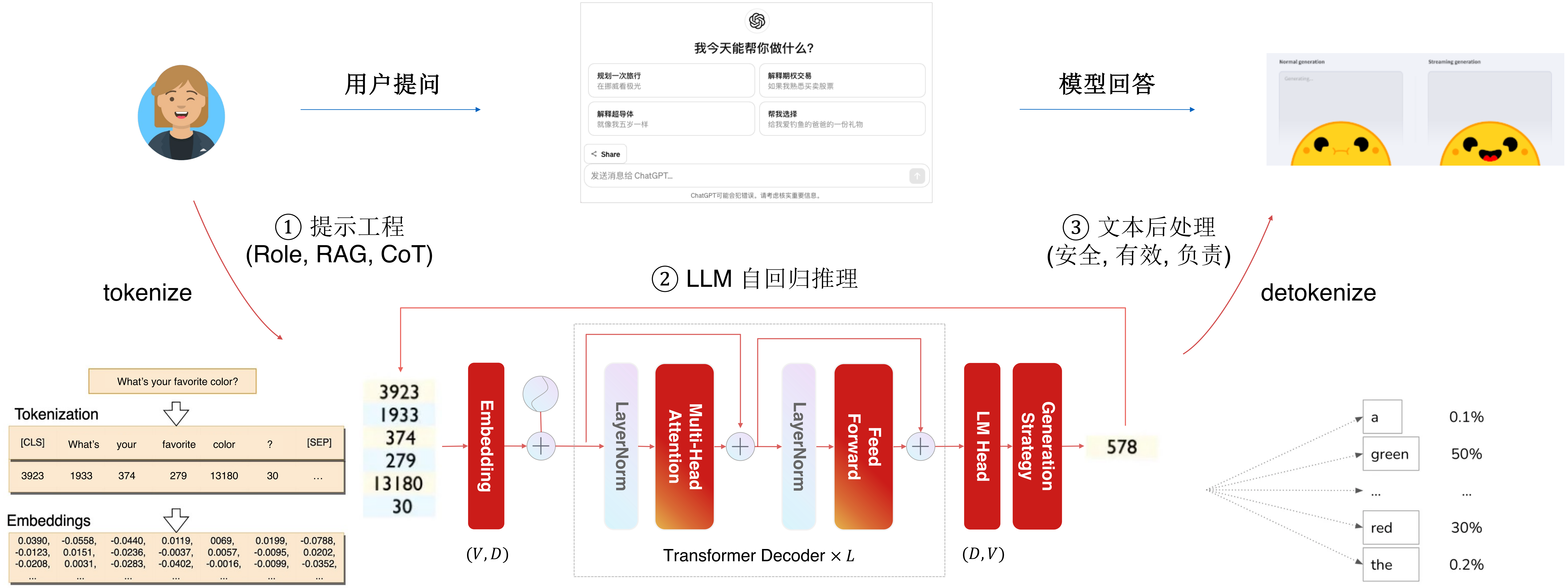
[1] OpenCompass Leaderboard. https://rank.opencompass.org.cn/home

[2] AlignBench, SafetyBench, AgentBench. https://llmbench.ai/align

[3] TensorRT-LLM Performance. https://github.com/NVIDIA/TensorRT-LLM/blob/main/docs/source/performance.md

目录

LLM 如何完成一次推理



ChatGPT, Claude, LLaMA, Mixtral, Yi, DBRX, Grok ...

ChatGLM, Erniebot, Qwen, Baichuan, DeepSeek, InternLM, UniGPT ...

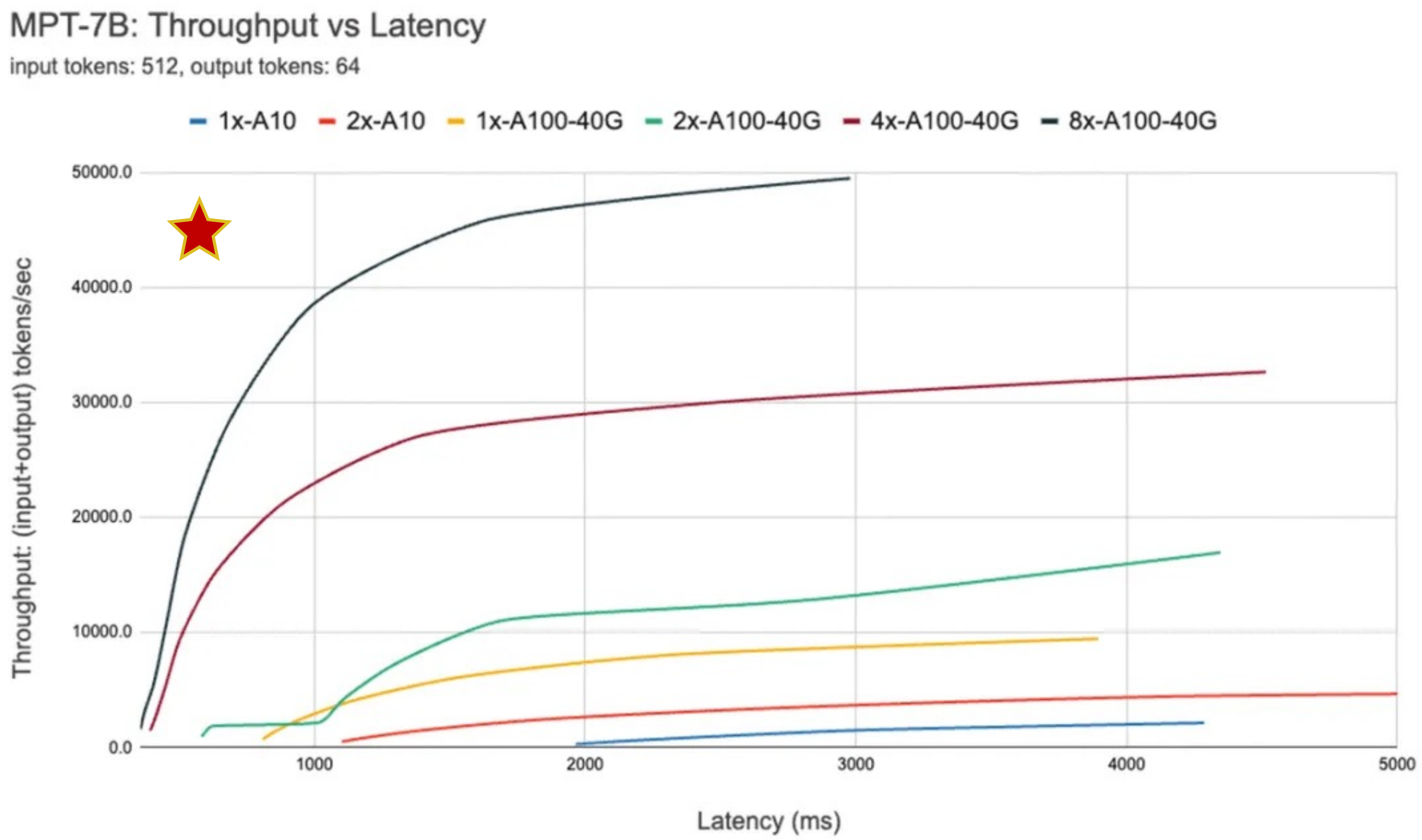
LLM 推理加速优化指标

- **用户关心的问题**
  - 模型生成质量能否满足我的要求？→ 推理加速要对齐模型原本精度（Accuracy **基本原则**）
  - 模型生成过程是否值得我的等待？→ 用户收到模型反馈不能等太久（Latency TTFT）
  - 模型生成速度能否跟上我的阅读？→ 模型每秒输出的字数要足够多（Latency TPOT）
- **工程师关心的问题**
  - 用户关心的问题
  - 在固定资源下能否服好更多用户？→ 追求**吞吐量**和**时延**的均衡（Throughput QPS）

[1] 语言大模型推理性能工程：最佳实践. https://mp.weixin.qq.com/s/mniKrBWkDE1tWWb2wQBDCA

LLM 推理加速优化指标

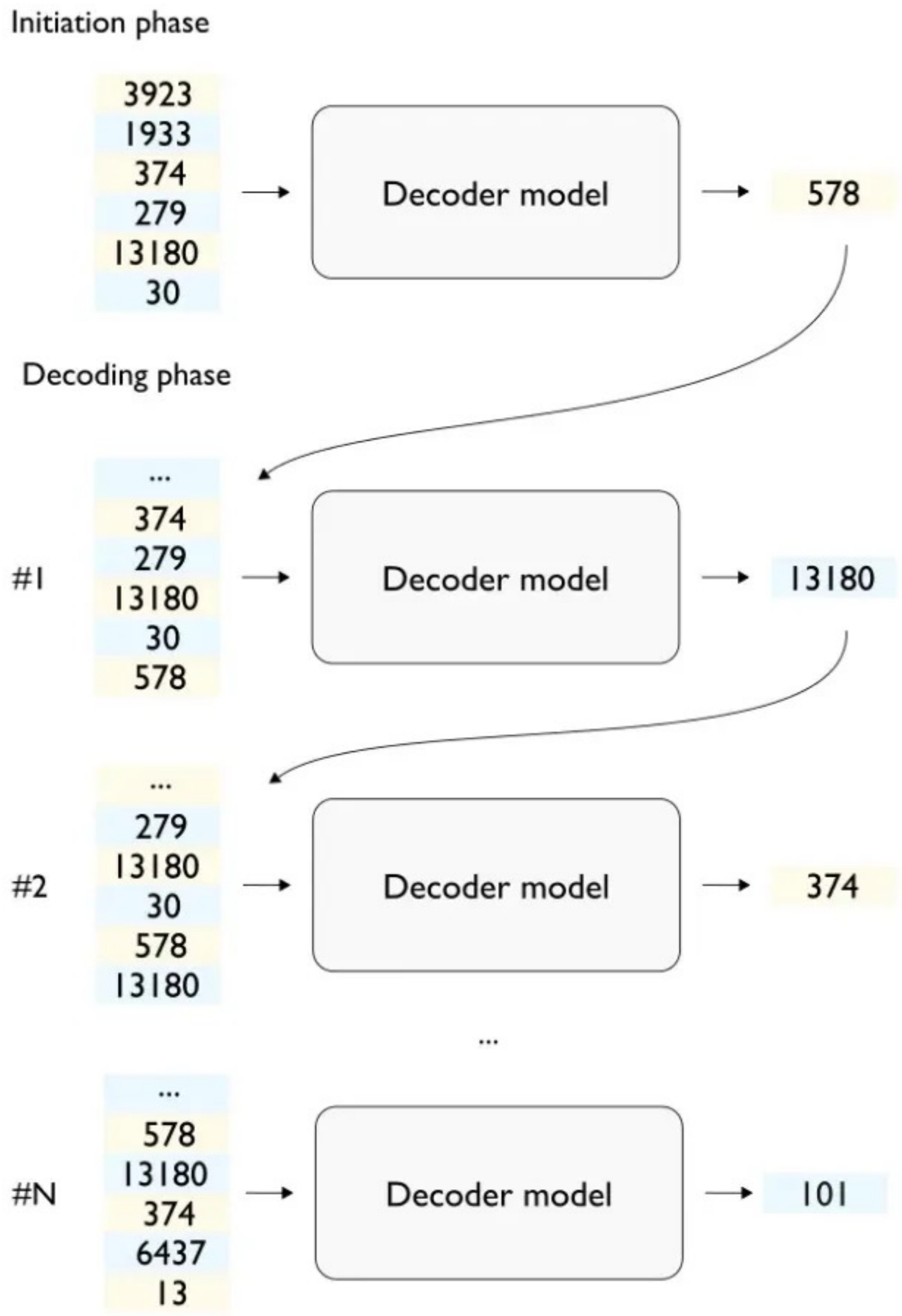
- Throughput ↑ vs. Latency ↓



Batchsize 逐渐从 1 增加到 256

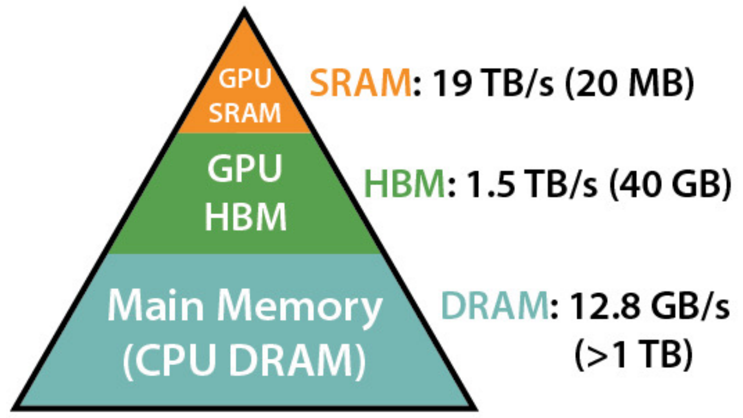
[1] 语言大模型推理性能工程：最佳实践. https://mp.weixin.qq.com/s/mniKrBWkDE1tWWb2wQBDCA

LLM 推理指标影响因素



Decoding 阶段，推理参数量为 $\boldsymbol{P}$ 的模型：

- 计算：$\sim 2 * P * B$ FLOPs (算力)
- 内存：$2 * P$ GB (FP16 模型)



A100-SXM-40G

Memory Hierarchy with Bandwidth & Memory Size

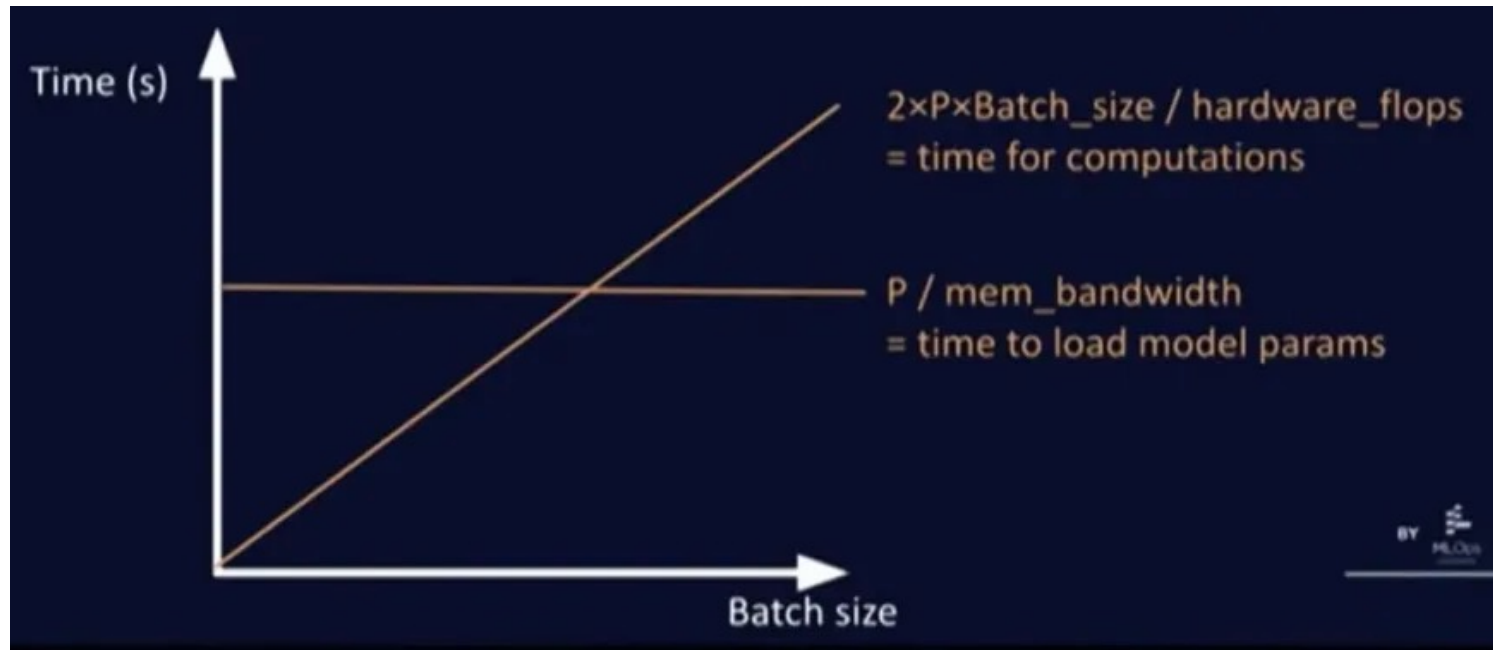
以 **A100-SXM-40G**，**LLaMA-7B** 模型为例：

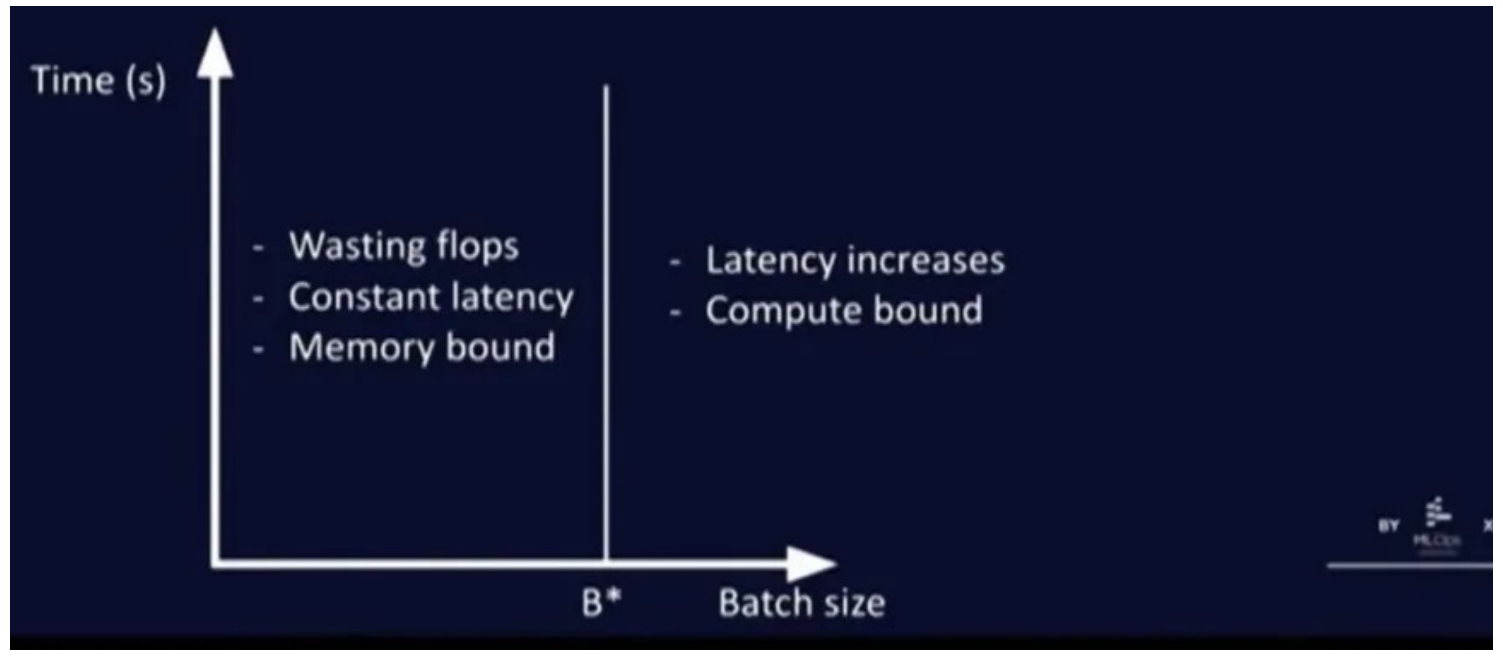
$$\frac{2 * 7}{1555} \gg \frac{2 * 7 * B * 10^9}{312 * 10^{12}}$$

内存搬运时间 ≫ 模型计算时间 (考虑 KV 缓存)

A100-SXM-40G: $B_{max} = 13 \ll 200$

H100-SXM-80G: $B_{max} = 33 \ll 590$





200

Peak Throughput

[1] Mistral AI：探索 LLM 推理的吞吐、时延及成本空间 https://www.youtube.com/watch?v=mYRqvB1_gRk&ab_channel=MLOps.community
[2] LLM推理入门指南①：文本生成的初始化与解码阶段. https://mp.weixin.qq.com/s/D9KPNI3CJ88l5_5vipjV3w
[3] Scaling Laws for Neural Language Models. http://arxiv.org/abs/2001.08361
[4] H100-SXM-80G, 显存带宽 3350 GB/s，FP16 算力 1979 TFLOPs. 显存带宽衡量了 GPU 单位时间能从显存读取的用于计算的数据量 (33 << 590)

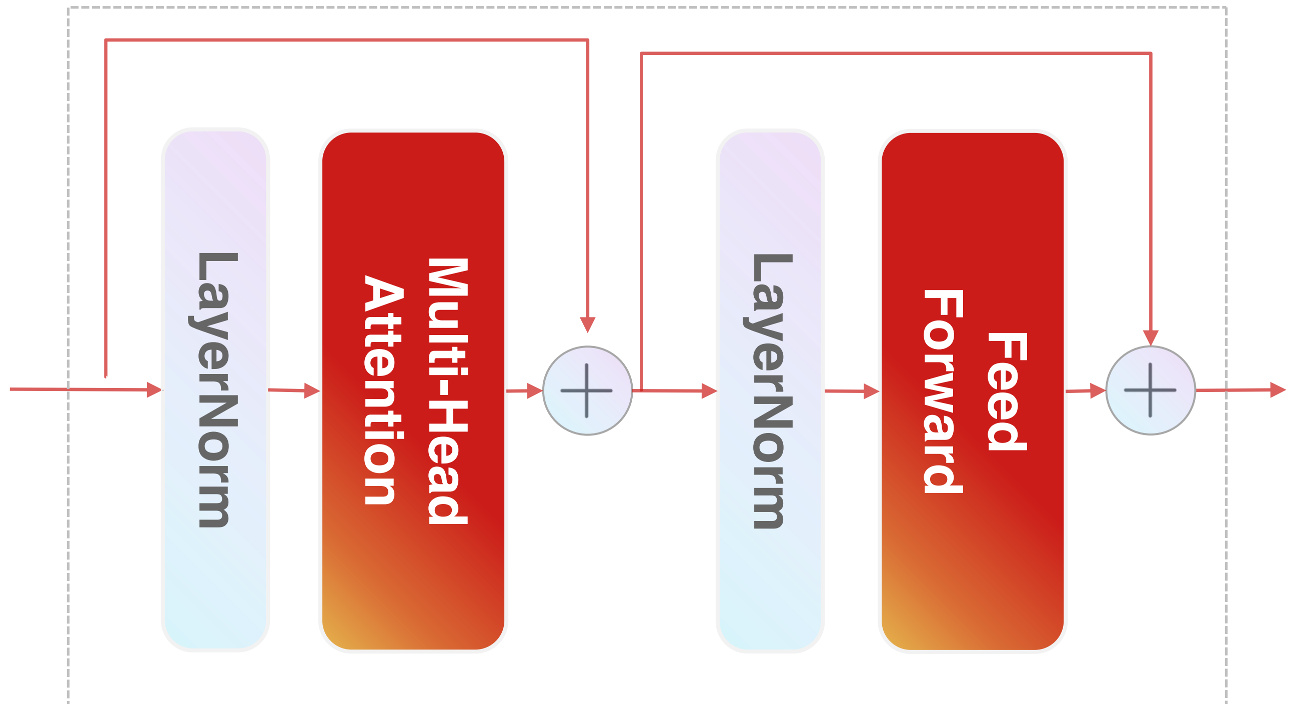
LLM 推理加速技术概览

【1】量化 (节约显存)

- Weight：GPTQ, AWQ
- Activation：KVQuant
- W&A：LLM.int8, SmoothQuant, OmniQuant
- …

【2】Attention & KV 缓存 (节约显存)

- Flash Attention/Decoding
- Paged/Chunk Attention
- StreamingLLM
- …



Transformer Decoder × L

$$Weight\ @\ Activation.\ \ nn.Linear$$

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

【3】批处理 (节约显存)

- Continuous Batching

【4】多卡并行 (增加计算单元)

- Megatron–TP

【5】稀疏化 (降低计算总量)

- SparseGPT, Wanda

【6】推测解码 (降低计算总量)

- SpecSampling
- Medusa, Hydra, EAGLE

【7】…

LLM 推理框架

- Python-based
  - Text Generation Interface @HuggingFace
  - vLLM @Berkeley
  - LightLLM @ModelTC, SenseTime
- 高效 CUDA-kernel
  - TensorRT-LLM, FasterTransformer, Triton-Inference-Server @NVIDIA
  - LMDeploy @InternLM, ShanghaiAILab
  - RTP-LLM @Alibaba
  - SiliconLLM @SiliconFlow
  - OmniForce @JD
- 本地/端侧部署
  - MLC-LLM @MLC-AI
  - PowerInfer @STJU
  - JittorLLMs @Tsinghua
  - Llama.cpp

[1] LLM 推理框架总结. https://mp.weixin.qq.com/s/P9rQWmQxqzyirt5v8aP0eA

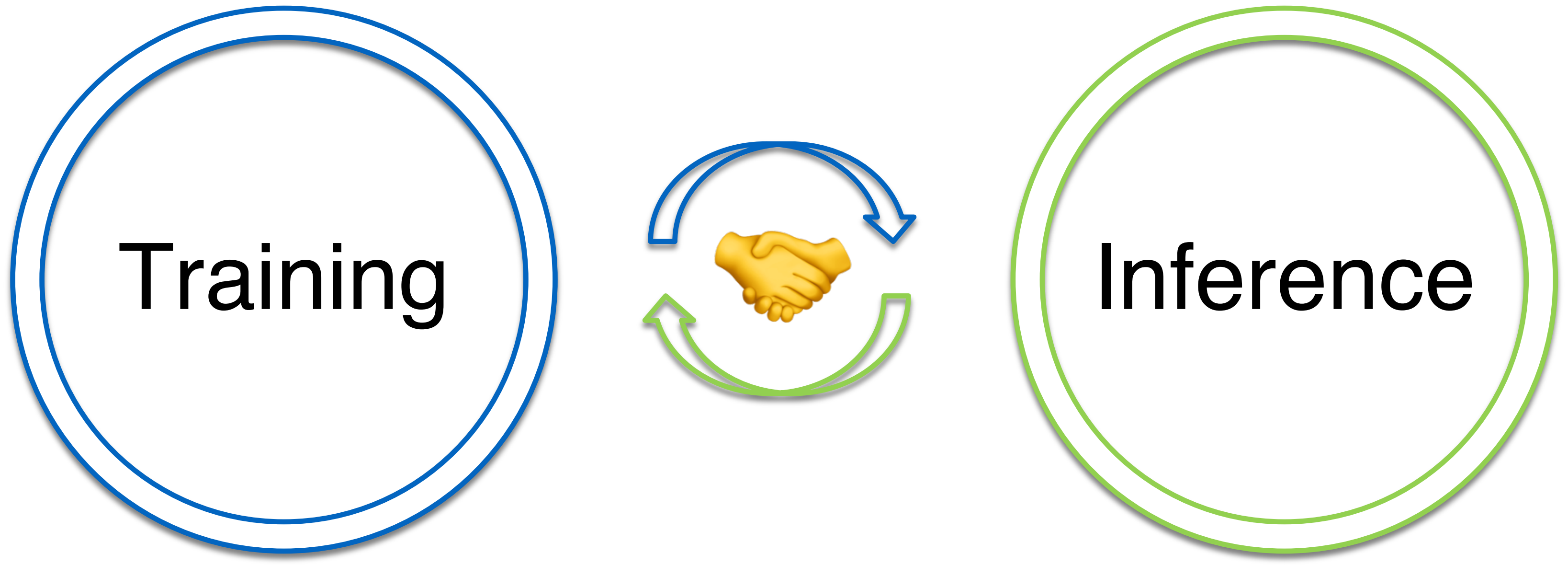
目录

1. LLM 推理加速技术概览
2. 推理 🤝 训练 协同演进
3. 推测解码与 Medusa
4. 未来展望

LLM 推理与训练协同演进

训推协同优化

- 量化
  - QLoRA, OneBit QAT
- Attention & KV 缓存
  - Grouped Query Attention
  - Sliding Window Attention
  - StreamingLLM
- 自适应模型
  - Early Exit, Mixture of Depth
- 推测解码
  - Medusa2

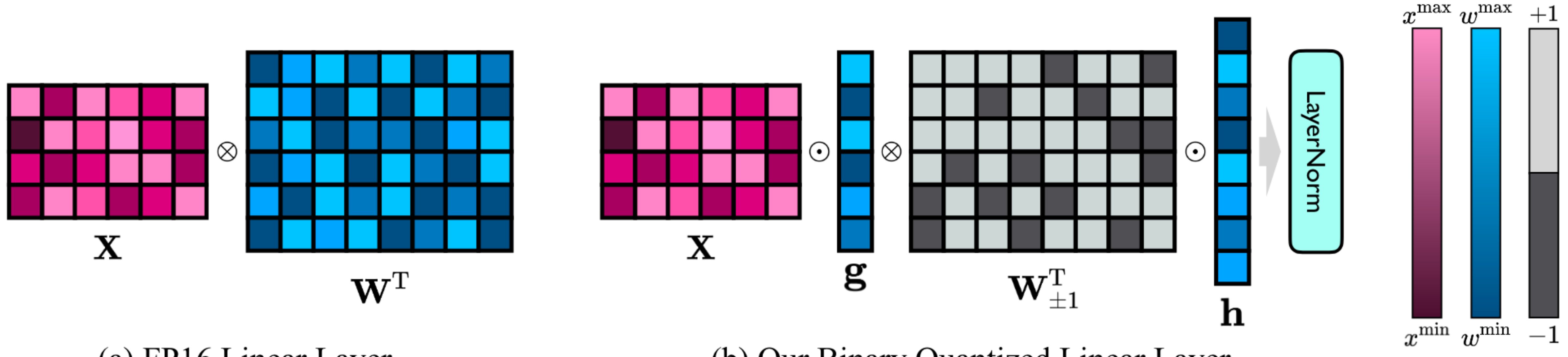


通用推理技术 (Post Training)

- 量化
  - GPTQ, AWQ, SmoothQuant
- Attention & KV 缓存
  - Flash Attention/Decoding
  - Paged/Chunk Attention
  - StreamingLLM
- 稀疏化
  - SparseGPT, Wanda
- 推测解码
  - SpecSampling
  - Medusa1, Hydra, EAGLE

LLM 训推协同优化：量化

- OneBit



(a) FP16 Linear Layer

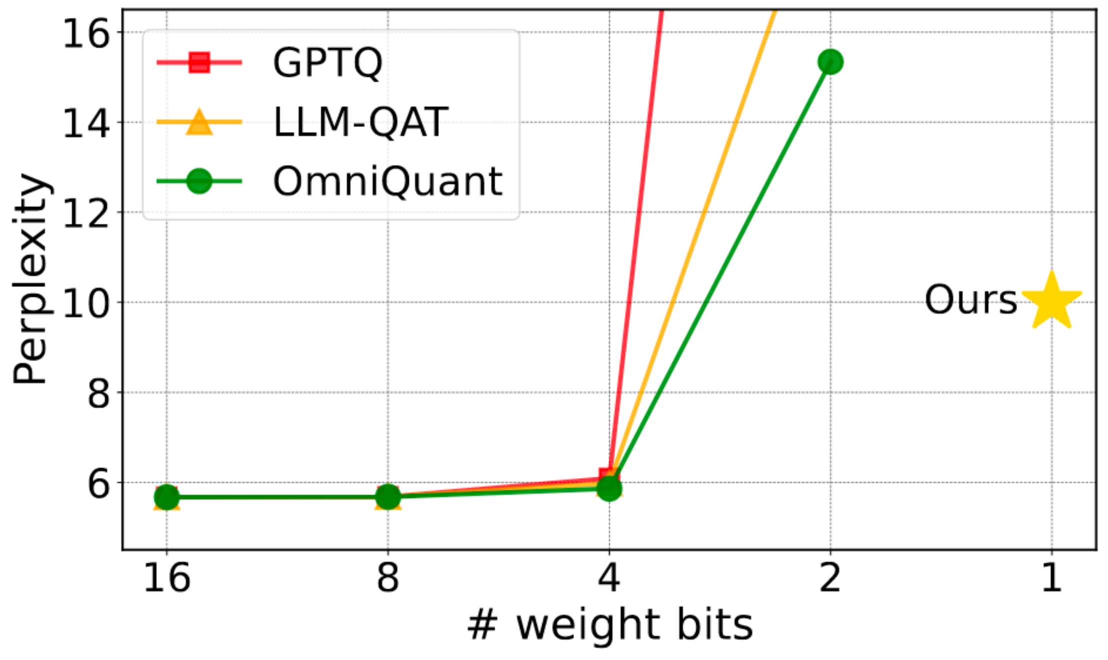
(b) Our Binary Quantized Linear Layer

$$\mathbf{X}\mathbf{W}^{\mathrm{T}} \approx \left[\left(\mathbf{X} \odot \mathbf{b}^{\mathrm{T}}\right) \mathbf{W}_{\text{sign}}^{\mathrm{T}}\right] \odot \mathbf{a}^{\mathrm{T}}.$$

$$\mathcal{L}_{\mathrm{KD}} = \mathcal{L}_{\mathrm{CE}} + \alpha \mathcal{L}_{\mathrm{MSE}}$$

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{c} P_c^{\mathcal{T}}\left(\mathbf{o}_i\right) \log P_c^{\mathcal{S}}\left(\mathbf{o}_i\right)$$

$$\mathcal{L}_{\mathrm{MSE}} = \sum_{i=1}^{n_s} \sum_{j=1}^{n_l} \left\| \frac{\mathbf{q}_{i,j}^{\mathcal{T}}}{\left\|\mathbf{q}_{i,j}^{\mathcal{T}}\right\|_2} - \frac{\mathbf{q}_{i,j}^{\mathcal{S}}}{\left\|\mathbf{q}_{i,j}^{\mathcal{S}}\right\|_2} \right\|_2^2$$

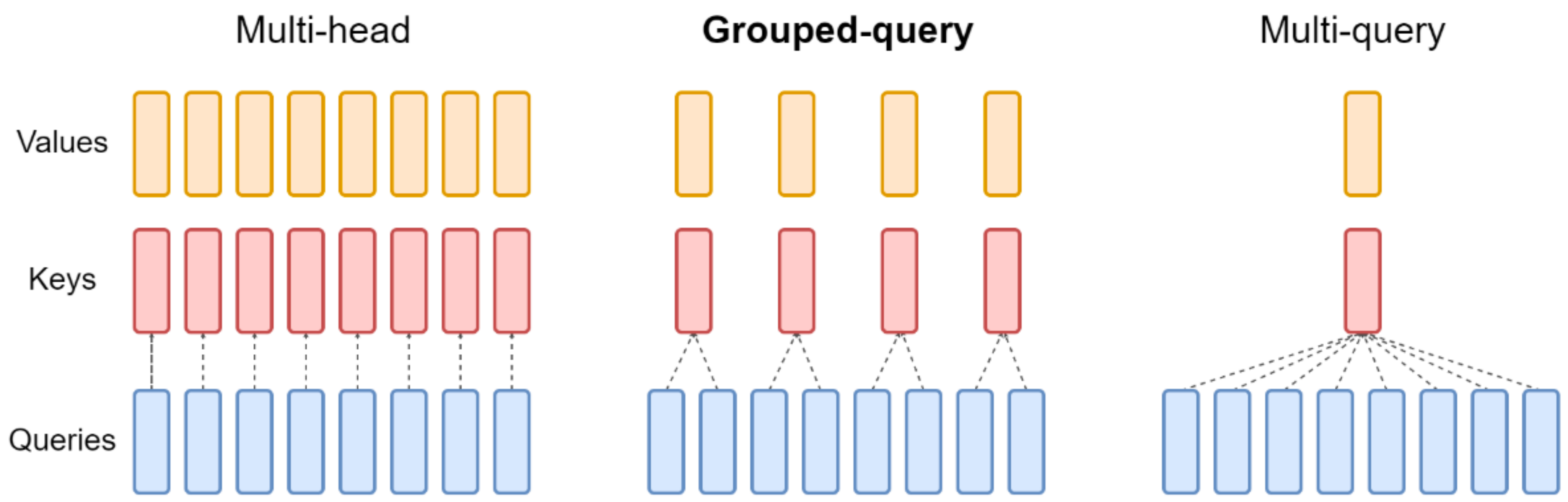


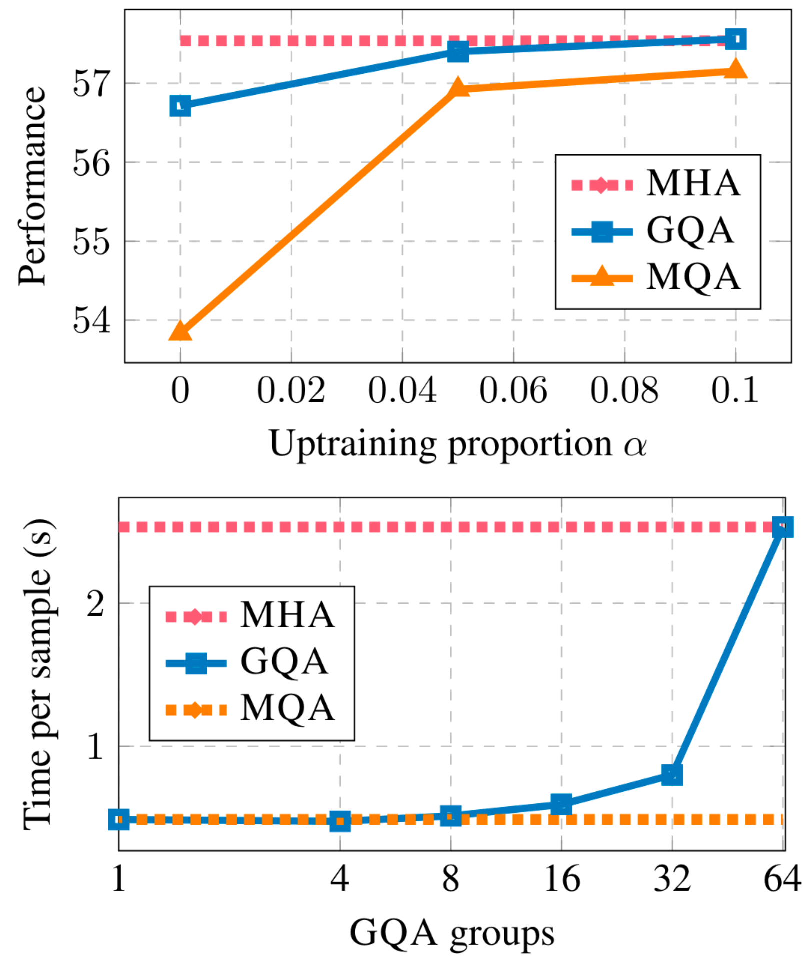
Models	Methods	Perplexity(↓)		Zero-shot Accuracy(↑)						
		Wiki2	C4	Winogrande	Hellaswag	PIQA	BoolQ	ARC-e	ARC-c	Avg.
OPT-1.3B	FP16	14.63	14.72	59.67	53.73	72.42	57.68	50.80	29.69	54.00
	GPTQ	9.5e3	3.8e3	49.33	25.57	52.07	39.60	26.68	23.63	36.15
	LLM-QAT	4.9e3	2.1e3	49.72	25.72	50.05	37.83	25.76	**25.09**	35.70
	OmniQuant	42.43	55.64	**51.85**	33.39	60.94	56.45	38.76	23.38	44.13
	OneBit	**25.42**	**22.95**	51.14	**34.26**	**62.57**	**59.45**	**41.25**	24.06	**45.46**
OPT-2.7B	FP16	12.47	13.17	60.93	60.59	74.81	60.28	54.34	31.31	57.04
	GPTQ	8.7e3	3.9e3	49.88	26.47	49.84	39.88	25.76	**26.02**	36.31
	LLM-QAT	3.7e3	1.4e3	52.09	25.47	49.29	37.83	24.92	25.60	35.87
	OmniQuant	30.25	41.31	51.62	**38.21**	62.19	54.25	40.82	24.74	45.31
	OneBit	**21.86**	**20.76**	**51.67**	38.18	**63.87**	**54.28**	**43.39**	24.40	**45.97**
LLaMA-7B	FP16	5.68	7.08	66.85	72.99	77.37	73.21	52.53	41.38	64.06
	GPTQ	1.9e3	7.8e2	49.41	25.63	49.95	43.79	25.84	27.47	37.02
	LLM-QAT	7.1e2	3.0e2	51.78	24.76	50.87	37.83	26.26	25.51	36.17
	OmniQuant	15.34	26.21	52.96	43.68	62.79	58.69	41.54	29.35	48.17
	OneBit	**10.38**	**11.56**	**60.30**	**50.73**	**67.46**	**62.51**	**41.71**	**29.61**	**52.05**
LLaMA-13B	FP16	5.09	6.61	70.17	76.24	79.05	68.47	59.85	44.54	66.39
	GPTQ	3.2e3	9.9e2	50.67	25.27	50.00	42.39	26.14	27.39	36.98
	LLM-QAT	1.8e3	1.2e3	51.62	25.40	50.33	37.83	27.02	26.87	36.51
	OmniQuant	13.43	19.33	53.83	54.16	68.99	62.20	**45.50**	30.38	52.51
	OneBit	**9.18**	**10.25**	**62.90**	**56.78**	**70.67**	**64.16**	44.53	**32.00**	**55.17**

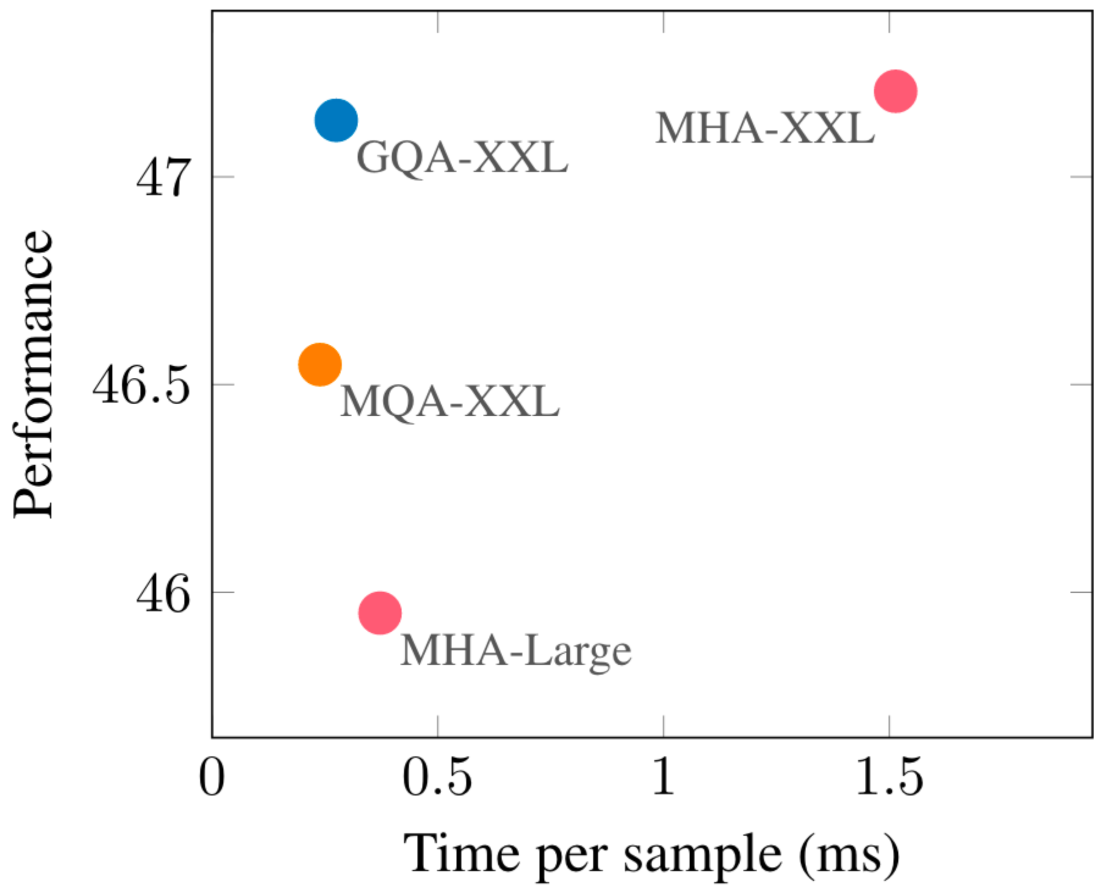
[1] OneBit: Towards Extremely Low-bit Large Language Models. https://arxiv.org/pdf/2402.11295

LLM 训推协同优化：Attention

- Grouped Query Attention 降低 KVCache 维度

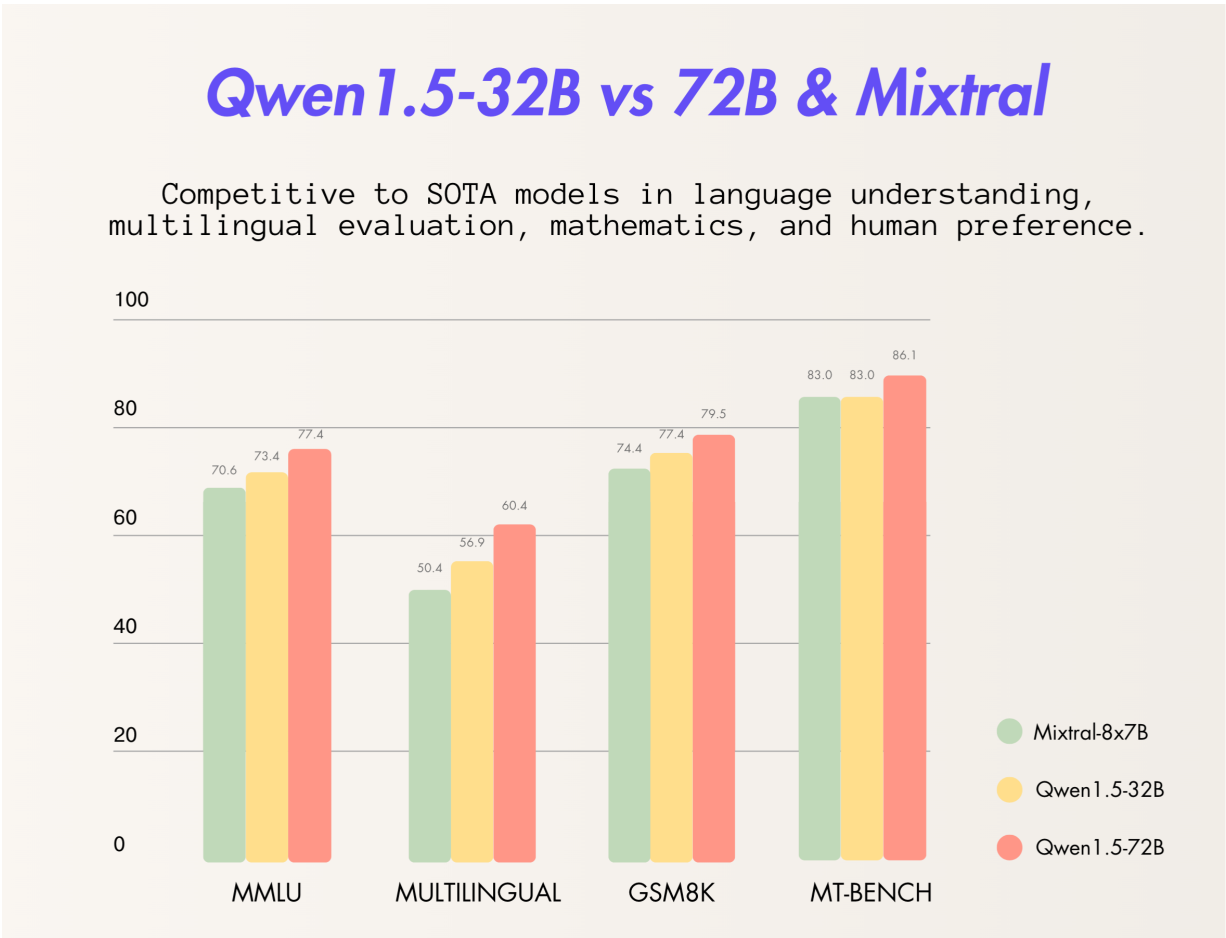






GQA in LLMs

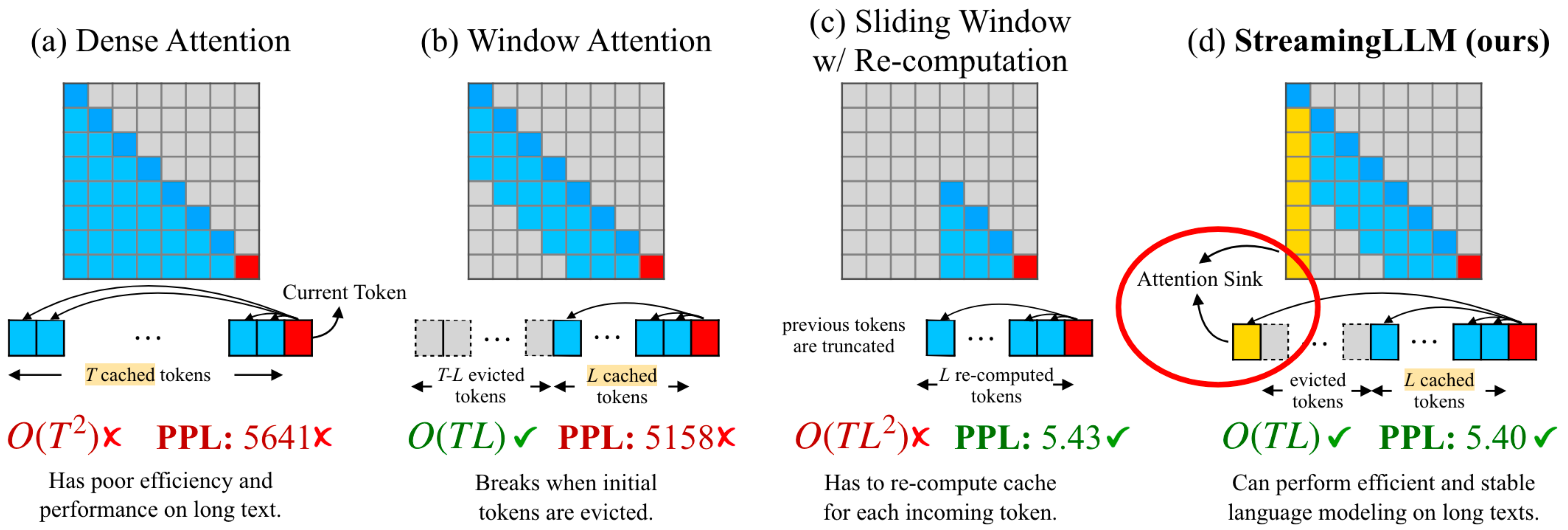
	Model	n_layers	n_heads	n_KV_heads	d_head	d_model	Attention
2	Llama-2-7B	32	32	32	128	4096	MHA
3	Llama-2-13B	40	40	40	128	5120	MHA
4	Llama-2-70B	80	64	8	128	8192	GQA
5	Falcon-7B	32	71	1	64	4544	MQA
6	Falcon-40B	60	128	8	64	8192	GQA
7	Falcon-180B	80	232	8	64	14848	GQA
8	Mistral-7B	32	32	8	128	4096	GQA
9	PaLM-8B	32	16	1	256	4096	MQA
10	PaLM-62B	64	32	1	256	8192	MQA
11	PaLM-540B	118	48	1	384	18432	MQA

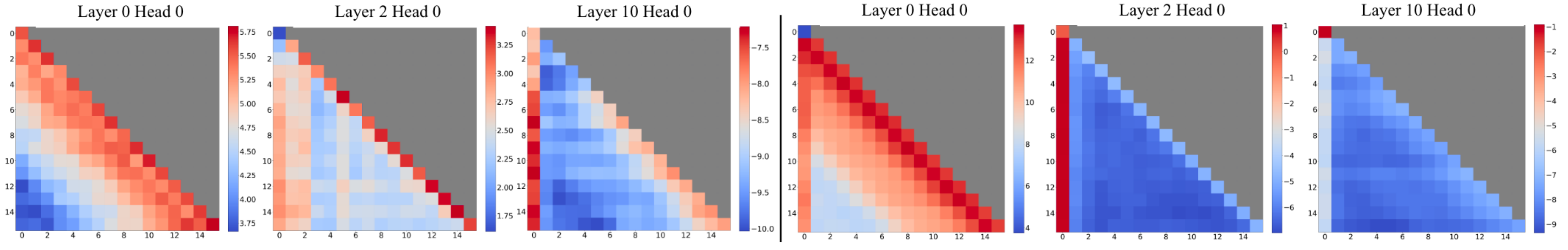


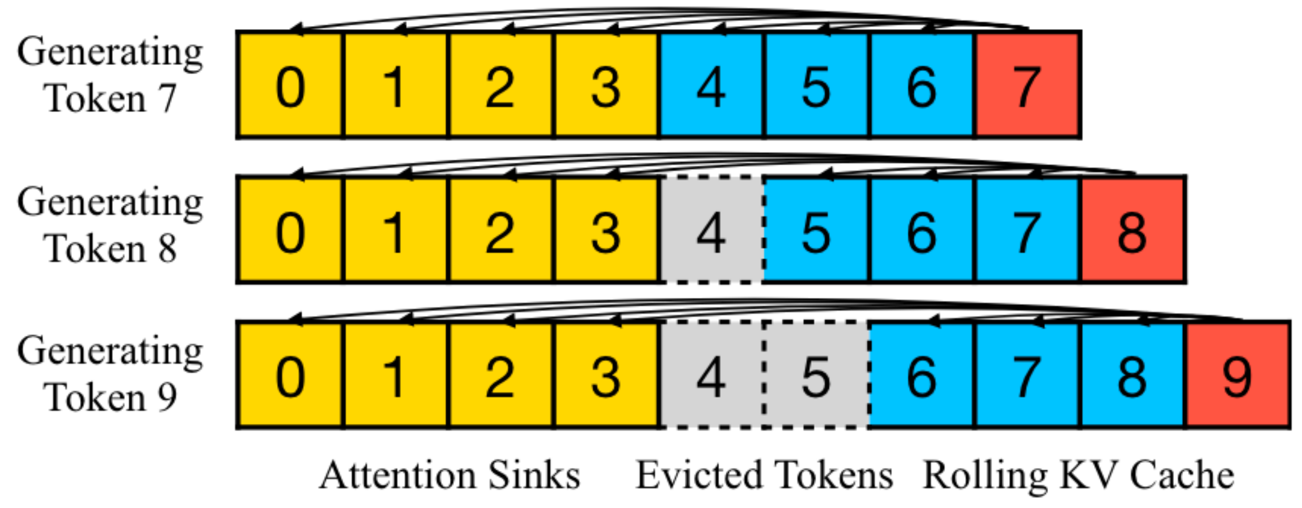
[1] GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. http://arxiv.org/abs/2305.13245
[2] Qwen1.5-32B：Qwen1.5语言模型系列的最后一块拼图. https://qwenlm.github.io/zh/blog/qwen1.5-32b/
[3] LLM推理入门指南②：深入解析KV缓存. https://mp.weixin.qq.com/s/WxbMFoSrKl0xqsUkzPLJHw

LLM 训推协同优化：Attention

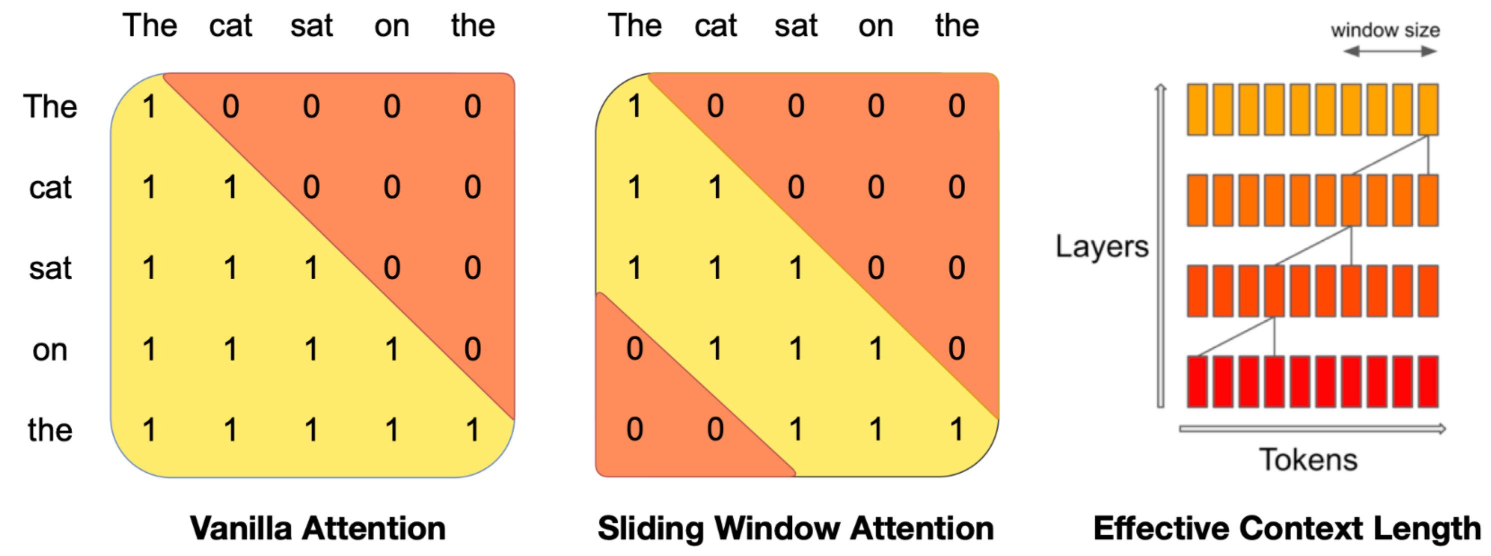
- Sliding Window Attention, StreamingLLM 降低 KVCache 长度







The KV cache of StreamingLLM.

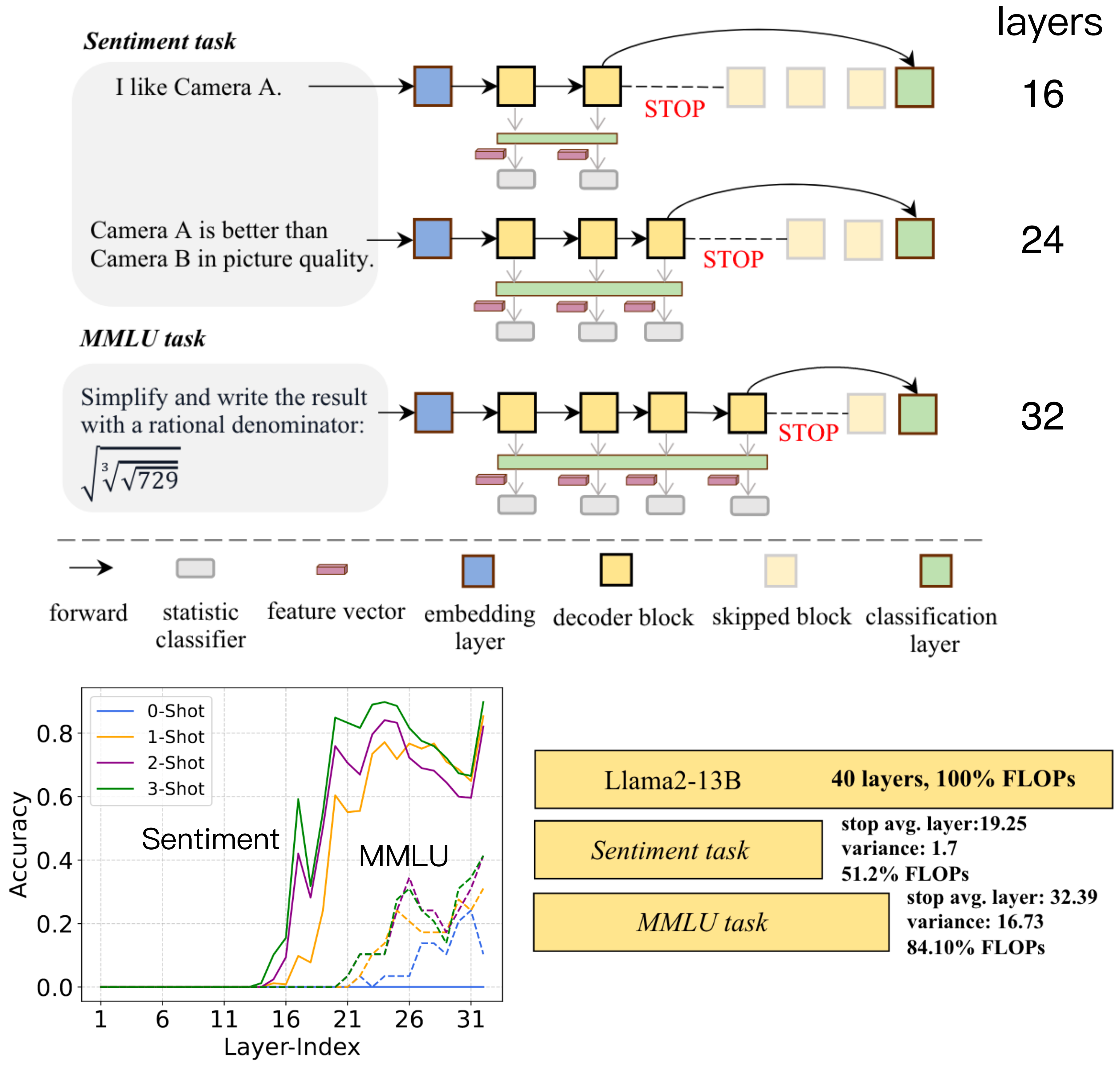


Sliding Window Attention in Mistral-7B.

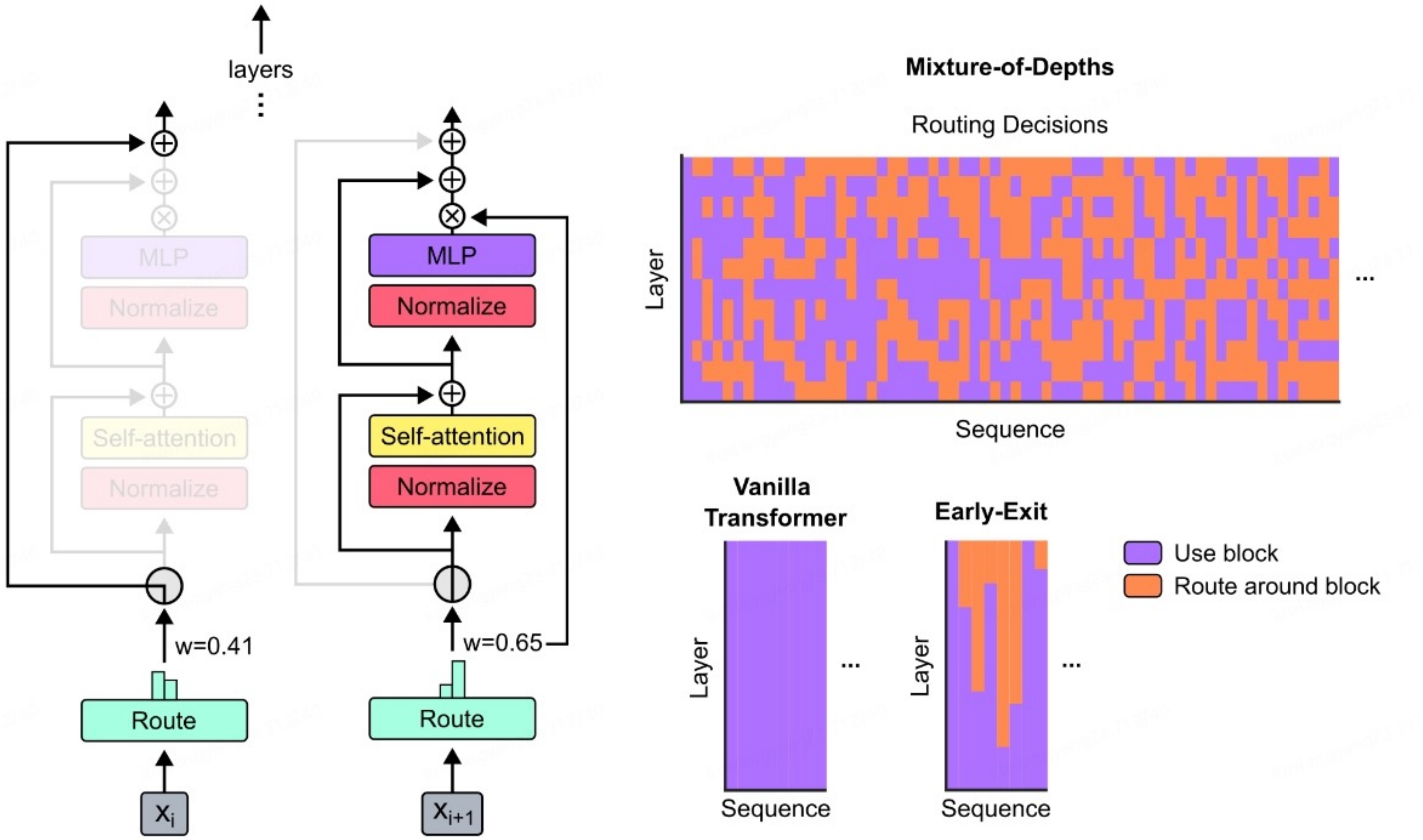
[1] StreamingLLM-Efficient Streaming Language Models with Attention Sinks. http://arxiv.org/abs/2309.17453
[2] Mistral 7B. http://arxiv.org/abs/2310.06825

LLM 训推协同优化：自适应模型

- Early Exit in AdaInfer



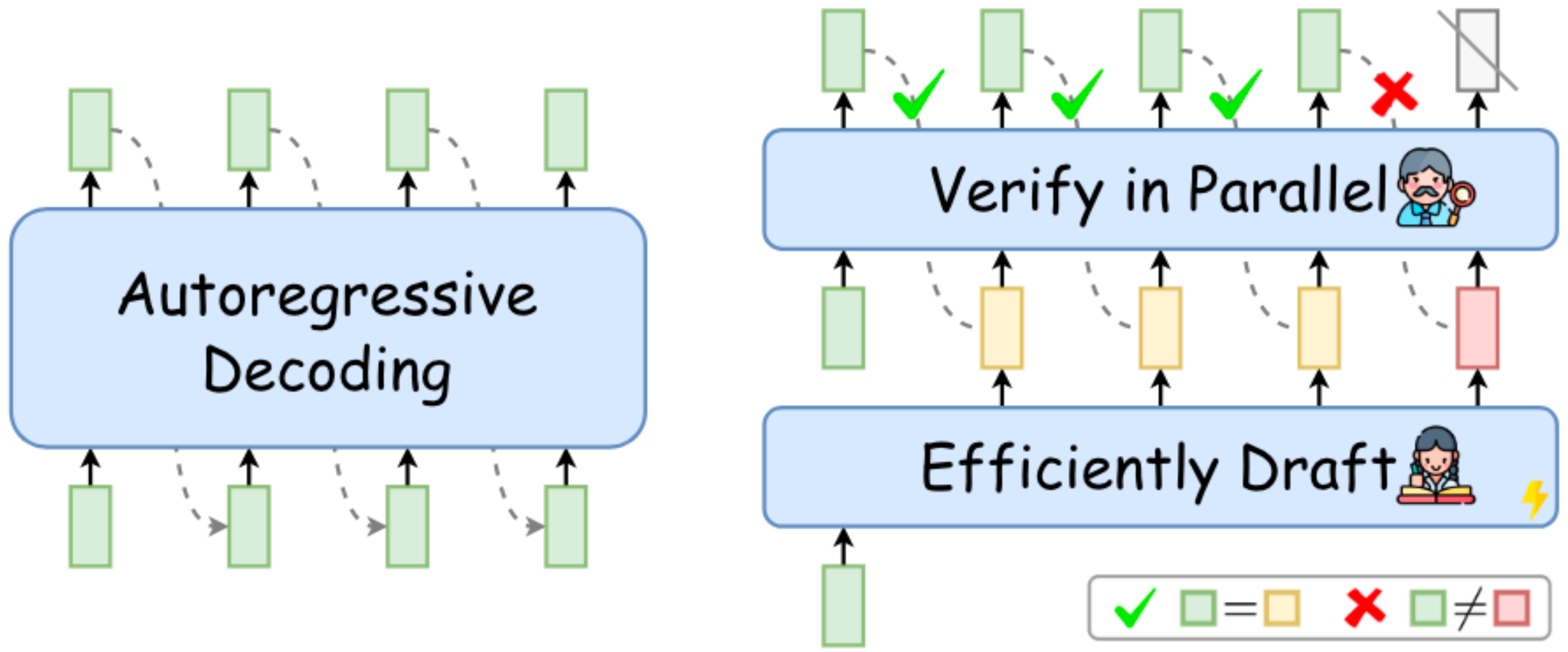
- Mixture of Depth (MoD)

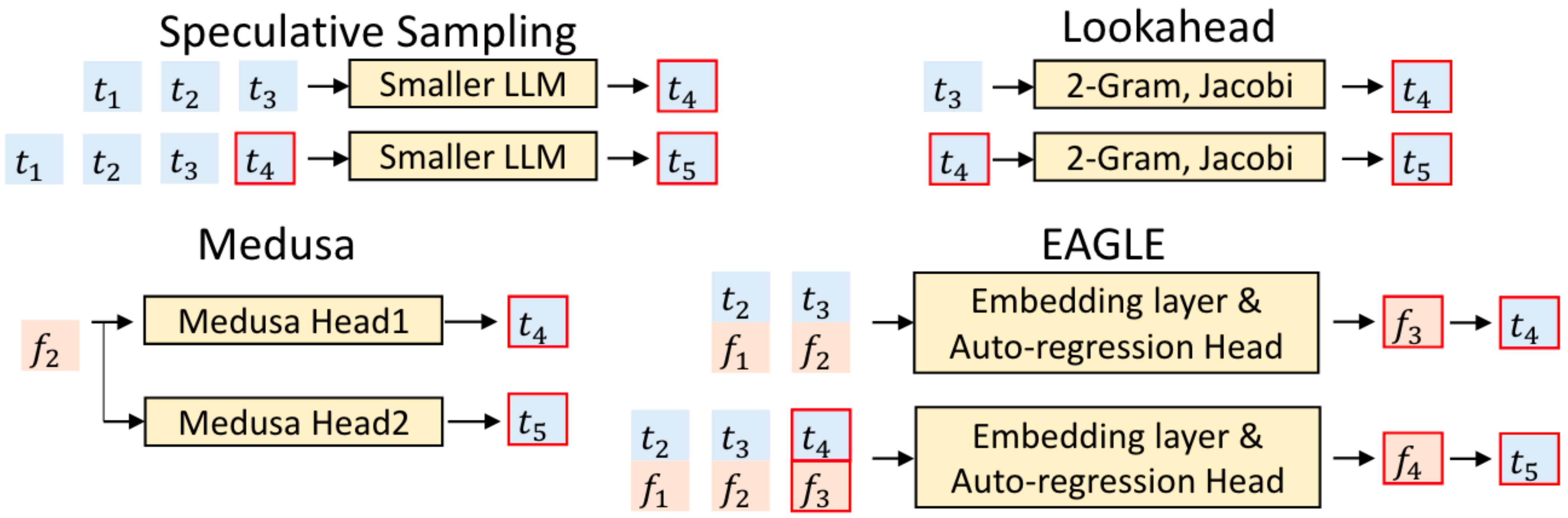


[1] Not all Layers of LLMs are Necessary during Inference. http://arxiv.org/abs/2403.02181
[2] Mixture-of-Depths: Dynamically allocating compute in transformer-based language models. http://arxiv.org/abs/2404.02258

目录

何为推测解码?





推测解码的三步：

(1) 生成 candidates.

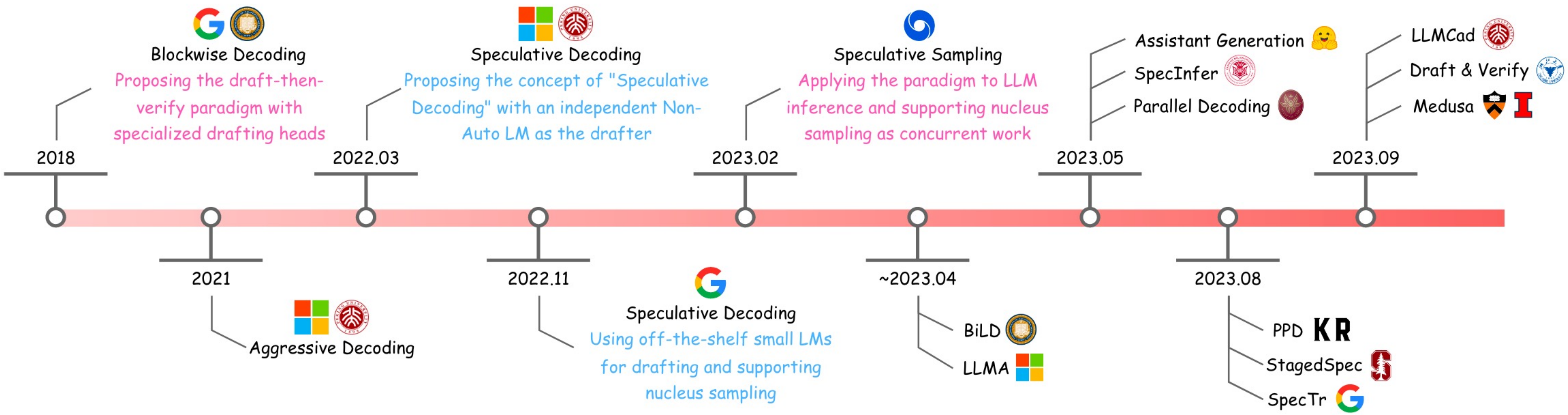
(2) 验证 candidates.

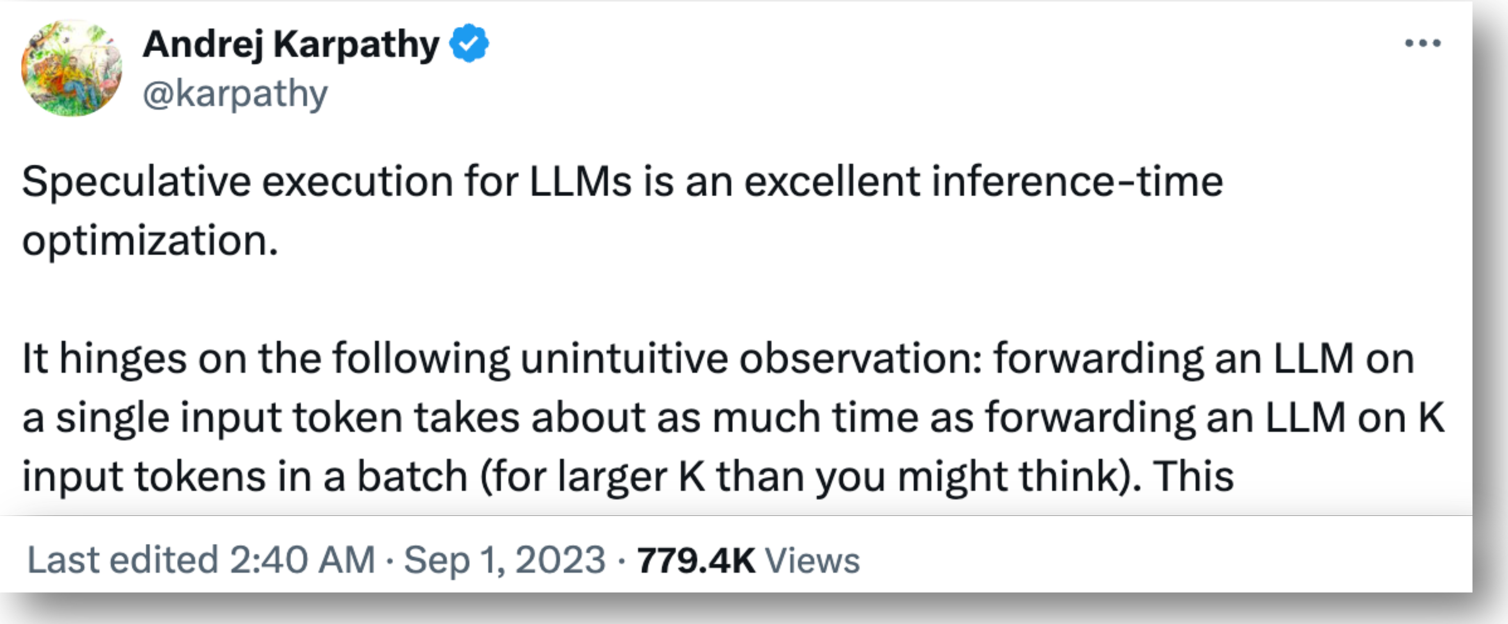
(3) 接收 candidates.

Methods	VERIFY $(\tilde{x}_i, p_i, q_i)$	CORRECT (p_c, q_c)
Greedy Decoding	$\tilde{x}_i = \arg\max q_i$	$x_{t+c} \leftarrow \arg\max q_c$
Nucleus Sampling	$r < \min\left(1, \frac{q_i(\tilde{x}_i)}{p_i(\tilde{x}_i)}\right), r \sim U[0,1]$	$x_{t+c} \sim \text{norm}(\max(0, q_c - p_c))$

[1] Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding. http://arxiv.org/abs/2401.07851
[2] EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty. http://arxiv.org/abs/2401.15077

推测解码的发展



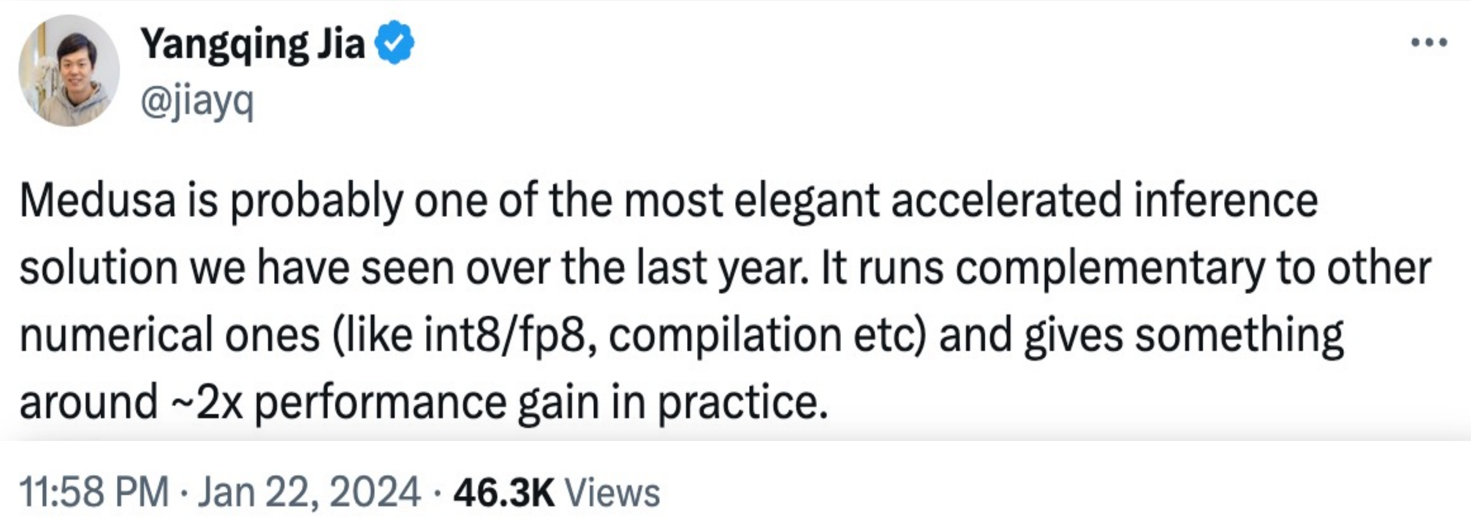


Andrej Karpathy @karpathy

Speculative execution for LLMs is an excellent inference-time optimization.

It hinges on the following unintuitive observation: forwarding an LLM on a single input token takes about as much time as forwarding an LLM on K input tokens in a batch (for larger K than you might think). This

Last edited 2:40 AM · Sep 1, 2023 · **779.4K** Views



Yangqing Jia @jiayq

Medusa is probably one of the most elegant accelerated inference solution we have seen over the last year. It runs complementary to other numerical ones (like int8/fp8, compilation etc) and gives something around ~2x performance gain in practice.

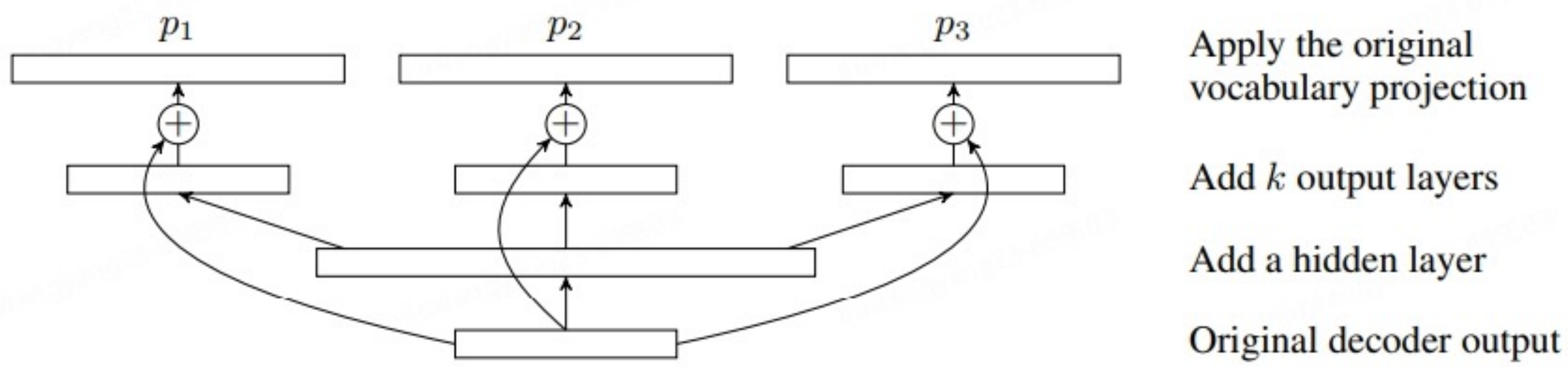
11:58 PM · Jan 22, 2024 · **46.3K** Views

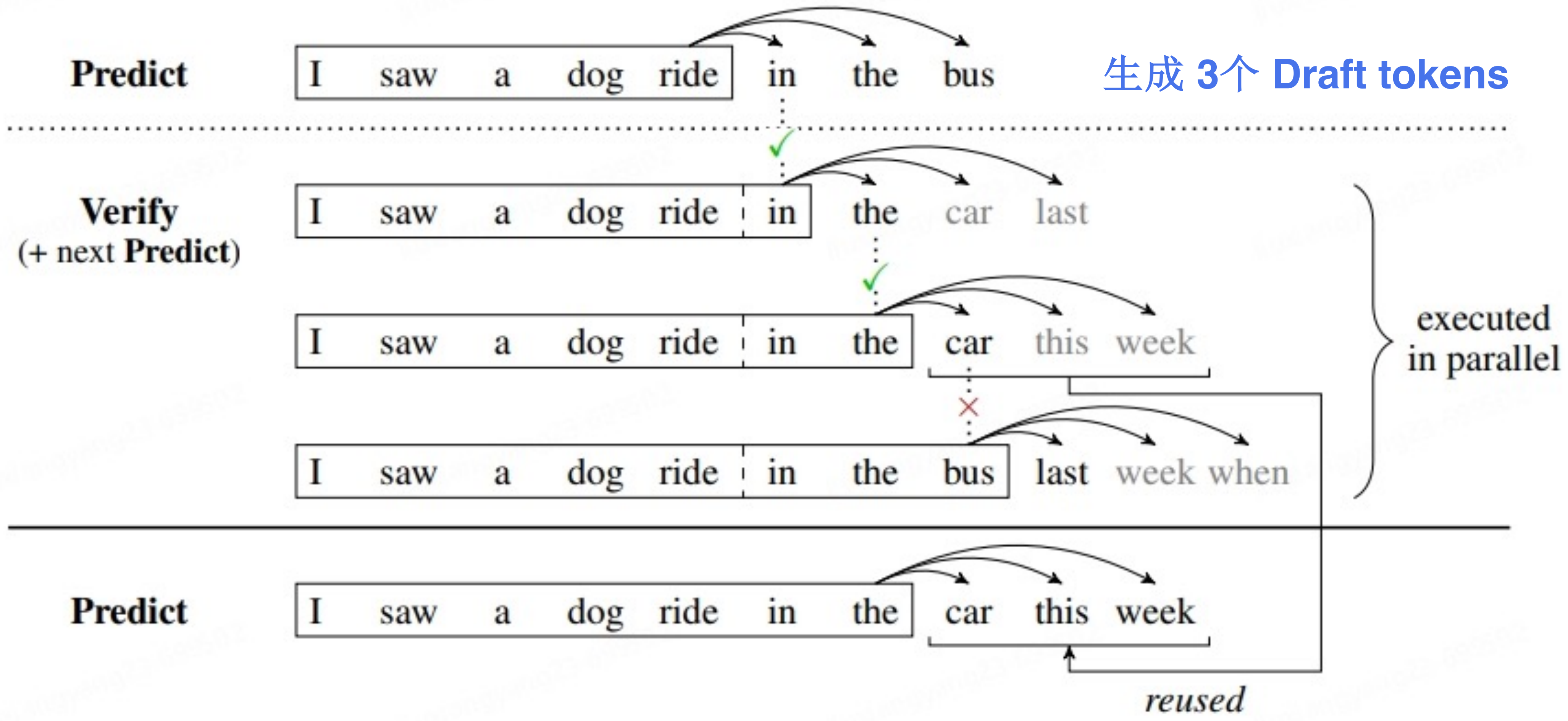
[1] Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding. http://arxiv.org/abs/2401.07851
[2] Andrej Karpathy comments Speculative Decoding. https://twitter.com/karpathy/status/1697318534555336961
[3] Yangqing Jia comments Medusa. https://twitter.com/jiayq/status/1749461664393810350

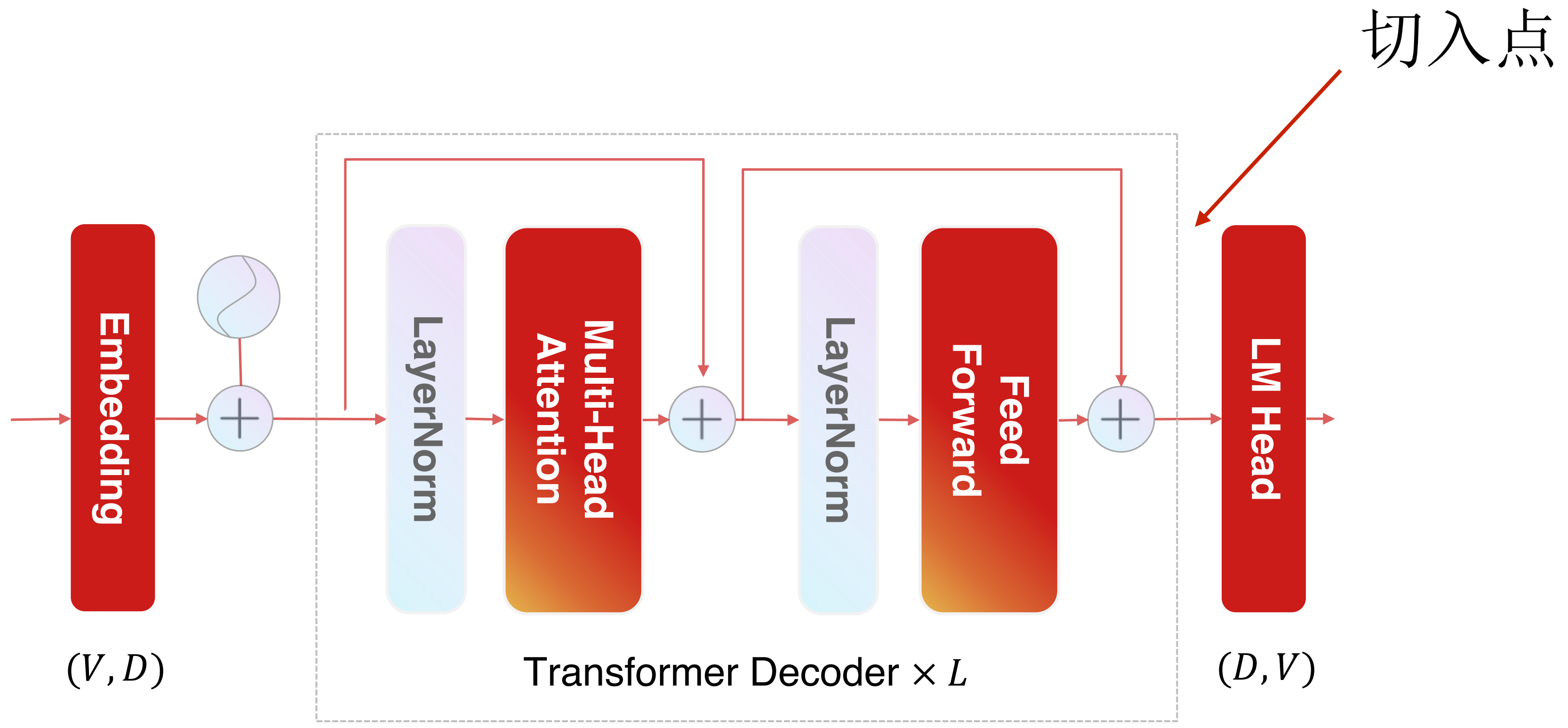
Blockwise Parallel Decoding

A continuation of k draft tokens

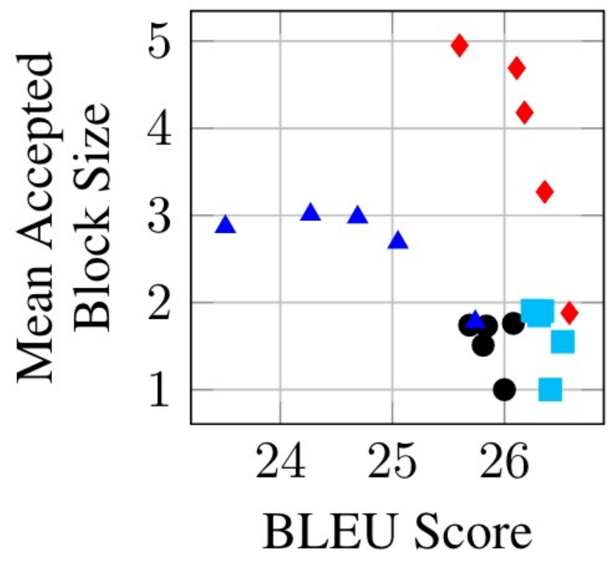
在大模型 Decoders 末尾，增加 1个多 LM_head 结构，生成 candidates. 生成长度为 m 序列的理想推理次数：$\frac{m}{k}+1$







k	Regular ●	Distillation ■	Fine Tuning ▲	Both ♦
1	26.00 / 1.00	26.41 / 1.00		
2	25.81 / 1.51	26.52 / 1.55	25.74 / 1.78	26.58 / 1.88
4	25.84 / 1.73	26.31 / 1.85	25.05 / 2.69	26.36 / 3.27
6	26.08 / 1.76	26.26 / 1.90	24.69 / 2.98	26.18 / 4.18
8	25.82 / 1.76	26.25 / 1.91	24.27 / 3.01	26.11 / 4.69
10	25.69 / 1.74	26.34 / 1.90	23.51 / 2.87	25.60 / 4.95




```
(0): ResBlock(
  (linear): Linear(in_features=4096, out_features=4096, bias=True)
  (act): SiLU()
) # m_t = r_t + h_t
```

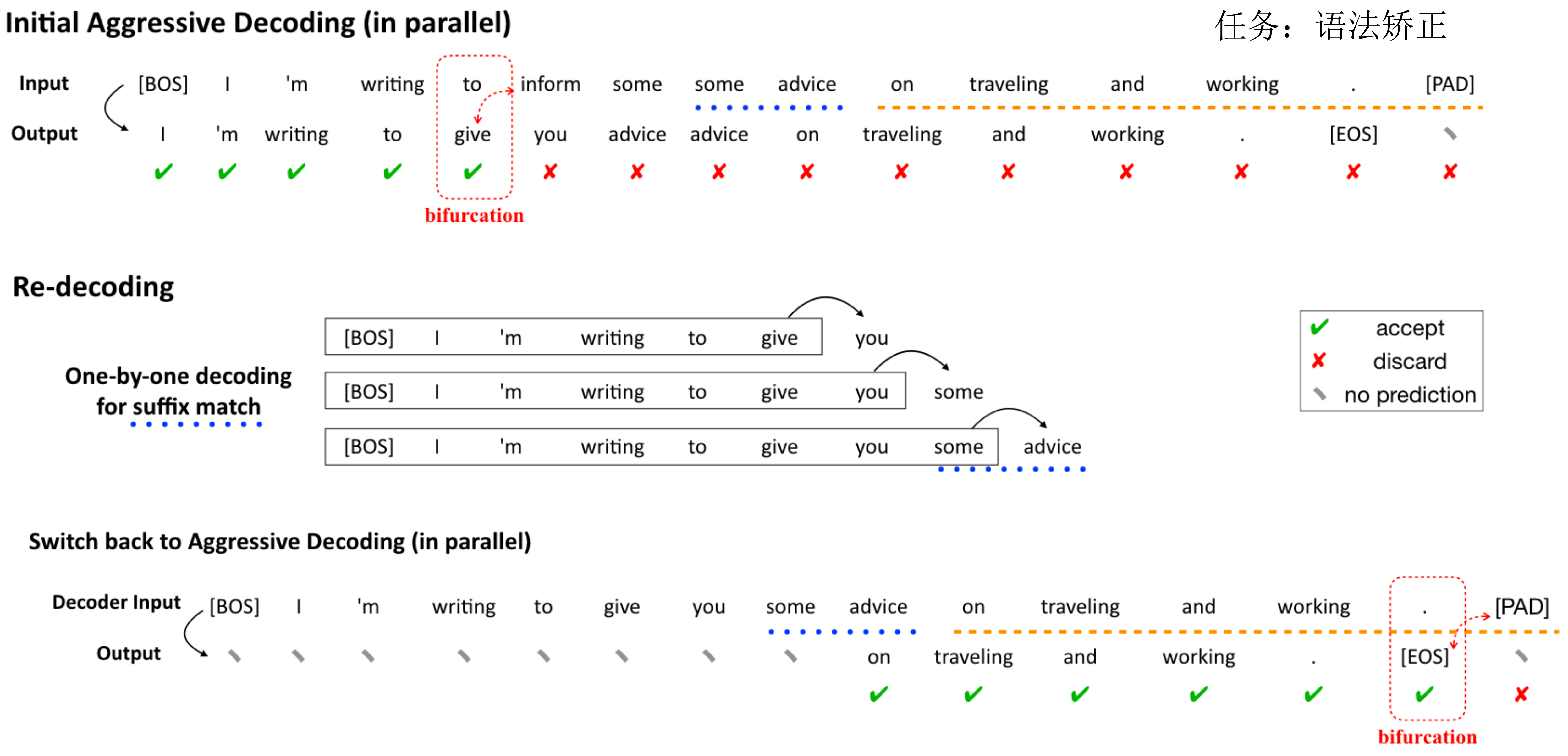
机器翻译 BLEU Score / 接收率

[1] Blockwise Parallel Decoding for Deep Autoregressive Models. https://arxiv.org/abs/1811.03115

Aggressive Decoding

考虑了有代表性的任务场景，并提出了使用分离式 Draft model 生成 candidates 的方式。

Input-guided Aggressive Decoding

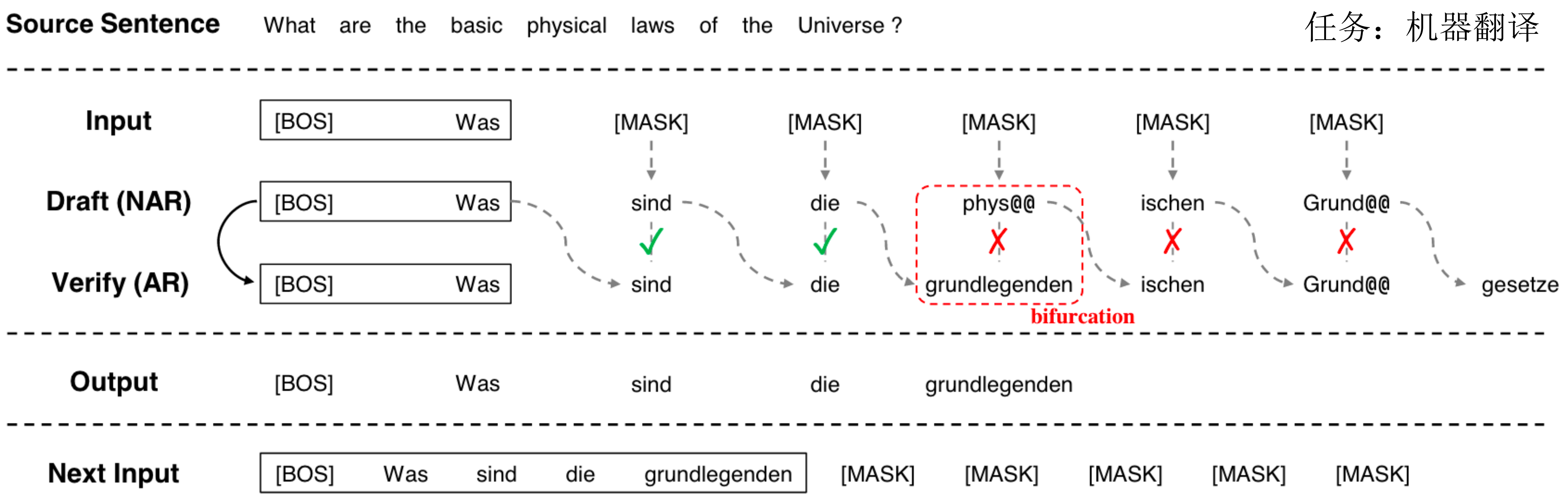


$$o_{j+1}^{*} = \arg\max_{o_{j+1}} \log P(o_{j+1} \mid \boldsymbol{o}_{\leq j}, \boldsymbol{x}; \boldsymbol{\Phi})$$ Output Token (AR)

$$= \arg\max_{o_{j+1}} \log P(o_{j+1} \mid \hat{\boldsymbol{o}}_{\leq j}, \boldsymbol{x}; \boldsymbol{\Phi})$$ Draft Token

$$= \arg\max_{o_{j+1}} \log P(o_{j+1} \mid \boldsymbol{x}_{\leq j}, \boldsymbol{x}; \boldsymbol{\Phi})$$ Input Token

Generalized Aggressive Decoding



$$\widetilde{\boldsymbol{o}}_{j+1\cdots j+k} = \arg\max_{\widetilde{o}_{j+1\cdots j+k}} \sum_{i=1}^{k} \log P\left(\widetilde{o}_{j+i} \mid \boldsymbol{o}_{\leq j}, \boldsymbol{x}; \boldsymbol{\Phi}_{\text{NAR}}\right)$$

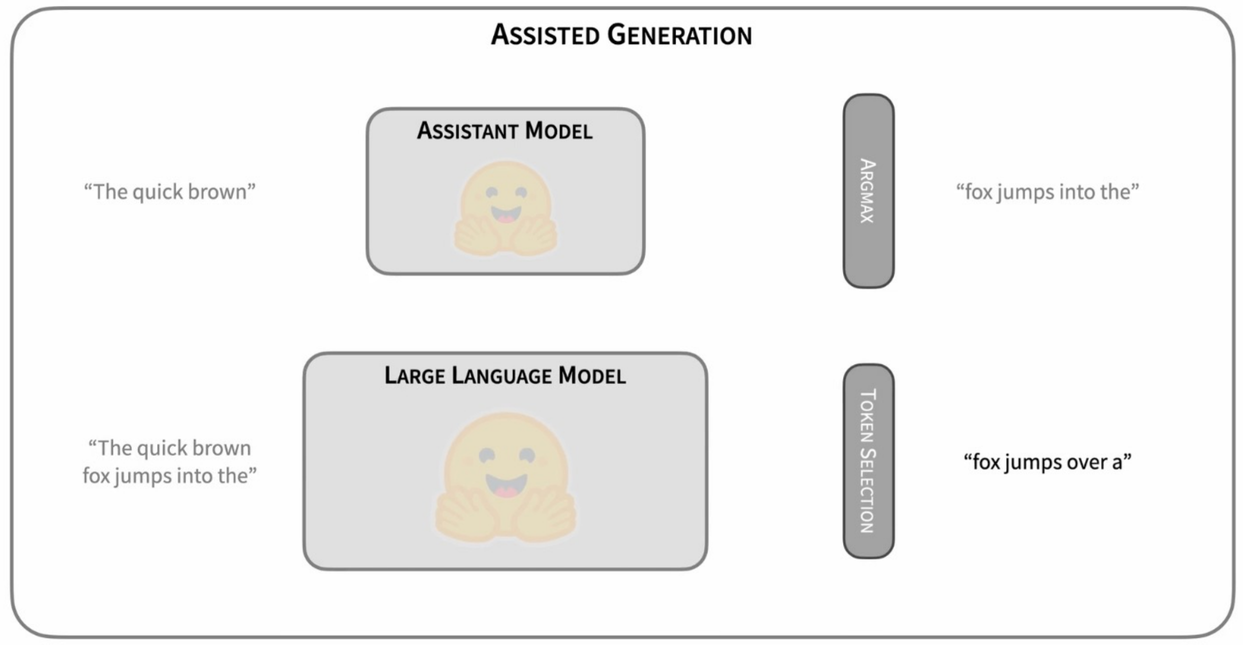
[1] Lossless Acceleration for Seq2seq Generation with Aggressive Decoding. http://arxiv.org/abs/2205.10350

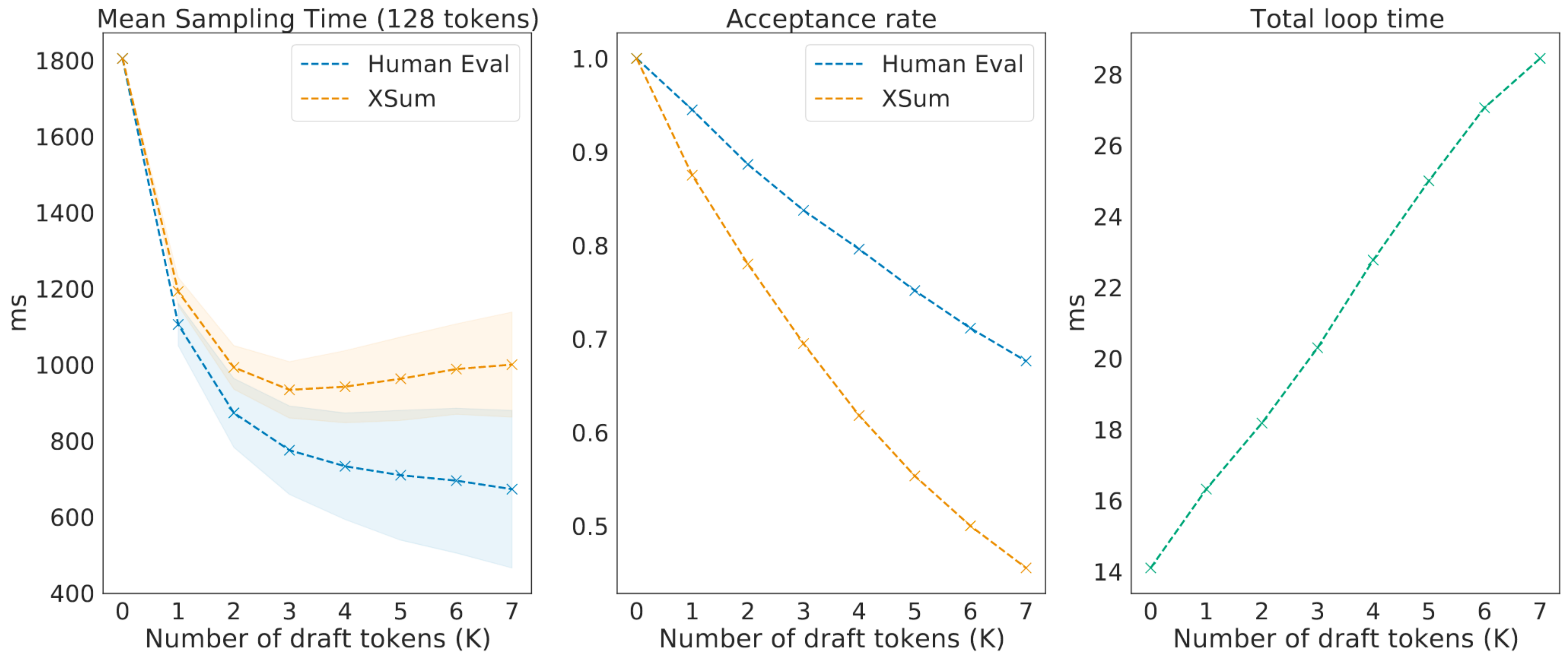
Speculative Sampling

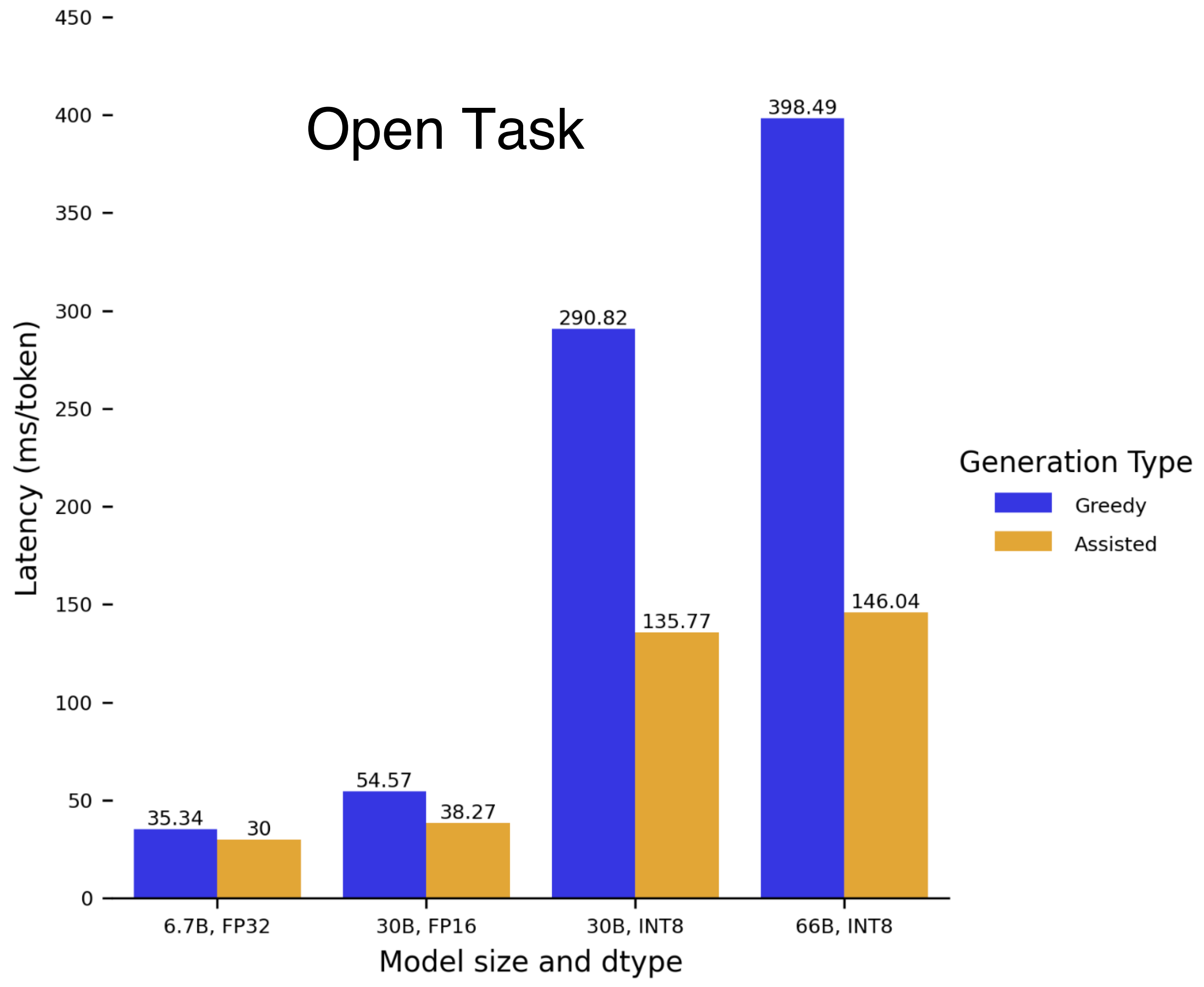
在分离式 Draft model 思想上，提出用同样结构的小模型 (SLM) 生成 candidates。

Model	d_{model}	Heads	Layers	Params	TPOT (ms)
Target (Chinchilla)	8192	64	80	70B	14.1
Draft	6144	48	8	4B	1.8

Sampling Method	Benchmark	Result	Mean Token Time	Speed Up
ArS (Nucleus)	XSum (ROUGE-2)	0.112	14.1ms/Token	1×
SpS (Nucleus)		0.114	7.52ms/Token	1.92×
ArS (Greedy)	XSum (ROUGE-2)	0.157	14.1ms/Token	1×
SpS (Greedy)		0.156	7.00ms/Token	2.01×
ArS (Nucleus)	HumanEval (100 Shot)	45.1%	14.1ms/Token	1×
SpS (Nucleus)		47.0%	5.73ms/Token	2.46×







Assistant Model

- `facebook/opt-125m`

Model Names:

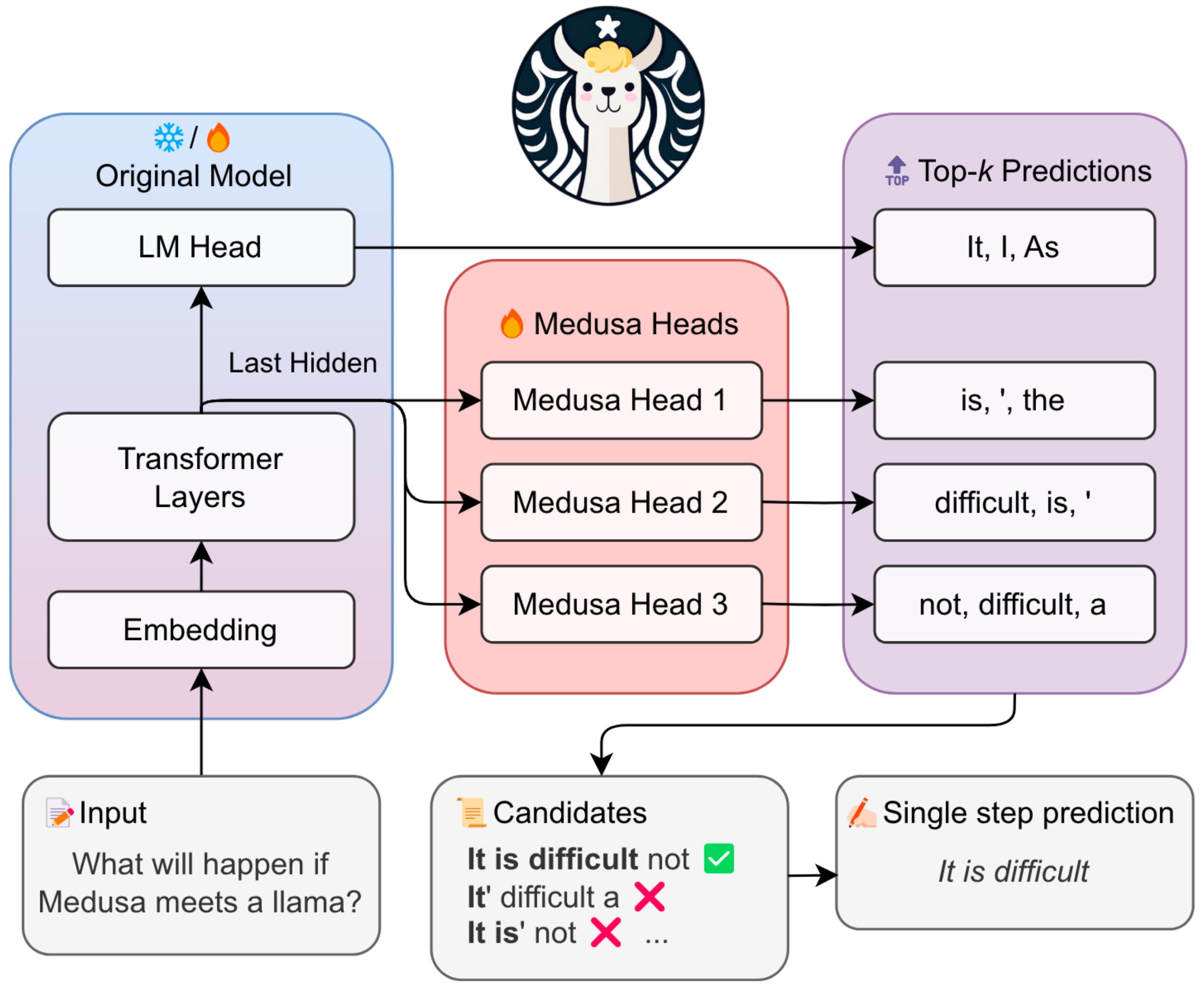
- 1.3B: `facebook/opt-1.3b`
- 6.7B: `facebook/opt-6.7b`
- 30B: `facebook/opt-30b`
- 66B: `facebook/opt-66b`

Dataset used as input prompt:

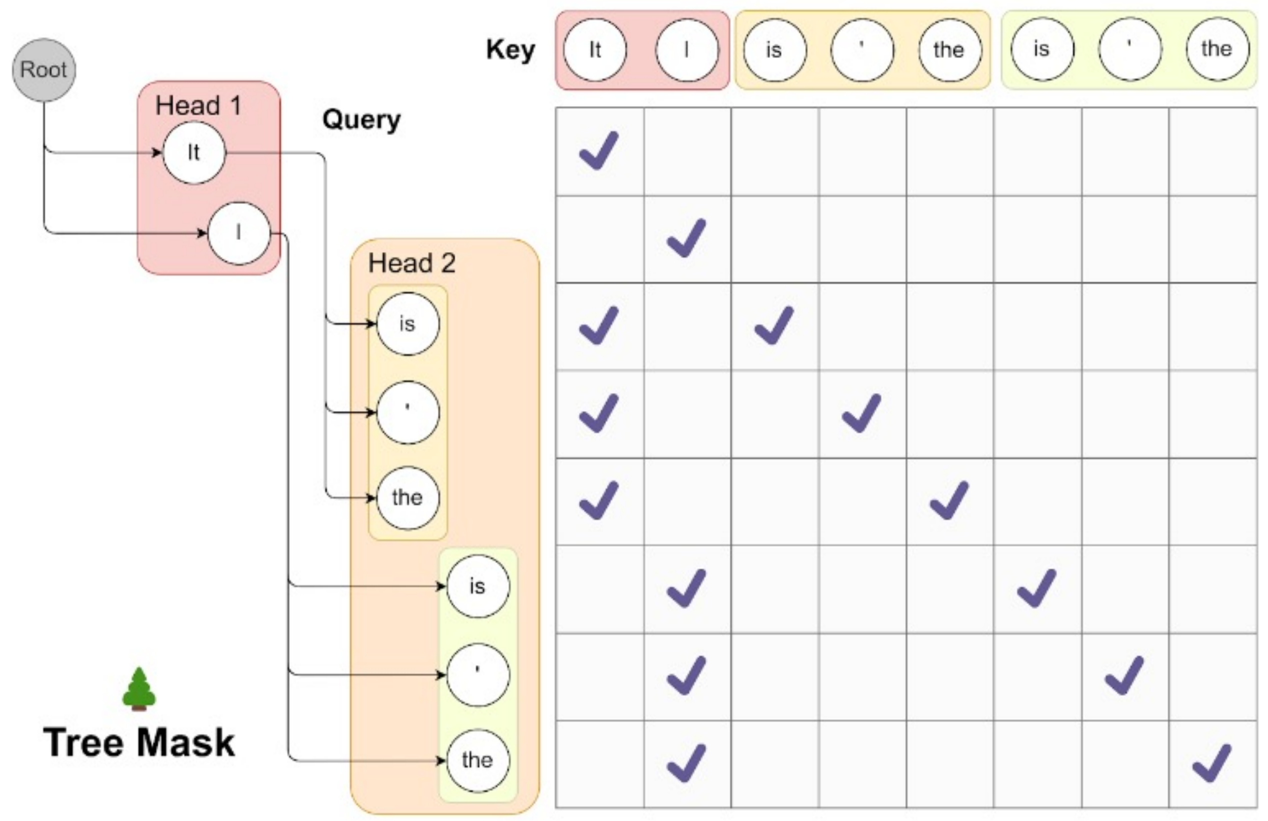
- C4 (en, validation set)

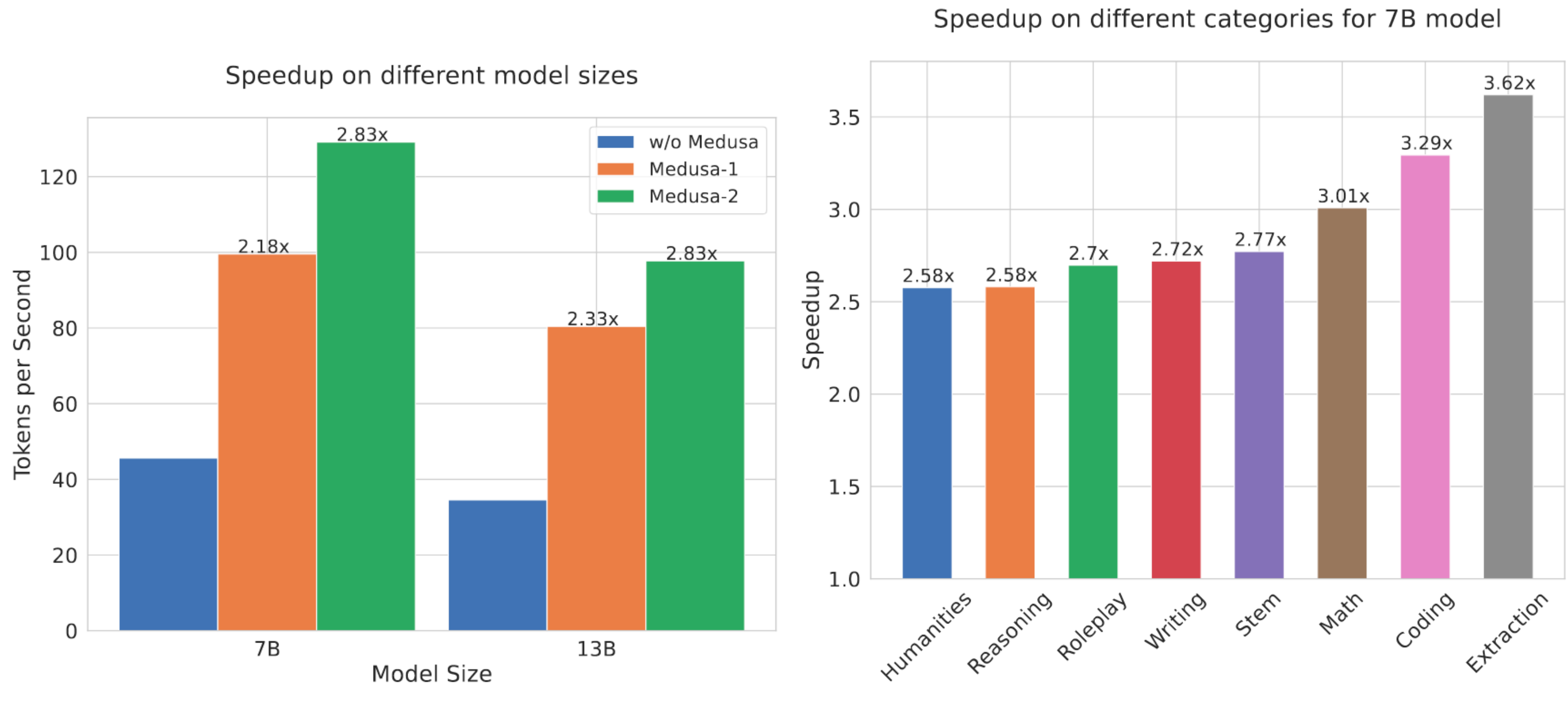
[1] Accelerating Large Language Model Decoding with Speculative Sampling. http://arxiv.org/abs/2302.01318
[2] Fast Inference from Transformers via Speculative Decoding. http://arxiv.org/abs/2211.17192
[3] HuggingFace Assisted Generation. https://huggingface.co/blog/assisted-generation

Medusa 训推协同



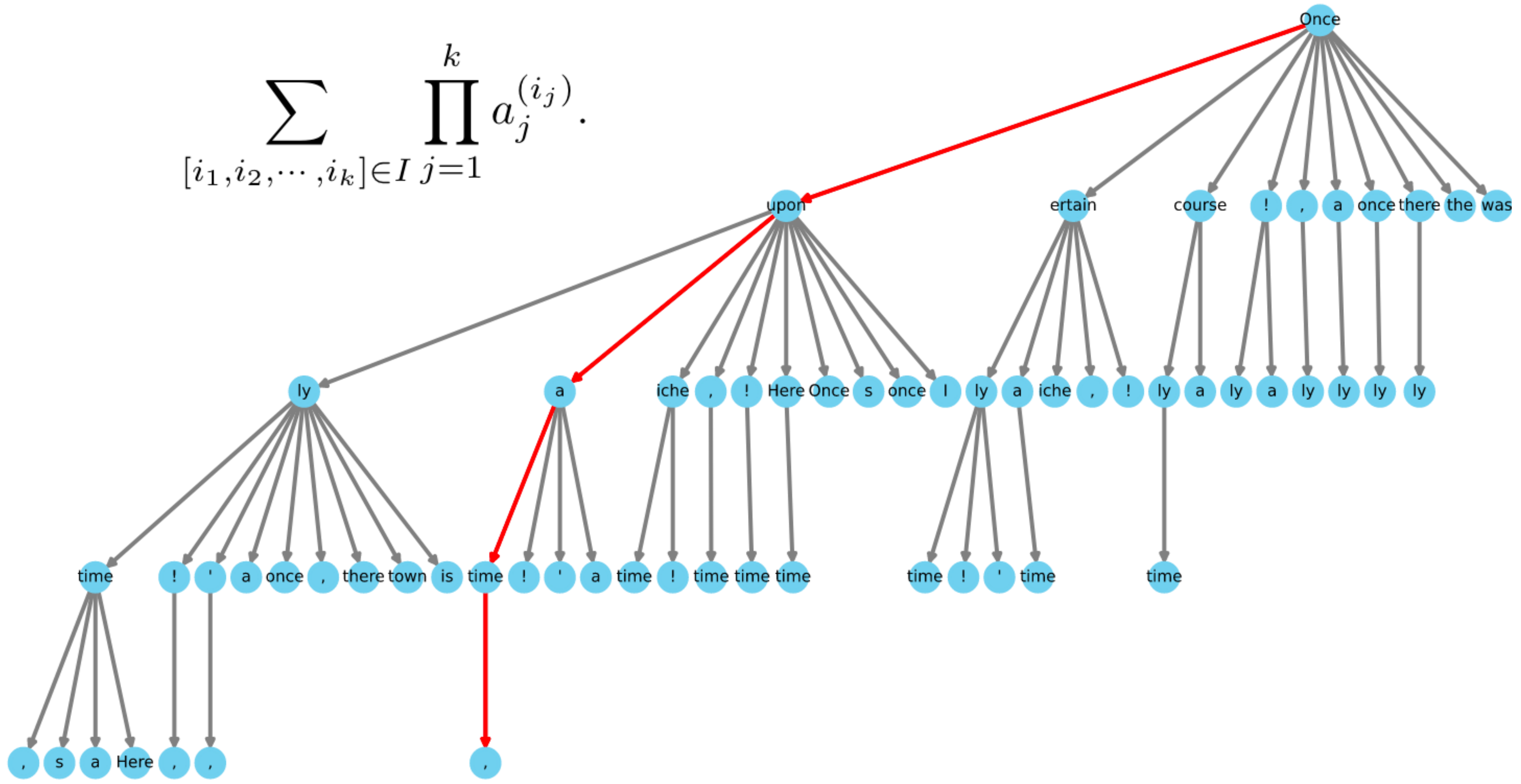
(B, 1, D) → (B, 1 + n_paths, D)





(a) (b)

$$\sum_{[i_1,i_2,\cdots,i_k]\in I}\prod_{j=1}^{k} a_j^{(i_j)}.$$



$$p_t^{(k)} = \mathrm{softmax}\left(W_2^{(k)} \cdot \left(\mathrm{SiLU}(W_1^{(k)} \cdot h_t) + h_t\right)\right), \text{ where } W_2^{(k)} \in \mathbb{R}^{d\times V}, W_1^{(k)} \in \mathbb{R}^{d\times d}.$$

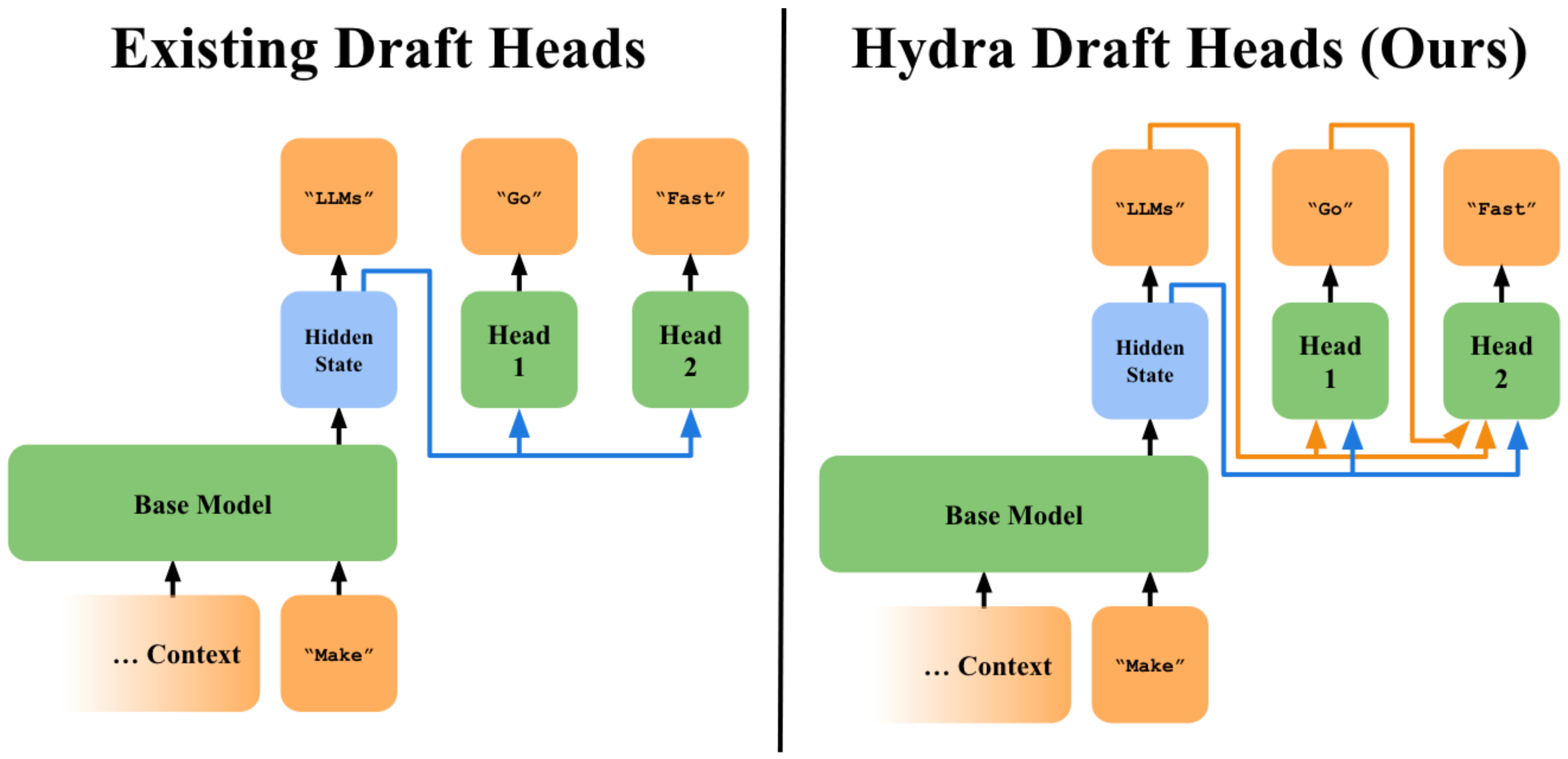
$$\mathcal{L}_{\text{MEDUSA-1}} = \sum_{k=1}^{K} -\lambda_k \log p_t^{(k)}(y_{t+k+1}).$$

$$\mathcal{L}_{\text{MEDUSA-2}} = \mathcal{L}_{\text{LM}} + \lambda_0 \mathcal{L}_{\text{MEDUSA-1}}.$$

$$\mathcal{L}_{\text{LM-distill}} = KL(p_{\text{original},t}^{(0)} || p_t^{(0)}),$$

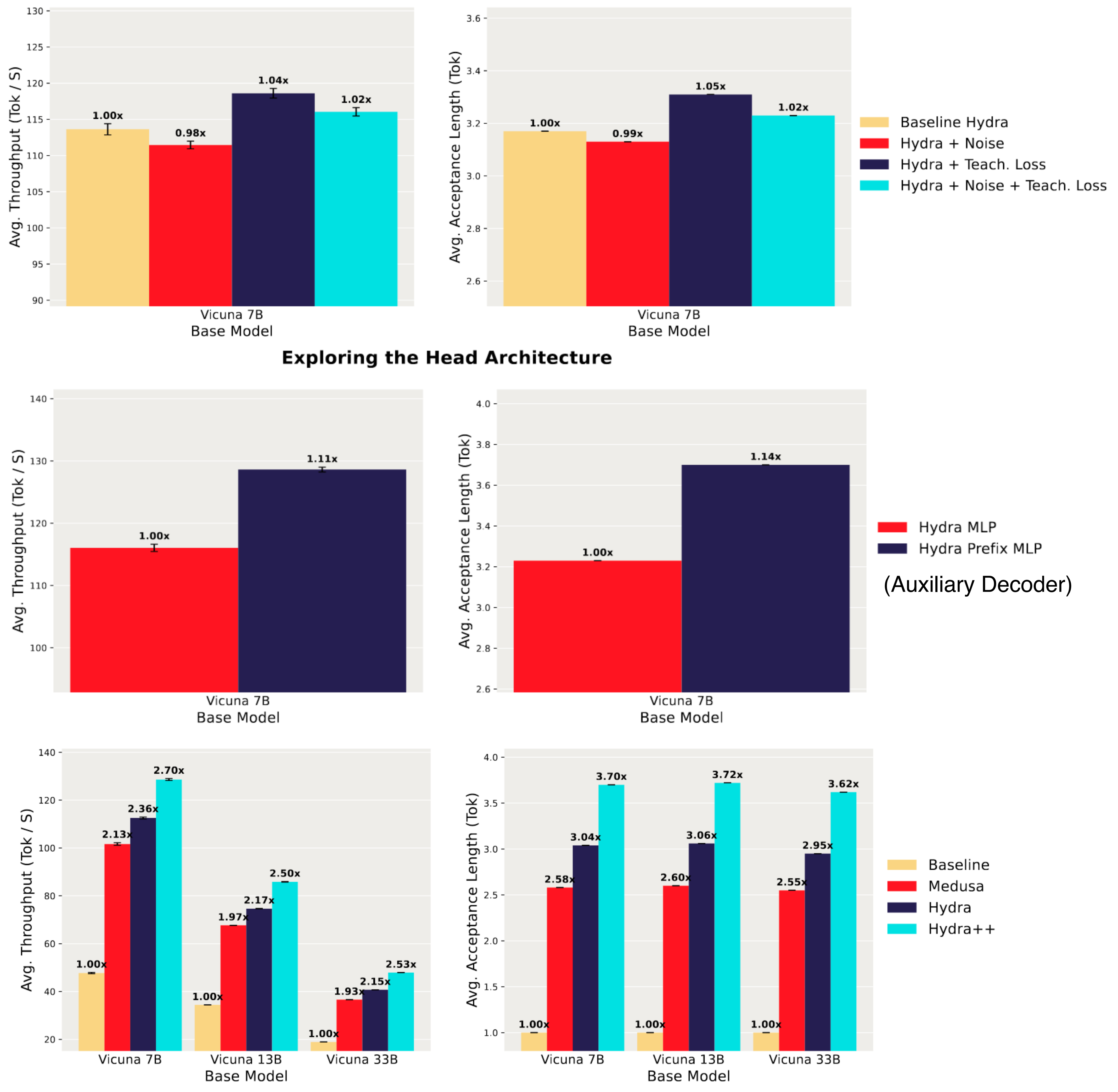
[1] Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads. http://arxiv.org/abs/2401.10774

Medusa 优化方向：Draft head



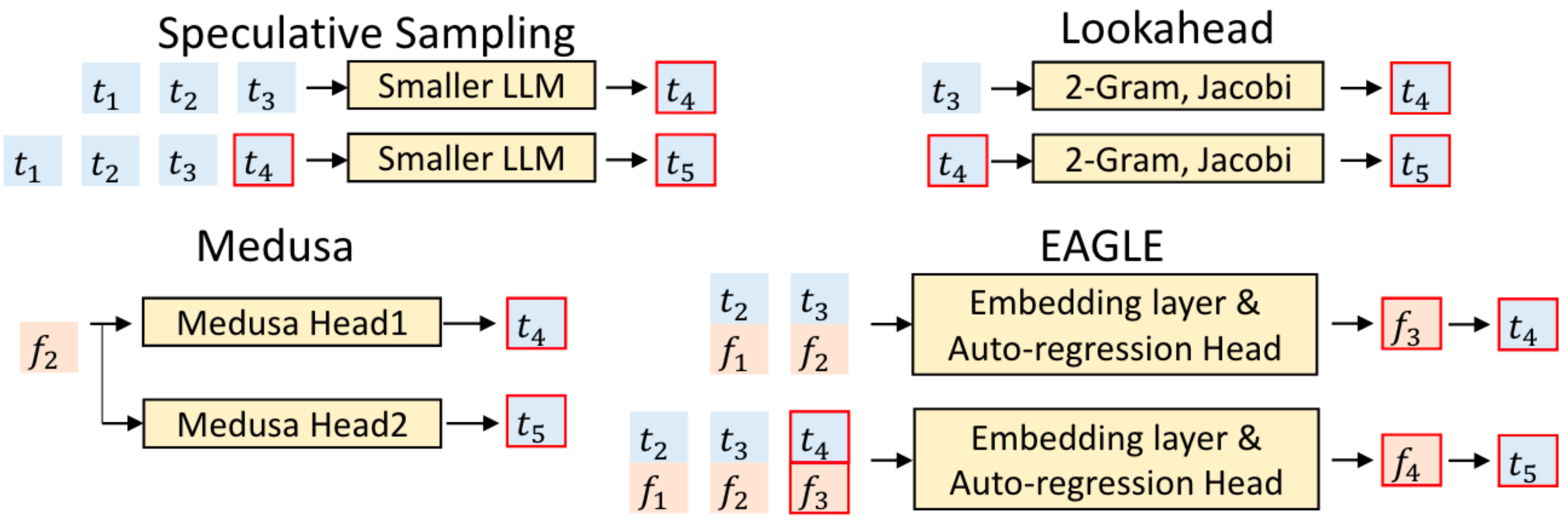
$$p_{\text{draft}}(\hat{x}_{t+i}|x_{\leq t}, \hat{x}_{t+1}, \ldots, \hat{x}_{t+i-1}) = p_{\text{draft}}(\hat{x}_{t+i}|x_{\leq t-1})$$

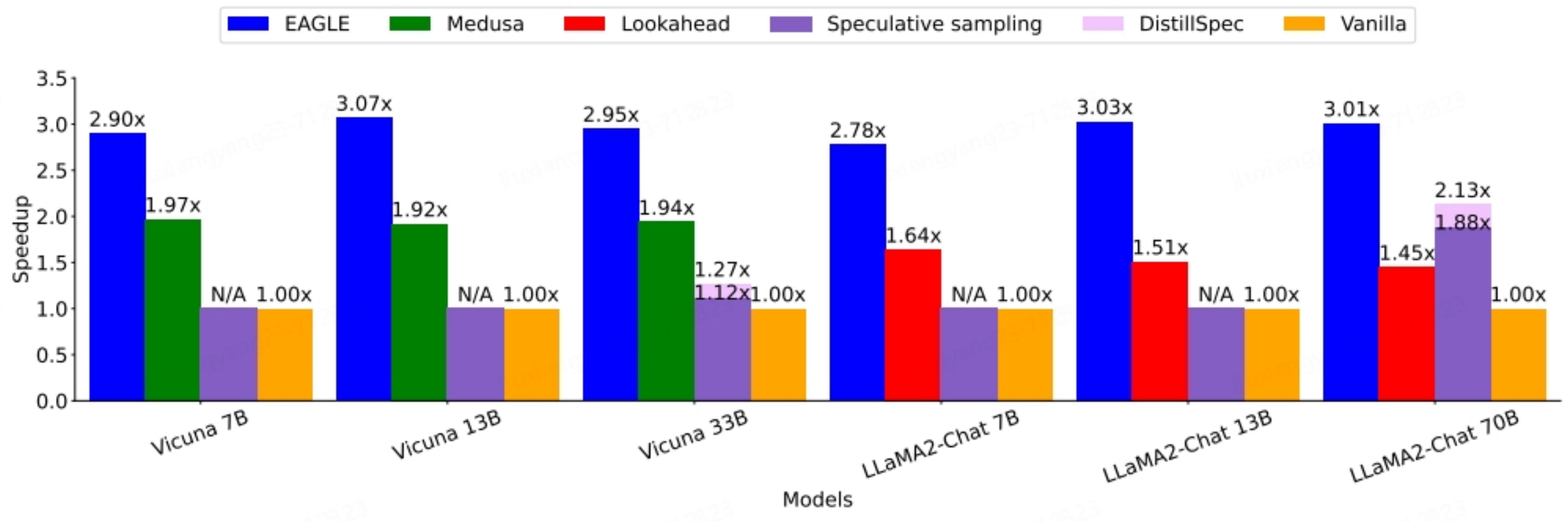
$$p_{\text{draft}}(\hat{x}_{t+i}|x_{\leq t}, \hat{x}_{t+1}, \ldots, \hat{x}_{t+i-1}) = f_{\text{Hydra},i}(h_{t-1}, x_t, \hat{x}_{t+1}, \ldots, \hat{x}_{t+i-1})$$

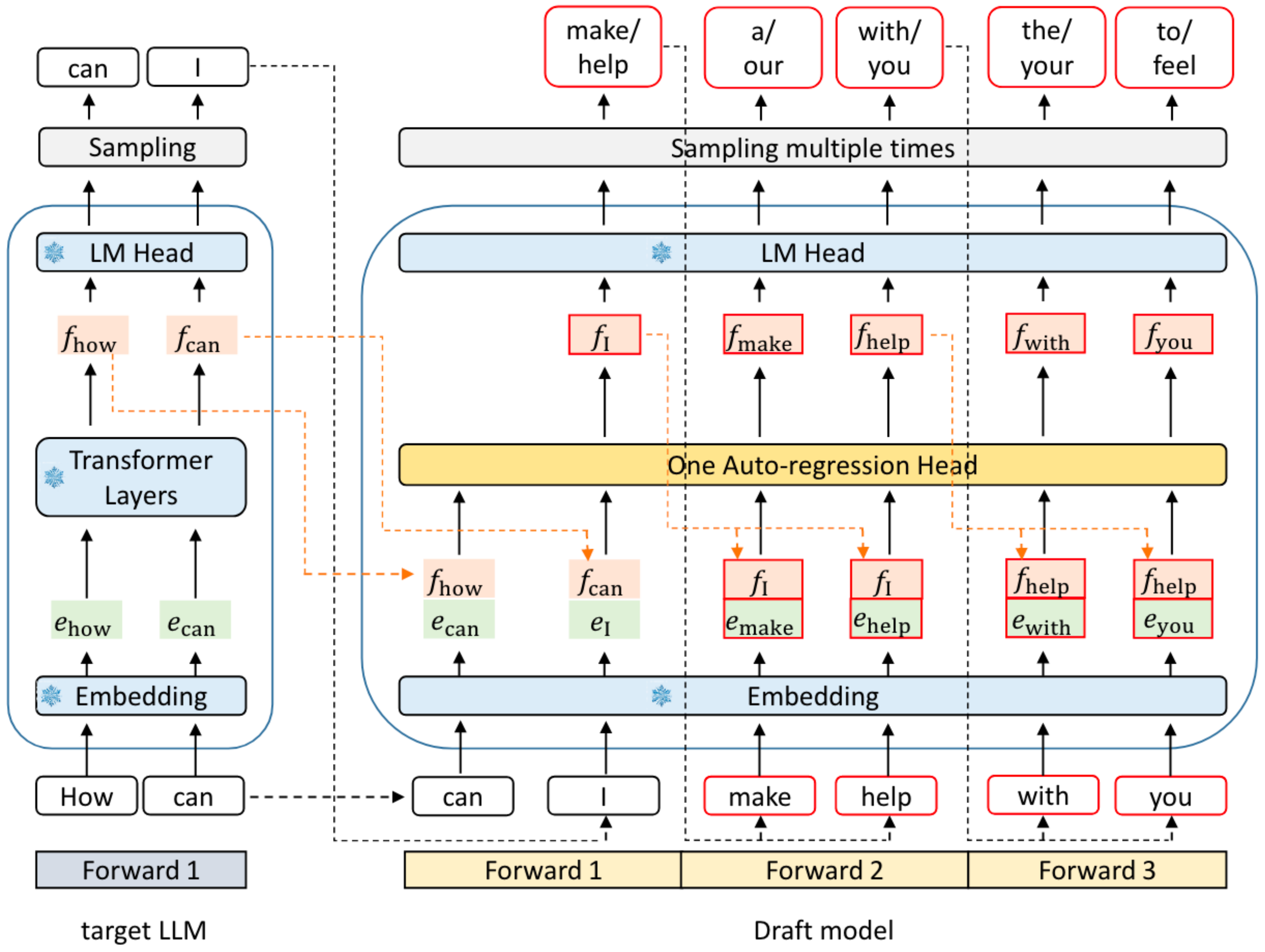


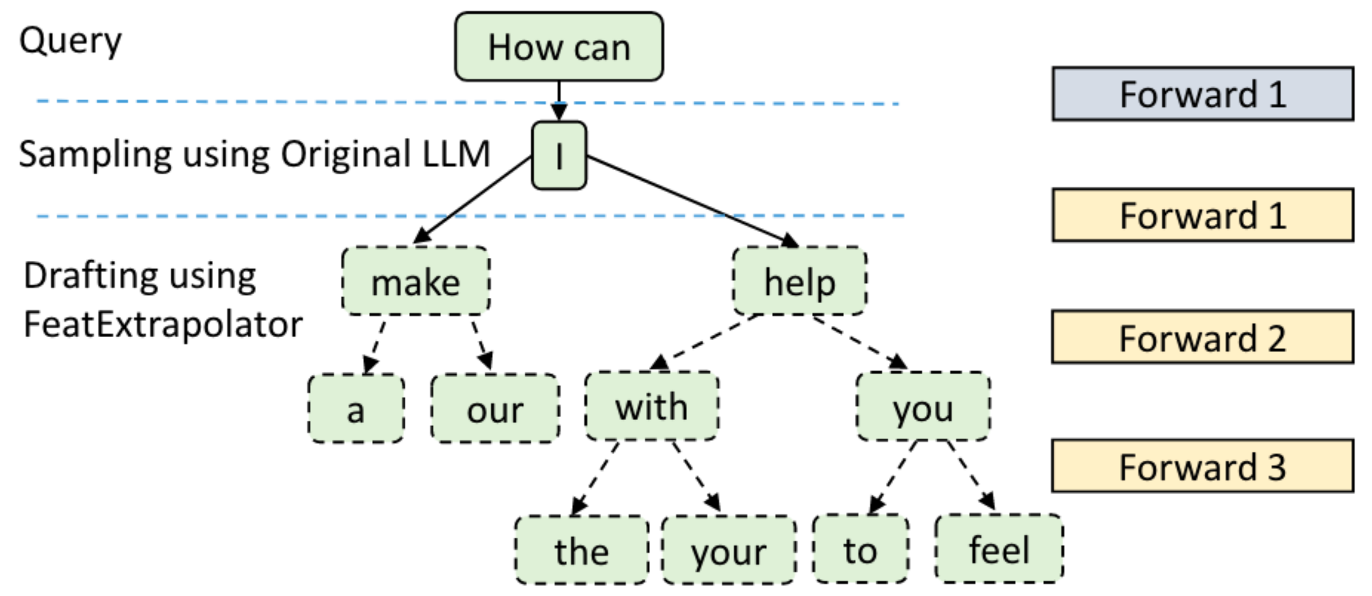
[1] Hydra: Sequentially-Dependent Draft Heads for Medusa Decoding. http://arxiv.org/abs/2402.05109

Medusa 优化方向：Draft head









Top2

[1] EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty. http://arxiv.org/abs/2401.15077

Medusa 优化方向：Draft tokens

Algorithm 1 Parallel Speculative Sampling (PaSS) with Parallel Look-ahead Embeddings

Given L look-ahead tokens $\texttt{[LA]}_1, \ldots, \texttt{[LA]}_L$ and minimum target sequence length T.
Given auto-regressive target model $q(.|.)$ and initial prompt sequence $x_0, \ldots, x_t$.
Initialise $n \leftarrow t$.
while $n < T$ **do**
  In parallel, sample the next token x_{n+1} and L draft tokens $\tilde{x}_1, \ldots, \tilde{x}_L$:

$$x_{n+1} \sim q(x|x_1, \ldots, x_n),\ \tilde{x}_1 \sim q(x|x_1, \ldots, x_n, \texttt{[LA]}_1),\ \ldots,\ \tilde{x}_L \sim q(x|x_1, \ldots, x_n, \texttt{[LA]}_1, \ldots, \texttt{[LA]}_L)$$

  Set $n \leftarrow n + 1$
  In parallel, compute $L + 1$ sets of logits from drafts $\tilde{x}_1, \ldots, \tilde{x}_L$:

$$q(x|x_1, \ldots, x_n),\ q(x|x_1, \ldots, x_n, \tilde{x}_1),\ \ldots,\ q(x|x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_L)$$

  for $t = 1 : L$ **do**
    Sample $r \sim U[0, 1]$ from a uniform distribution.
    if $r < \min\left(1, \frac{q(\tilde{x}_t|x_1, \ldots, x_{n-1}, \ldots, x_{n+t-1})}{q(\tilde{x}_t|x_1, \ldots, x_{n-1}, \texttt{[LA]}_1, \ldots, \texttt{[LA]}_t)}\right)$ **then**
      Set $x_{n+t} \leftarrow \tilde{x}_t$ and $n \leftarrow n + 1$
    else
      Sample

$$x_{n+t} \sim (q(x|x_1, \ldots, x_{n-1}, \ldots, x_{n+t-1}) - q(x|x_1, \ldots, x_{n-1}, \texttt{[LA]}_1, \ldots, \texttt{[LA]}_t))_+$$

      and Exit for loop.
    end if
  end for
  If all L tokens $x_{n+1}, \ldots, x_{n+L}$ are accepted, sample extra token $x_{n+L+1} \sim q(x|x_1, \ldots, x_{n+L})$ and set $n \leftarrow n + 1$.
end while

	The Stack			Wikipedia		
Temperature	0.8	0.5	0.2	0.8	0.5	0.2
Auto-regressive	12.52	12.69	12.72	12.45	12.30	12.55
[UNK] look-ahead	12.25	12.43	12.26	12.30	12.16	11.88
PaSS	9.79	9.46	8.96	10.23	9.78	9.43

# LA tokens	Time
2	10.03
4	9.79
6	9.66
8	9.94

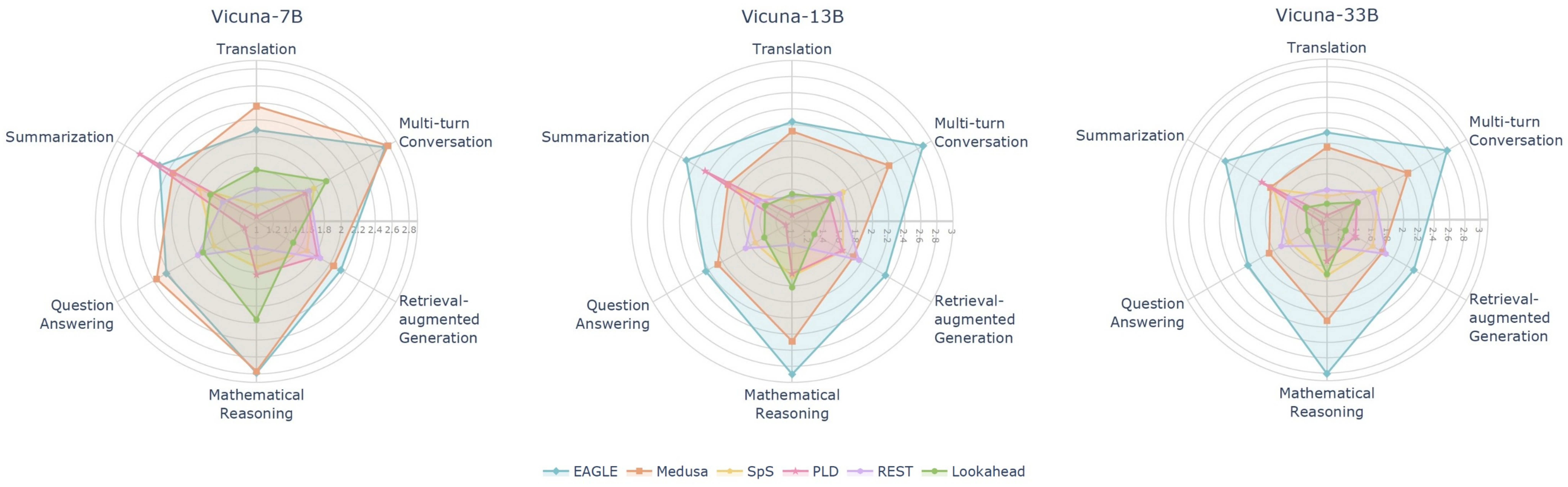
输入 32，输出 512

	PASS@1		PASS@10	
	Time	Perf.	Time	Perf.
Auto-regressive	10.52 sec	13.2 %	10.15 sec	22.5 %
PaSS	7.17 sec	13.4 %	8.17 sec	22.5 %

HumanEval 4-Lookahead tokens

[1] PaSS: Parallel Speculative Sampling. http://arxiv.org/abs/2311.13581

推测解码 Benchmark



Spec-Bench

Subtask	Dataset	#Samples
Multi-turn Conversation	MT-bench	80
Translation	WMT14 DE-EN	80
Summarization	CNN/Daily Mail	80
Question Answering	Natural Questions	80
Mathematical Reasoning	GSM8K	80
Retrieval-aug. Generation	Natural Questions	80
Overall	-	480

Vicuna-7B-v1.3

Models	Multi-turn Conversation	Translation	Summa-rization	Question Answering	Mathematical Reasoning	Retrieval-aug. Generation	Overall
Medusa 🥇	**2.79x**	**2.36x**	2.14x	**2.36x**	2.77x	2.05x	**2.42x**
EAGLE 🥈	2.75x	2.08x	2.32x	2.23x	**2.79x**	**2.15x**	2.39x
Hydra 🥉	2.51x	2.01x	1.84x	2.09x	2.58x	1.83x	2.15x
Lookahead	1.95x	1.61x	1.63x	1.73x	2.16x	1.50x	1.77x
PLD	1.67x	1.06x	**2.59x**	1.16x	1.63x	1.83x	1.66x
REST	1.72x	1.38x	1.46x	1.80x	1.31x	1.87x	1.59x
SpS	1.78x	1.19x	1.78x	1.58x	1.54x	1.69x	1.59x

Vicuna-33B-v1.3

Models	Multi-turn Conversation	Translation	Summa-rization	Question Answering	Mathematical Reasoning	Retrieval-aug. Generation	Overall
EAGLE 🥇	**2.81x**	**2.14x**	**2.53x**	**2.19x**	**3.01x**	**2.31x**	**2.50x**
Hydra 🥈	2.63x	2.05x	2.08x	2.16x	2.76x	2.11x	2.31x
Medusa 🥉	2.22x	1.95x	1.85x	1.87x	2.32x	1.84x	2.01x
SpS	1.79x	1.31x	1.80x	1.57x	1.73x	1.69x	1.65x
REST	1.71x	1.39x	1.57x	1.69x	1.34x	1.89x	1.59x
PLD	1.45x	1.06x	1.98x	1.07x	1.54x	1.43x	1.41x
Lookahead	1.46x	1.21x	1.32x	1.29x	1.71x	1.28x	1.38x

[1] Spec-Bench: A Comprehensive Benchmark and Unified Evaluation Platform for Speculative Decoding. https://sites.google.com/view/spec-bench

目录

未来展望

- Medusa
  - v1 通用高效的 Draft heads “自蒸馏” 微调，与知识蒸馏和重组等方法的结合
  - v2 Draft heads 监督与大模型训练流程 (Pretrain + SFT + RLHF) 的协同
  - 大参数量/长词表 Draft heads 接收率低，高频词分类子空间划分
  - 高效的 Draft heads/tokens/candidates 等策略
  - Draft heads 和 candidates 数量之间的均衡，即实际场景下 heads 能接受的数量上限
  - “推测–验证–接收”模式下 KVCache 的高效管理
- 通用推理
  - 动态推理能力，Task/Layer/Token–level 模型结构自动调整和计算资源分配
  - 高精度的 KVCache 量化，以及与原始模型生成质量对齐技术
  - 高质量的 Prompt 压缩重写技术

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software