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

京东算法工程师 / 谢帅

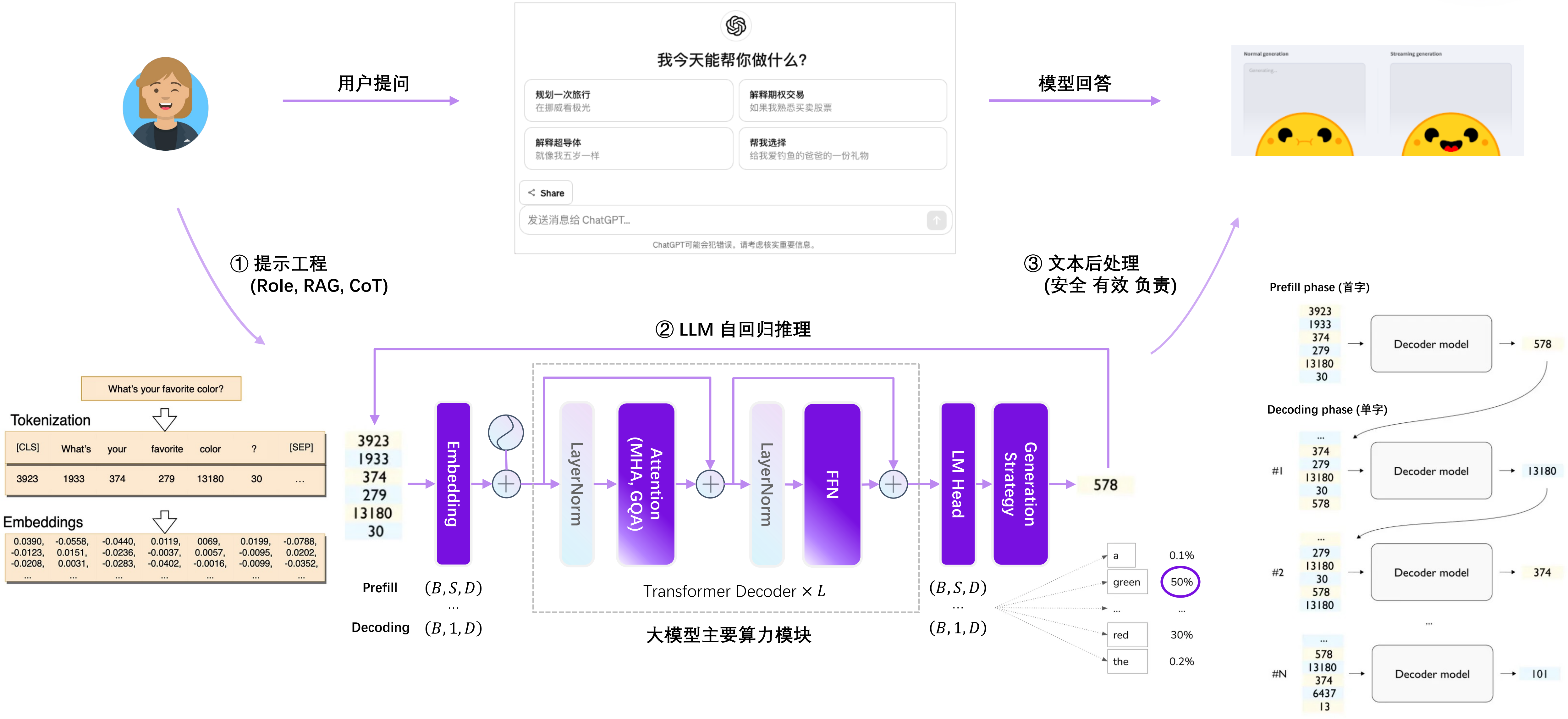
CONTENTS

Summary Of Work

1 大模型推理优化概览

- ✓ 大模型如何完成一次推理
- ✓ 推理优化指标及影响因素
- ✓ 推理优化技术概览
- ✓ 推理框架总结

大模型如何完成一次推理



推理优化指标及影响因素

【1】用户关心的问题 (SLOs)

精度
速度
“又快又好”

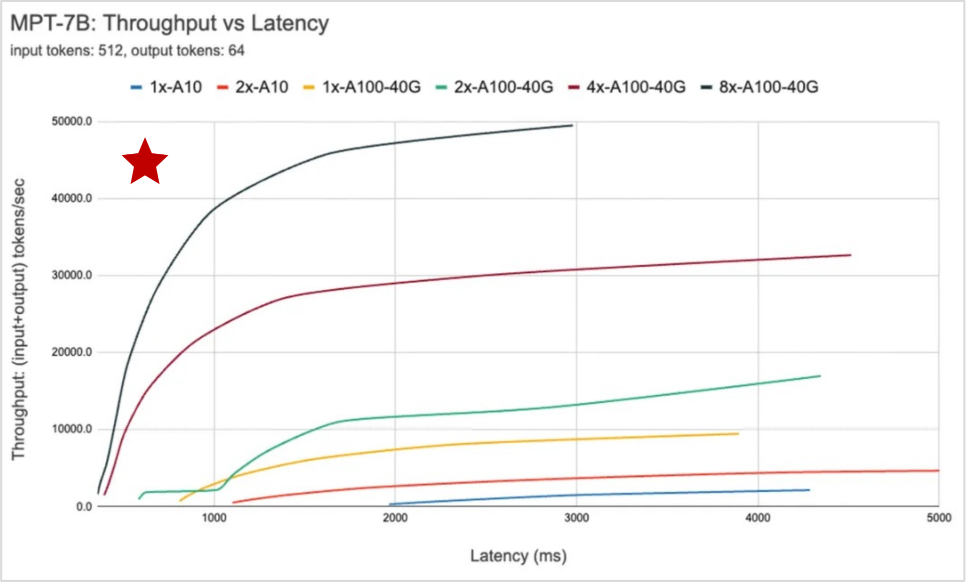
- 模型生成质量能否满足我的要求？→ 推理优化要对齐模型原本精度
  Accuracy 基本原则 (e.g. AlignBench score, PPL)
- 模型生成过程是否值得我的等待？→ 用户收到模型反馈不能等太久
  Latency TTFT (e.g. 首字 200ms)
- 模型生成速度能否跟上我的阅读？→ 模型每秒输出的字数要足够多
  Latency TPOT (e.g. 单字 50ms)

【2】工程师关心的问题 (多视角 SLOs + Cost)

吞吐
成本
“又便宜”

- 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 e.g. 单机承载 10 users 满足 90% SLOs

Throughput ↑ Latency ↓

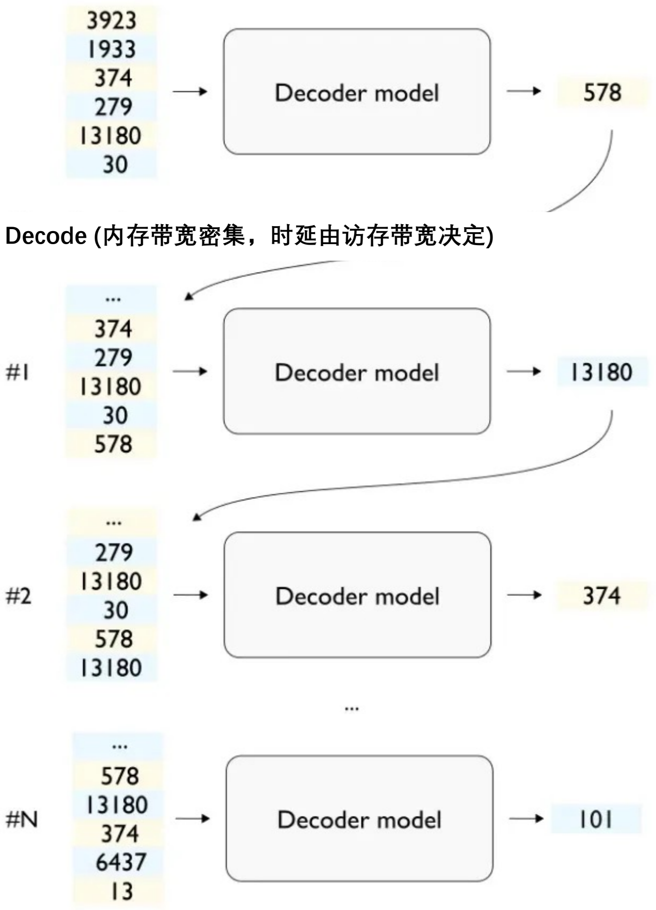


不同硬件资源下 MPT-7B Throughput 和 Latency 关系
(batchsize 逐渐从 1 增加到 256, T↑, L ↓)

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

推理优化指标及影响因素

Prefill (计算密集，时延由算力决定)

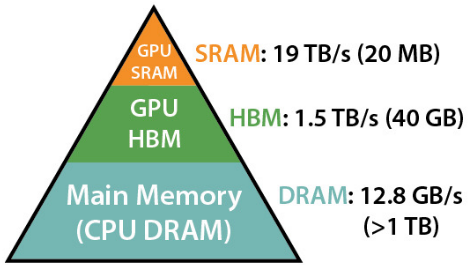


模型参数量 P，解码阶段 $S = 1$，FP16 精度 $m = 1$

计算量 (FLOPs)	$\sim 2 * P * B * S$	计算：稀疏化/并行解码
内存量 (GB)	$\sim 2 * P * m$	IO：量化/FA/GQA/KVCache

$$B^* = \frac{hardware\_flops}{mem\_bandwidth} \qquad B^+ = \frac{total\_mem - model\_mem}{max\_seq\_mem}$$ (KVCache)

硬件	显存 (GB)	算力 (TFLOPs)	带宽 (GB/s)	B^*	B^+
A100	40	312	1555	200	13
H100	80	590	3350	590	33



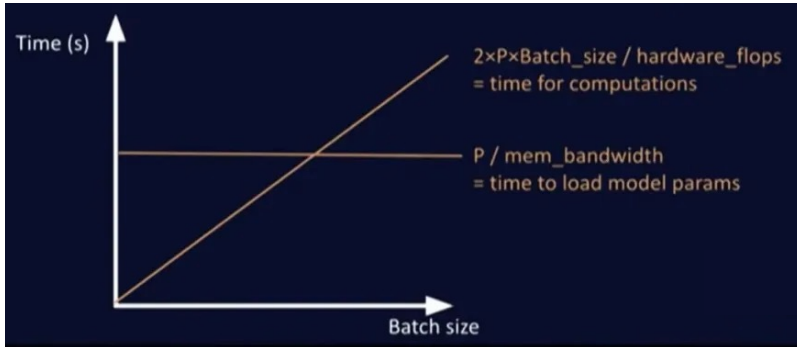
Memory Hierarchy with Bandwidth & Memory Size

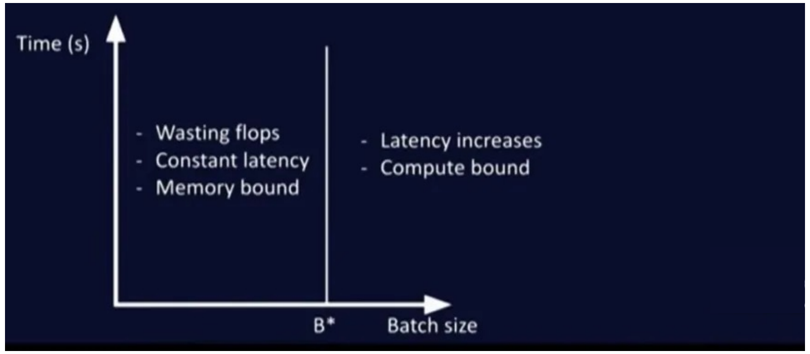
A100-40G

内存搬运时间 ≫ 模型计算时间

$$\frac{2 * 7}{1555} \gg \frac{2 * 7 * 13 * 10^9}{312 * 10^{12}}$$

以 A100-40G，Llama-7B 模型为例



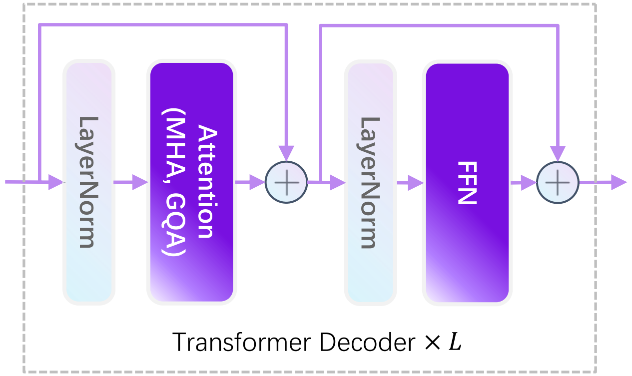


200
Peak Throughput

[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/D9KPNI3CJ88I5_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)

推理优化技术概览

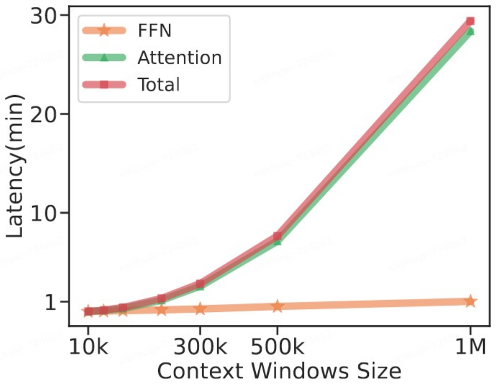
【1】大模型主要算力模块



① $GEMM(W, A)$

② $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$

【2】FFN/Attention 计算占比 vs. 句子长度



【1】量化 (IO)

- GPTQ, AWQ, GPTVQ, VPTQ, KVQuant, LLM.int8, SmoothQuant

【2】注意力 & KVCache (IO, 显存)

- Flash Attention/Decoding, StreamingLLM

【3】稀疏化 (计算)

- SparseGPT, Wanda, Double Sparsity

【4】推测解码 (计算)

- Speculative Sampling, Medusa, Hydra, EAGLE

【5】并行化 (计算)

- Megatron Tensor Parallel, Mooncake 4D Parallel

【6】批处理 (显存)

- Continuous Batching & Page Attention

[1] MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention. http://arxiv.org/abs/2407.02490
[2] Awesome-LLM-Inference: A curated list of Awesome LLM Inference Papers with codes. https://github.com/DefTruth/Awesome-LLM-Inference

推理框架总结

【1】服务端部署

- Text Generation Interface @HuggingFace
- vLLM @Berkeley
- LightLLM @ModelTC, SenseTime
- TensorRT-LLM, FasterTransformer, Triton-Inference-Server @NVIDIA
- LMDeploy @InternLM, ShanghaiAILab
- RTP-LLM @Alibaba
- SiliconLLM @SiliconFlow
- OmniForce-LLM @JD

Python-level 易用性强，迭代快，更易 DIY 引入 New Features

Cpp/Cuda-level 性能更强，优化更底层，更深入硬件特性

【2】端侧部署

- Ollama
- Llama.cpp
- MLC-LLM @MLC-AI
- PowerInfer @STJU
- JittorLLMs @Tsinghua

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

Summary Of Work

2 推理 & 训练协同演进

- ✓ 训推协同优化方法总览
- ✓ GQA
- ✓ StreamingLLM

推理与训练协同演进

训推协同优化

【1】量化 (显存/超低 bit 精度)

- QLoRA, OneBit QAT

【2】注意力 & KVCache (IO/显存)

- Flash Attention, GQA, StreamingLLM

【3】推测解码 (计算)

- Medusa2

【4】稀疏化 (计算)

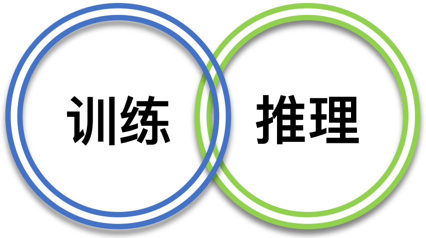
- Sparse Training

【5】并行化 (计算)

- 4D Parallel

【6】自适应模型 (计算)

- Early Exit, Mixture of Depth (MoD)



通用推理技术 (Post Training)

【1】量化 (IO)

- GPTQ, AWQ, GPTVQ, VPTQ, KVQuant, LLM.int8, SmoothQuant

【2】注意力 & KVCache (IO/显存)

- Flash Attention/Decoding, StreamingLLM

【3】推测解码 (计算)

- Speculative Sampling, Medusa, Hydra, EAGLE

【4】稀疏化 (计算)

- SparseGPT, Wanda, Double Sparsity

【5】并行化 (计算)

- Megatron Tensor Parallel, Mooncake 4D Parallel

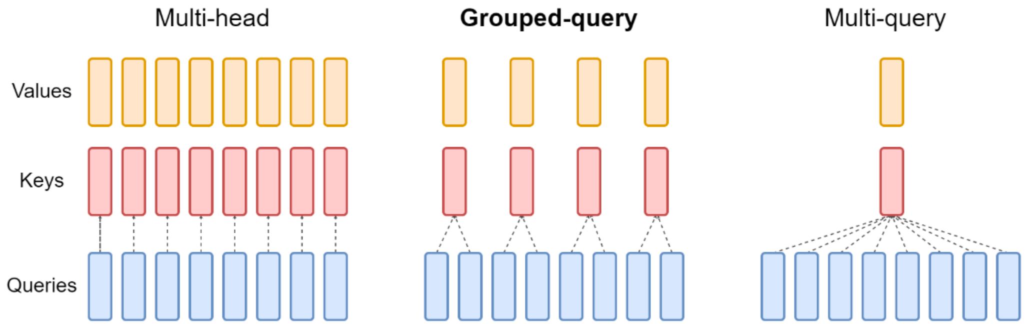
【6】批处理 (显存)

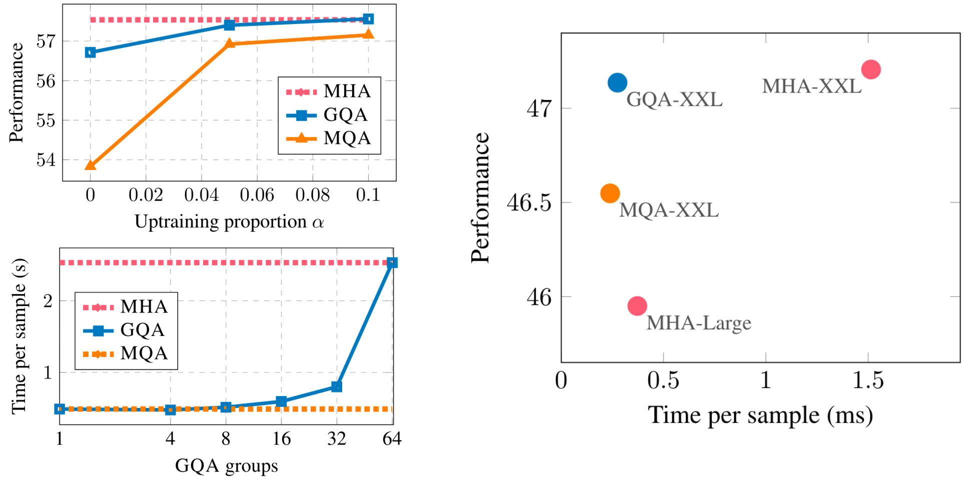
- Continuous Batching & Page Attention

训推协同优化：GQA

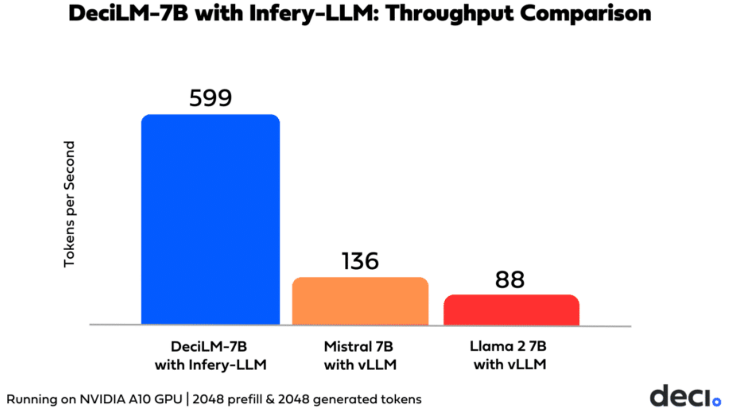
训练 → 推理

【1】MHA vs. GQA vs. MQA





【2】Variable GQA



deci
DeciLM-7B
Open LLM Leaderboard Scores

Model	Leaderboard Average	ARC	HellaSwag	MMLU	Truthful QA	Winogrande	GSM8K
DeciLM 7B-Base	61.55	59.39	82.51	59.76	40.33	79.95	47.38
Mistral-7B-v0.1	60.97	59.98	83.31	64.14	42.15	78.37	37.83
Vicuna-13B-v1.5	55.41	57.08	81.24	56.67	51.51	74.66	11.30
Llama 2 13B-chat-hf	54.91	59.04	81.94	54.64	44.12	74.51	15.24
Llama 2-7B-hf	50.97	53.07	78.59	46.87	38.76	74.03	14.48

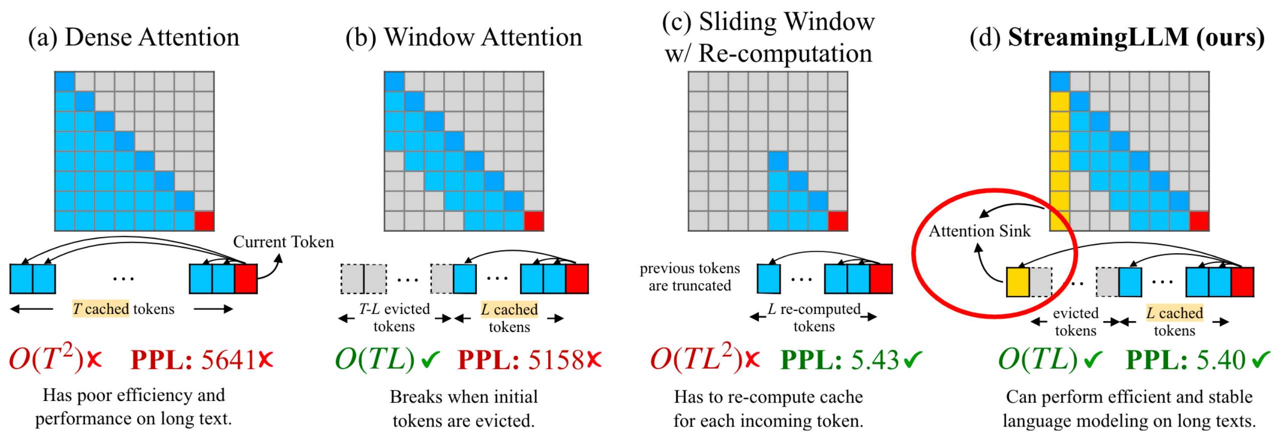
```
# 常见 GQA 模型，group_size=8，如 Mistral-7B，GQA 可将此值进一步缩小到 1,2,4 组合
"num_key_value_heads_per_layer": [4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2,
                                  2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4]
```


[1] GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. http://arxiv.org/abs/2305.13245
[2] LLM推理入门指南②：深入解析KV缓存. https://mp.weixin.qq.com/s/WxbMFoSrKl0xqsUkzPLJHw

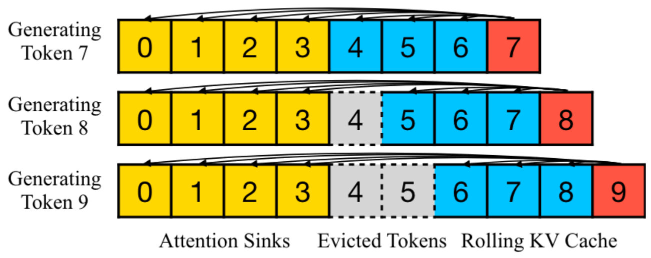
训推协同优化：StreamingLLM

推理 → 训练

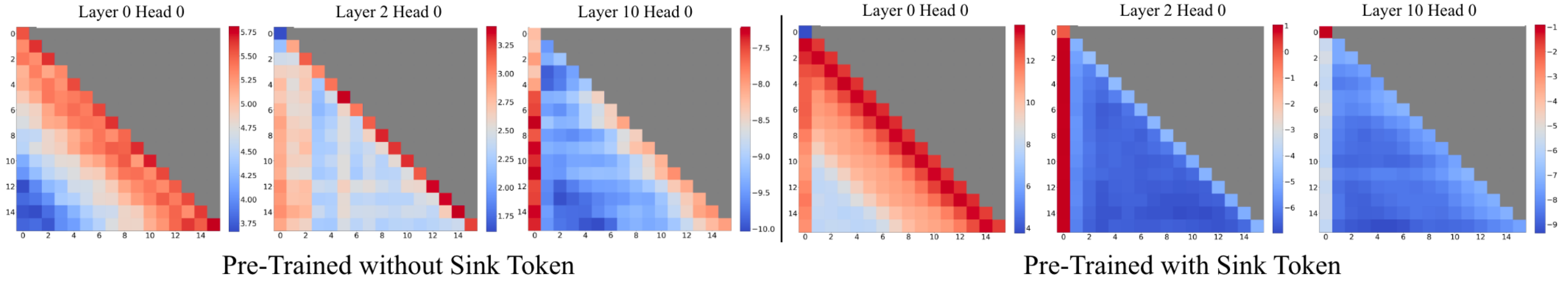
【1】StreamingLLM 计算复杂度 & 恒定 KV Cache



【2】StreamingLLM KV Cache 更新策略



【3】Attention logits 可视化 (Pre-Trained with vs. without Sink Token)



[1] StreamingLLM: Efficient Streaming Language Models with Attention Sinks. http://arxiv.org/abs/2309.17453

Summary Of Work

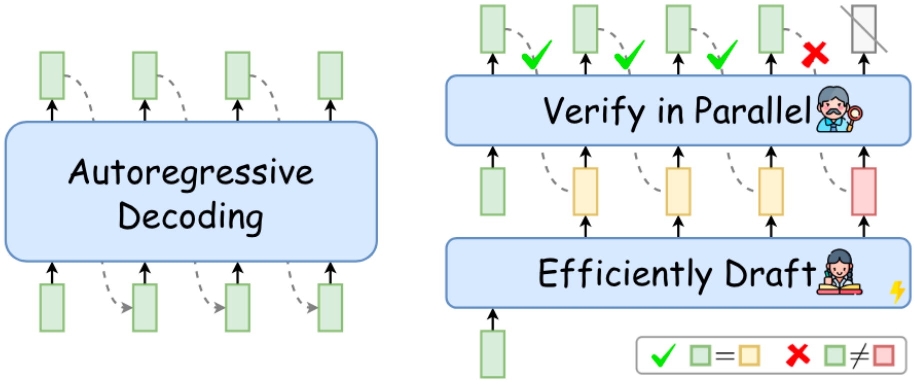
3 Medusa与推测解码

- ✓ 推测解码浅析 & 发展
- ✓ 分离式大小模型 (Speculative Sampling)
- ✓ 组合式共生模型 (Medusa Parallel Decoding)
- ✓ Medusa 实践 & 优化

推测解码浅析

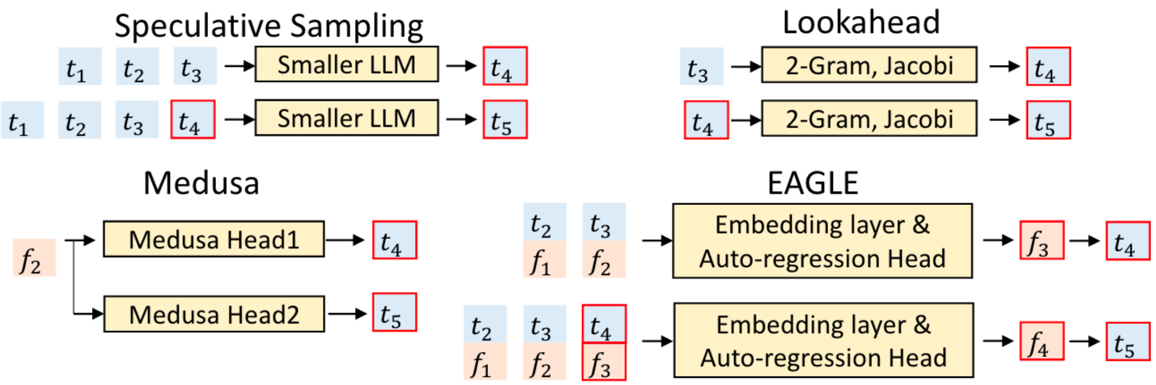
【1】自回归解码 vs. 推测解码

“单词” → “词组”



$k = 1$ real token　　$k = 5$ draft tokens

【2】常见的 Draft tokens 生成方式



Draft tokens:

低成本推理快的 “小模型” 生成的可能正确的 tokens

推测解码应用 Draft tokens 的三步:

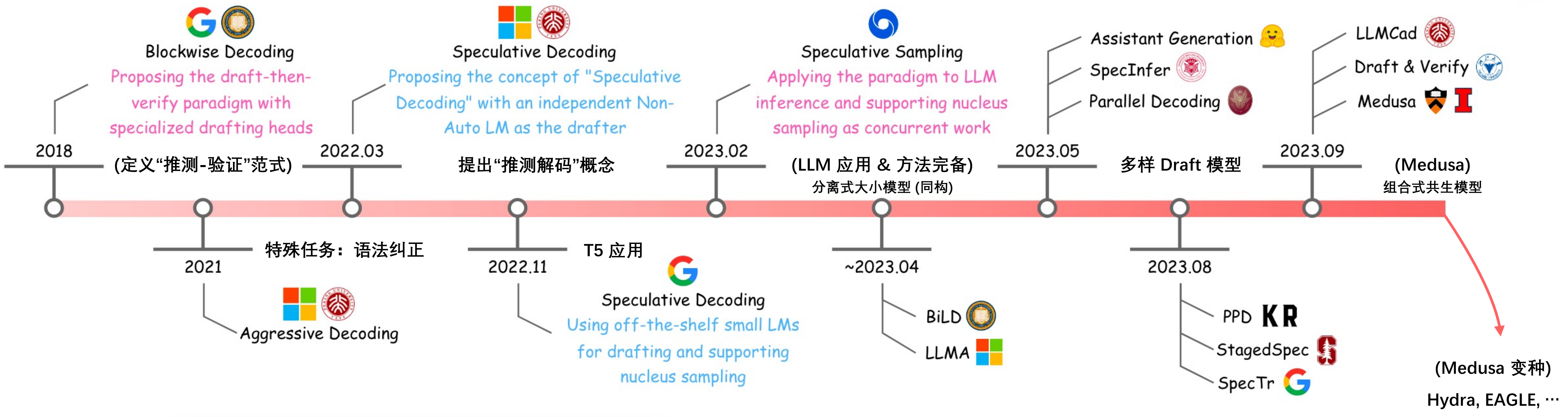
① 推测 → ② 验证 → ③ 接收

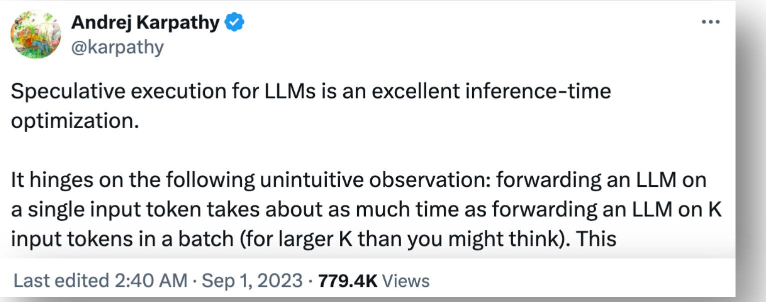
【3】不同采样策略下推测解码的验证方法

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

[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

推测解码时间线



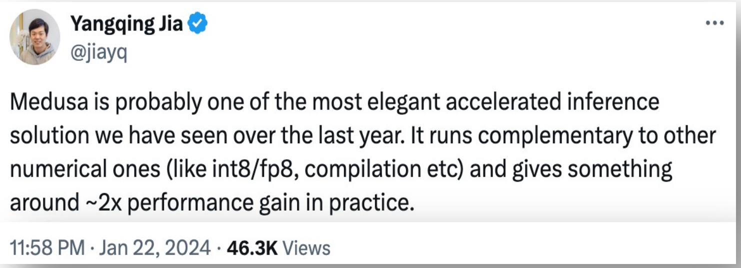


Andrej Karpathy @karpathy

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

Speculative Sampling

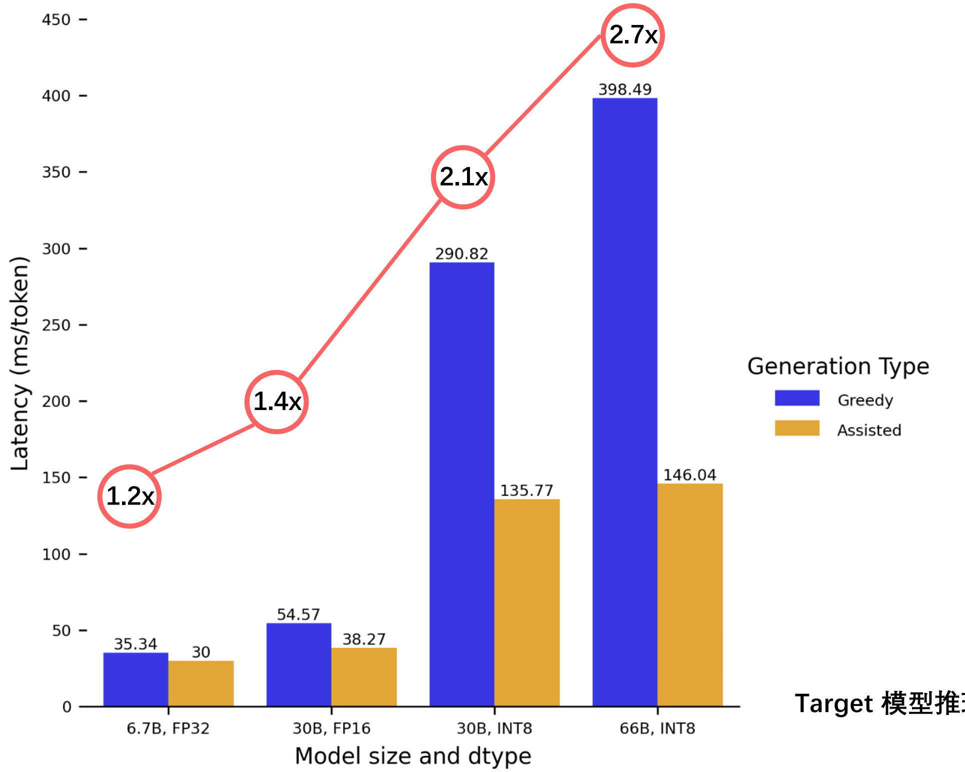
动机： ① Decoding 阶段算力未充分应用 **推理时间 $T_k \approx T_1$**，② LLMs 同构小参数模型与 Target 模型分布接近，更适合做 Draft
贡献： ① 验证了大小模型协同的推测解码的有效性 (**2.5x**)，② 证明 SpS 采样策略与 ArS 采样结果等价

【1】Assisted Generation: HuggingFace 第7种解码策略

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

1 Target tok = 7.8 Draft toks

【2】OPT 不同参数量模型 FP16, INT8 加速比例



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)

Target 模型推理时间越长，加速越明显

[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

Speculative Sampling

DataFun. | DA数智大会 Data+AI Conference

动机： ① Decoding 阶段算力未充分应用 **推理时间 $T_k \approx T_1$**，② LLMs 同构小参数模型与 Target 模型分布接近，更适合做 Draft

贡献： ① 验证了大小模型协同的推测解码的有效性 (**2.5x**)，② 证明 SpS 采样策略与 ArS 采样结果等价

【1】Speculative Sampling 采样策略

Algorithm 2 Speculative Sampling (SpS) with Auto-Regressive Target and Draft Models

Given lookahead K and minimum target sequence length T.
Given auto-regressive target model $q(.|.)$, and auto-regressive draft model $p(.|.)$, initial prompt sequence $x_0, \dots, x_t$.　　实际分布　　推测分布
Initialise $n \leftarrow t$.
while $n < T$ **do**

1. Draft
　for $t = 1 : K$ **do**
　　Sample draft auto-regressively $\tilde{x}_t \sim p(x|, x_1, \dots, x_n, \tilde{x}_1, \dots, \tilde{x}_{t-1})$　串行从推测分布采样 K drafts
　end for

2. Verify
　In parallel, compute $K+1$ sets of logits from drafts $\tilde{x}_1, \dots, \tilde{x}_K$:　并行计算所有 drafts 实际分布

$$q(x|, x_1, \dots, x_n),\ q(x|, x_1, \dots, x_n, \tilde{x}_1),\ \dots,\ q(x|, x_1, \dots, x_n, \tilde{x}_1, \dots, \tilde{x}_K)$$

　for $t = 1 : K$ **do**　　Q：推测分布 $p(x)$ 和实际分布 $q(x)$ 采样是否等价?
　　Sample $r \sim U[0, 1]$ from a uniform distribution.　　Modifed Rejection Sampling
　　if $r < \min\left(1, \frac{q(x|x_1,\dots,x_{n+t-1})}{p(x|x_1,\dots,x_{n+t-1})}\right)$, **then**　　Accepted
　　　Set $x_{n+t} \leftarrow \tilde{x}_t$ and $n \leftarrow n+1$.
　　else　　Rejected
　　　sample $x_{n+t} \sim (q(x|x_1, \dots, x_{n+t-1}) - p(x|x_1, \dots, x_{n+t-1}))_+$ and exit for loop.
　　end if　　候选 tokens 子分布 (ReLU + 求和归一化)
　end for
　If all tokens $x_{n+1}, \dots, x_{n+K}$ are accepted, sample extra token $x_{n+K+1} \sim q(x|, x_1, \dots, x_n, x_{n+K})$ and set $n \leftarrow n+1$.
end while

[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

【2】SpS 与 ArS 采样等价性证明

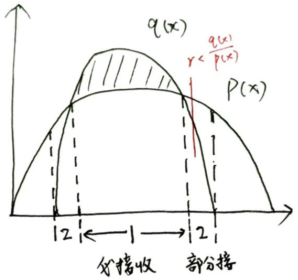
Theorem 1 (Modified Rejection Sampling recovers the target distribution). *Given discrete distributions q, p and a single draft sample $\tilde{x} \sim p$, let X be the final resulting sample. For $X = x$ to be true, we must either sample $\tilde{x} = x$ and then accept it, or resample it after $\tilde{x}$ (of any value) is rejected. Hence:*

$$\mathbb{P}(X = x)$$
$$= \mathbb{P}(\tilde{x} = x)\mathbb{P}(\tilde{x}\ accepted|\tilde{x} = x) + \mathbb{P}(\tilde{x}\ rejected)\mathbb{P}(X = x|\tilde{x}\ rejected)$$

x 被采样的两个条件概率：首次 + 二次
证明二者的和 $= q(x)$

For the first term, we apply the acceptance rule:

$$\mathbb{P}(\tilde{x} = x)\mathbb{P}(\tilde{x}\ accepted|\tilde{x} = x)$$
$$= p(x)\min\left(1, \frac{q(x)}{p(x)}\right)$$
$$= \min(p(x), q(x))$$



For the second conditional term, we apply the resampling rule:

$$\mathbb{P}(X = x|\tilde{x}\ rejected) = (q(x) - p(x))_+$$

Where $(.)_+$ denotes:

$$(f(x))_+ = \frac{\max(0, f(x))}{\sum_x \max(0, f(x))}$$

Finally, we calculate the probability of rejection:

$$\mathbb{P}(\tilde{x}\ rejected) = 1 - \mathbb{P}(\tilde{x}\ accepted)$$　**1 – 任意 x 被接收概率**
$$= 1 - \sum_{x'} \mathbb{P}(X = x', \tilde{x}\ accepted)$$
$$= 1 - \sum_{x'} \min(p(x'), q(x'))$$
$$= \sum_{x'} q(x') - \min(p(x'), q(x'))$$
$$= \sum_{x'} \max(0, q(x') - p(x'))$$　**= 候选子分布 概率和**

This is equal to the denominator of $(q(x) - p(x))_+$, so:

$$\mathbb{P}(\tilde{x}\ rejected)\mathbb{P}(X = x|\tilde{x}\ rejected) = \max(0, q(x) - p(x))$$

Hence:

$$\mathbb{P}(X = x)$$
$$= \underbrace{\min(p(x), q(x))}_{首次} + \underbrace{\max(0, q(x) - p(x))}_{二次}$$
$$= q(x)$$

$if\ p(x) > q(x), P(x) = q(x) + 0 = q(x),$
$if\ p(x) \le q(x), P(x) = p(x) + q(x) - p(x) = q(x).$

and we have recovered the desired target.

Speculative Sampling

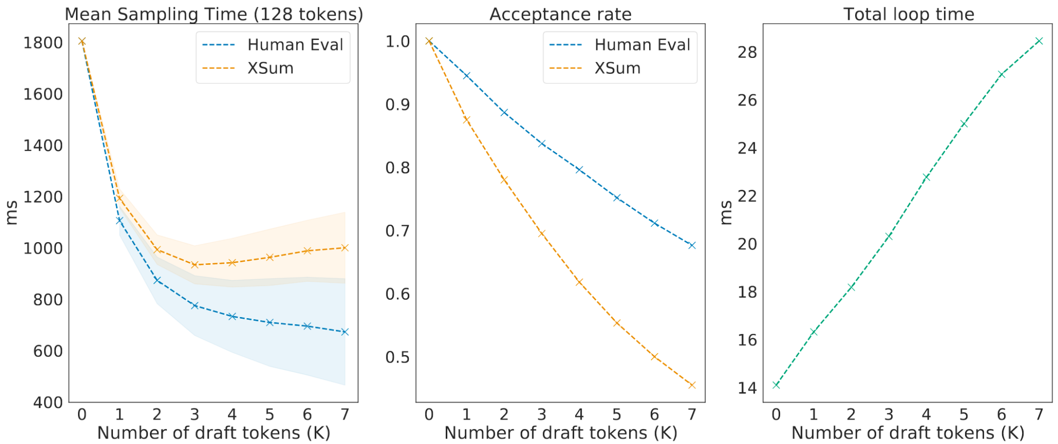
动机： ① Decoding 阶段算力未充分应用 **推理时间 $T_k \approx T_1$**，② LLMs 同构小参数模型与 Target 模型分布接近，更适合做 Draft

贡献： ① 验证了大小模型协同的推测解码的有效性 (**2.5x**)，② 证明 SpS 采样策略与 ArS 采样结果等价

【1】下游任务 生成质量评估

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×

【2】下游任务 K 消融实验



结论：

① OpenTask HumanEval 使用 SpS 采样结果与 ArS 相当，采样策略合理

② 代码任务相较文本摘要，推测未来 tokens 能力更强，K 可以设置更大

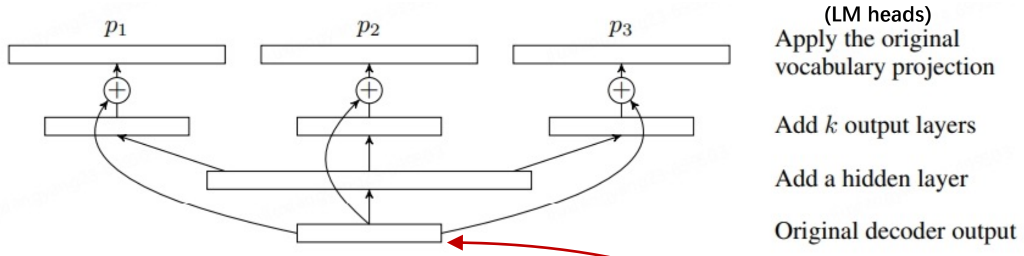
[1] Accelerating Large Language Model Decoding with Speculative Sampling. http://arxiv.org/abs/2302.01318
[2] Fast Inference from Transformers via Speculative Decoding. http://arxiv.org/abs/2211.17192
[3] HuggingFace Assisted Generation. https://huggingface.co/blog/assisted-generation

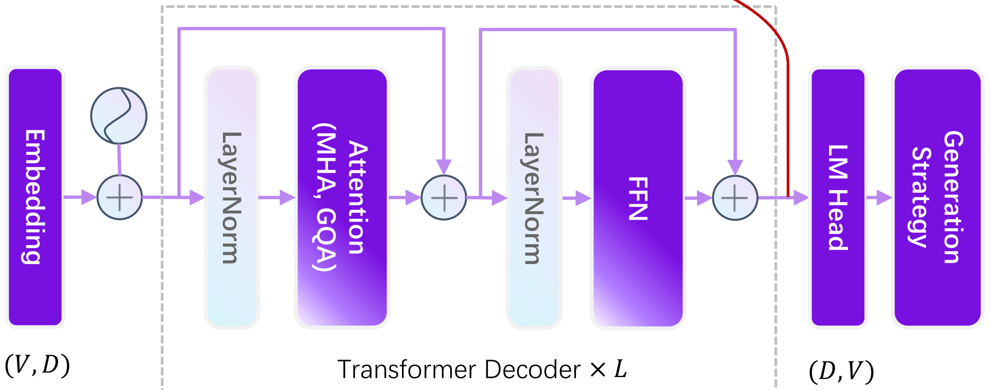
Blockwise Parallel Decoding

Blockwise：A continuation of k draft tokens (词组). 组合式共生模型 “萌芽” 工作.

方法：在 Transformer Decoders 末尾，增加到 k 个并行的 LM heads，并行生成未来多个位置的 tokens，生成长度 m 理想推理次数：$\frac{m}{k}+1$

【1】BPD 并行解码头

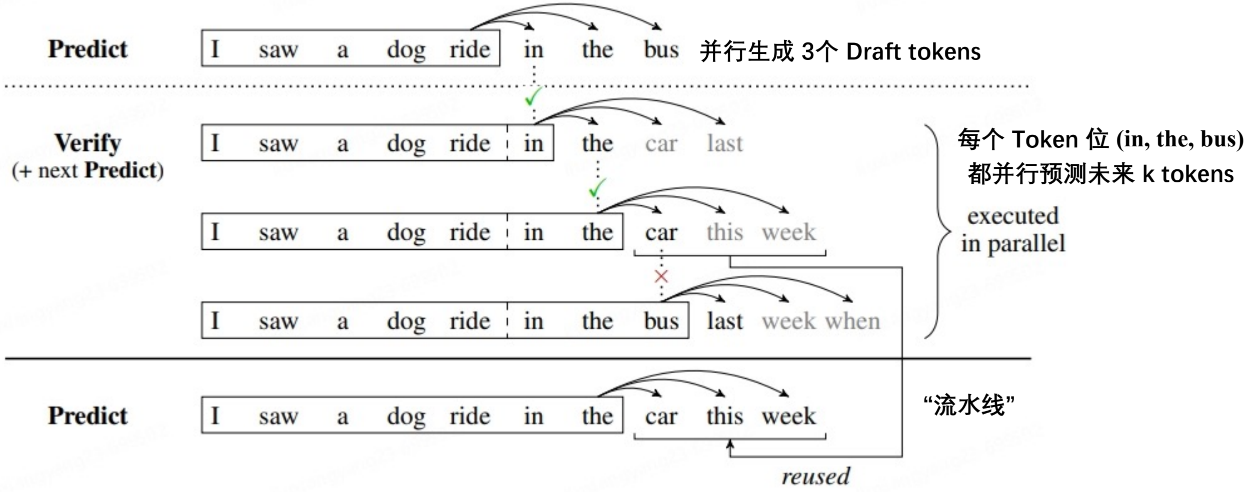




【2】标准 LLM Transformer 结构
(Basemodel, 主要算力模块只过 1 遍)

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

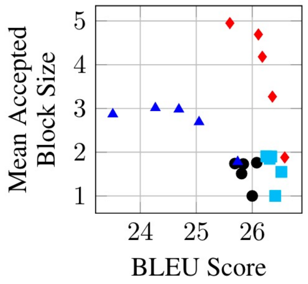
【2】“推测-验证-接受” 过程
(Input: I saw a dog ride)



【4】方法优化以及 k 消融实验
(Task: WMT 2014 English-German translation)

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

(精度, 速度)



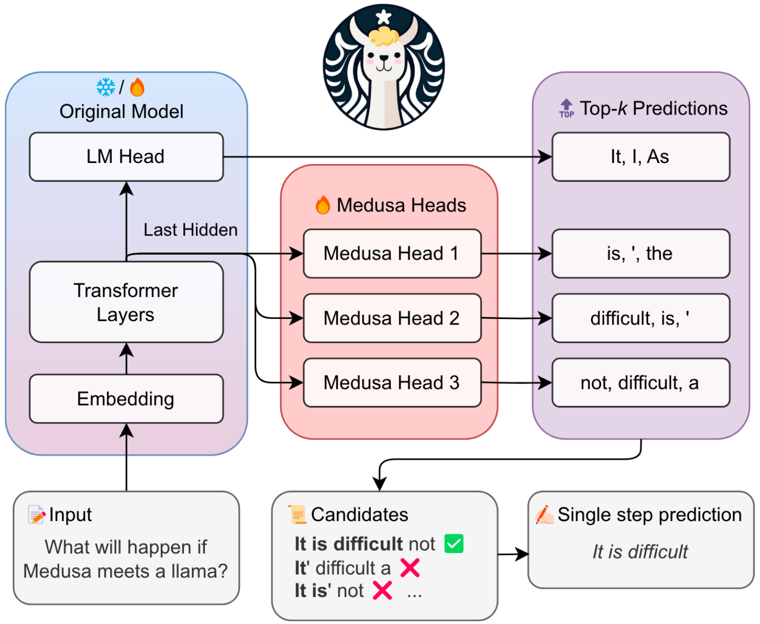
Medusa Parallel Decoding

DataFun. | DA数智大会 Data+AI Conference

动机： ① SpS 分离式大小模型部署复杂，显存成本高，采样效率低，训练 Draft 成本高难以快速适配，② BPD top1 接受率低，但 topk 接受率高

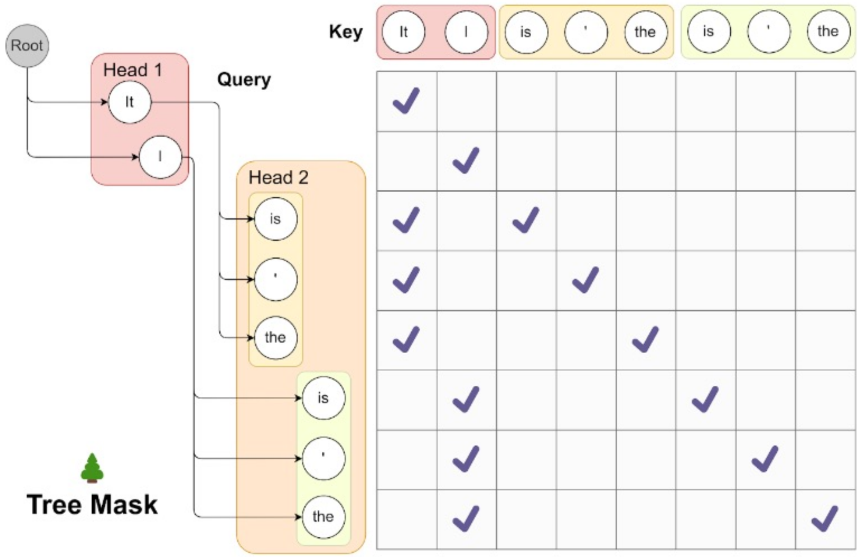
贡献： ① 简洁的组合式共生 Medusa 结构，② Tree Attention 提升接受率，③ Typical Acceptance，④ 推理速度提升 **2.2~3.6x**

【1】Medusa 并行解码头



【2】Tree Attention 并行验证 n 条 draft path

Decoding $(B, 1, D) \rightarrow (B, 1+n, D)$ 充分利用算力



$n = 8 = 2 + 6$

① **Frozen Backbone**

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

② **Joint Training (two-stage is better)**

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

$$\mathcal{L}_{\text{LM-distill}} = KL(p_{\text{original},t}^{(0)} || p_t^{(0)}), \quad \text{(RLHF, w/o training data)}$$

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

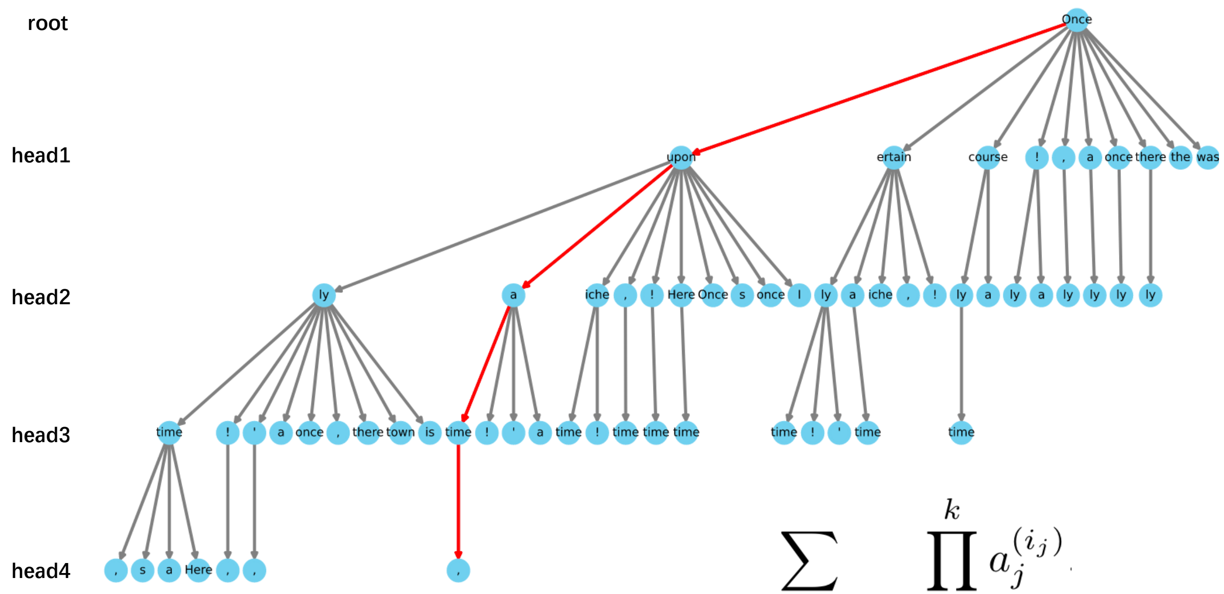
Medusa Parallel Decoding

DataFun. | DA数智大会 Data+AI Conference

动机： ① SpS 分离式大小模型部署复杂，显存成本高，采样效率低，训练 Draft 成本高难以快速适配，② BPD top1 接受率低，但 topk 接受率高

贡献： ① 简洁的组合式共生 Medusa 结构，② Tree Attention 提升接受率，③ Typical Acceptance，④ 推理速度提升 **2.2~3.6x**

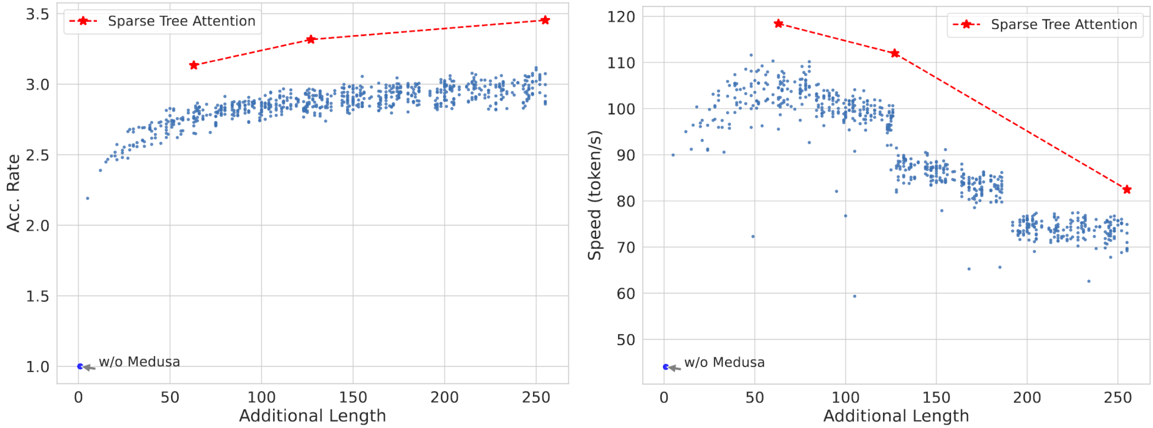
【1】Medusa TopK Tree Paths 剪枝优化



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

i-th top prediction of the k-th head

【2】Trade-off btw Acc.Rate and Speed
(挑选更可能接收的路径)



(a) Acc. Rate vs. Num Tree Paths   (b) Speed vs. Num Tree Paths

蓝色：随机采样 Paths；红色：按左式选择 (64, 128, 256)

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

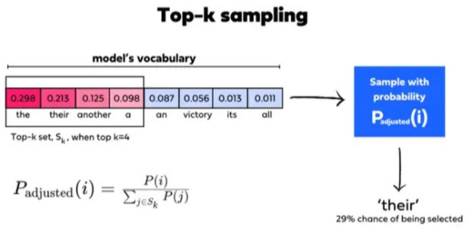
Medusa Parallel Decoding

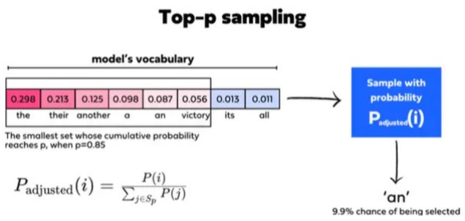
动机：① SpS 分离式大小模型部署复杂，显存成本高，采样效率低，训练 Draft 成本高难以快速适配，② BPD top1 接受率低，但 topk 接受率高

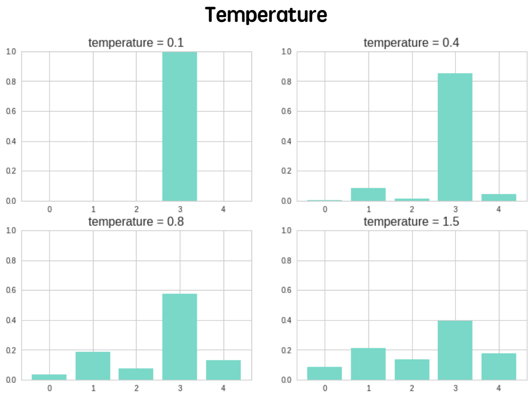
贡献：① 简洁的组合式共生 Medusa 结构，② Tree Attention 提升接受率，③ Typical Acceptance，④ 推理速度提升 **2.2~3.6x**

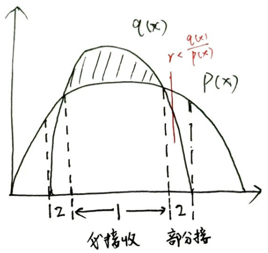
【1】Speculative Sampling 采样的问题

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	$r < \min\left(1, \frac{q_i(\tilde{x}_i)}{p_i(\tilde{x}_i)}\right), r \sim U[0,1]$	$x_{t+c} \sim \text{norm}(\max(0, q_c - p_c))$







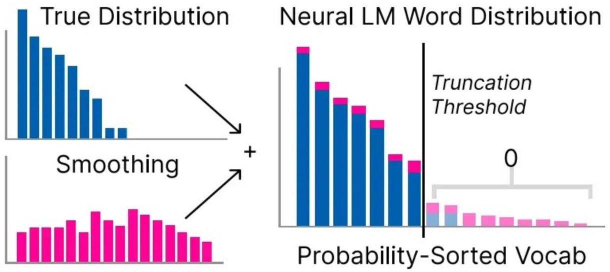


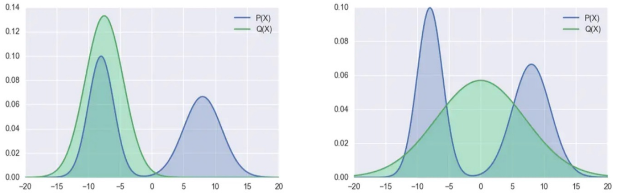
① T 升高，开放式任务 q(x)/p(x) ↓多样性 ↓

② 联合采样验证成本高，两份 probs 速度慢

【2】Typical Acceptance 放松接收条件 (Truncation Sampling)

$$p_{\text{original}}(x_{n+k} | x_1, x_2, \cdots, x_{n+k-1}) > \min\left(\epsilon, \delta \exp\left(-H(p_{\text{original}}(\cdot | x_1, x_2, \cdots, x_{n+k-1}))\right)\right)$$

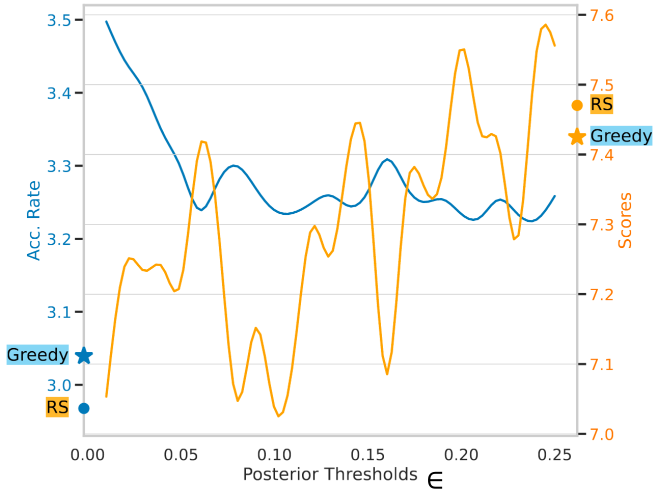




$$\begin{aligned}\eta &= \lambda_{x<i} \cdot Q(X_i | x_{<i}) \\ &= \min(\bar{\lambda} \cdot Q(X_i | x_{<i}), \bar{\lambda}_{x<i} \cdot Q(X_i | x_{<i})) \\ &= \min\left(\frac{\bar{\lambda} \cdot (1+\delta)}{|\mathcal{V}|}, Q'(X_i | x_{<i})\right) \\ &= \min\left(\frac{\bar{\lambda} \cdot (1+\delta)}{|\mathcal{V}|}, \alpha \exp(-h(x_{<i}))\right)\end{aligned}$$

(上下文无关，上下文相关)

$$\eta = \min(\epsilon, \alpha \exp(-h_\theta(x_{<i})))$$



Medusa + 三种采样 (Greedy vs. RS vs. TA)

① TA 相较 Greedy 接收率也提升 10%

② TA 相较 RS 接受率提升明显，选择 0.25 质量与 RS 相当，但接收率从 3.0 提升到 3.5

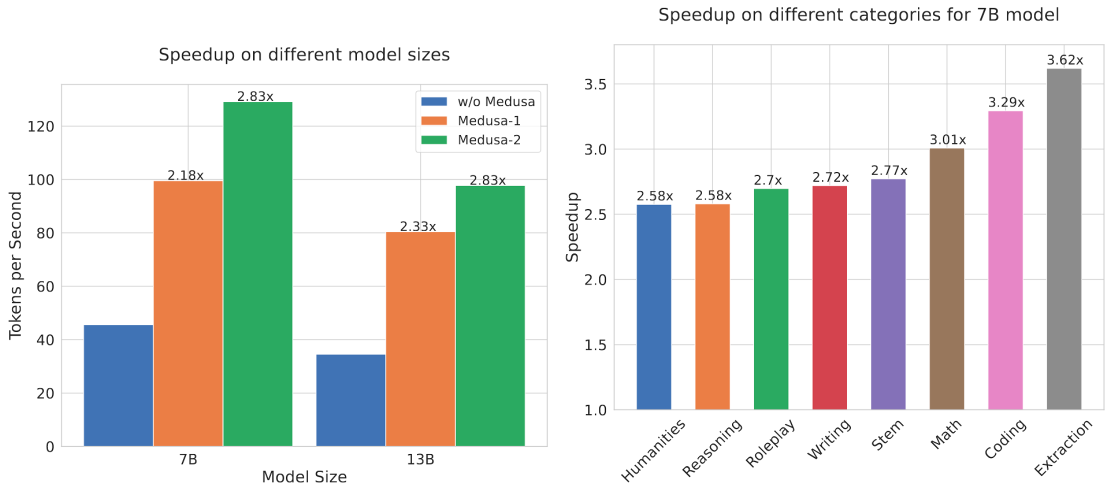
③ TA 采样验证成本低，一份 probs 速度快

[1] Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads. http://arxiv.org/abs/2401.10774
[2] Truncation Sampling as Language Model Desmoothing. http://arxiv.org/abs/2210.15191
[3] 深入LLM投机采样(上) https://waytoagi.feishu.cn/wiki/U1BywrrxTibXOAkgF5Yc2uWynRh

Medusa Parallel Decoding

动机： ① SpS 分离式大小模型部署复杂，显存成本高，采样效率低，训练 Draft 成本高难以快速适配，② BPD top1 接受率低，但 topk 接受率高
贡献： ① 简洁的组合式共生 Medusa 结构，② Tree Attention 提升接受率，③ Typical Acceptance，④ 推理速度提升 **2.2~3.6x**

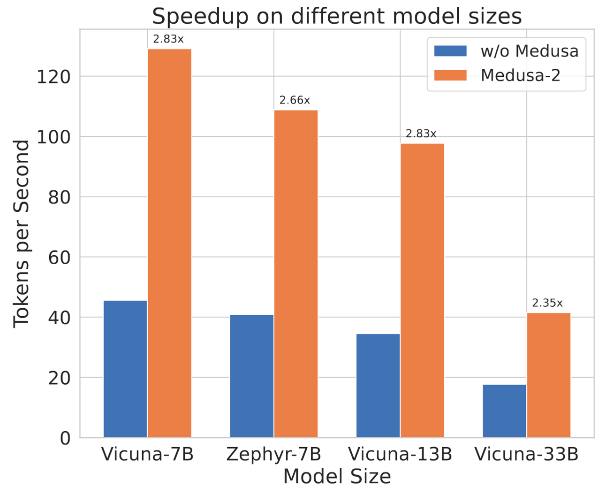
【1】MT-Bench 8类下游任务速度提升
(Vicuna 7B/13B 在 ShareGPT 两阶段 SFT)



(a) (b)

Figure 4: Left: Speed comparison of baseline, MEDUSA-1 and MEDUSA-2 on Vicuna-7B/13B. MEDUSA-1 achieves more than 2× wall-time speedup compared to the baseline implementation while MEDUSA-2 further improves the speedup by a significant margin. Right: Detailed speedup performance of Vicuna-7B on 8 categories from MT-Bench.

【2】自蒸馏训练的有效性
(Zephyr-7B 和 Vicuna-33B 未开放训练数据)



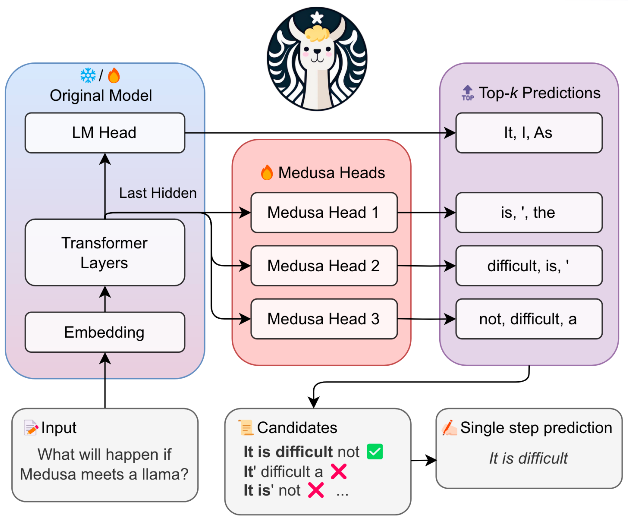
Model Name	Vicuna-7B	Zephyr-7B	Vicuna-13B	Vicuna-33B
Acc. rate	3.47	3.14	3.51	3.01
Overhead	1.22	1.18	1.23	1.27
Quality	6.18 (+0.01)	7.25 (-0.07)	6.43 (-0.14)	7.18 (+0.05)

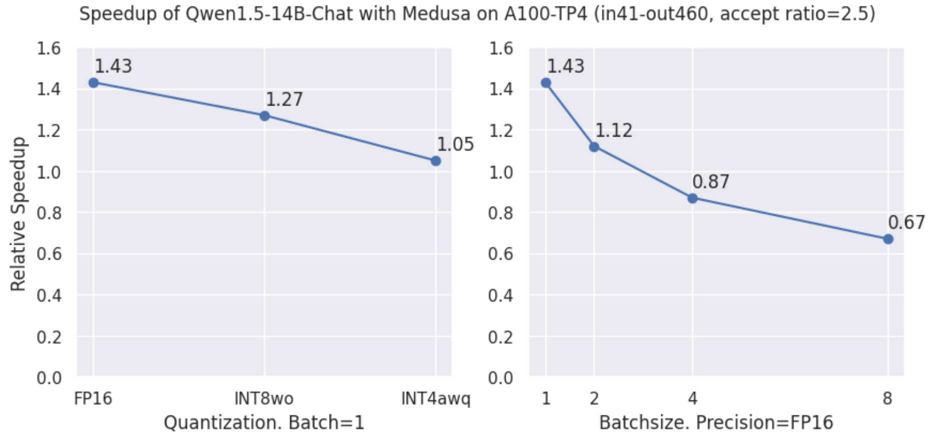
Medusa2 训练后 MT Bench score 基本不变

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

Medusa 实践

- **方法选型**
  - Medusav1：训练资源有限，通用数据集，不接受原模型精度演化；适合第三方模型通用加速
  - Medusav2：训练资源充分，开放数据集，可接受原模型精度变化；适合自有训练模型的团队
- **训练优化 (ChatRhino-14B + 3 Medusa heads)**
  1. 自蒸馏数据集
     - ShareGPT 68k en + 38k cn；vs. 原始数据集 Acc.Rate 1.2x
     - Input Sequence 添加 noise 增加样本多样性
  2. 高效训练
     - Original Model 低 bits 量化加载 GPTQ/AWQ；Heads Acc.Rate 不影响，节省显存
     - Medusa Heads 以 SFT/QLoRA 方式微调均可；Heads Acc.Rate 不影响，后者微调更快 v2 采用
     - Cache Hidden States 省去 Original Model 开销；① 快速尝试不同优化 Tricks ② 支持更大参数如 405B 微调
  3. 蒸馏 Loss
     - Soft Label 对齐 Original Model 分布；vs. Hard Label Acc.Rate 1.04x
  4. Medusa LM Heads 计算优化 (主要计算开销)
     - LM Heads 参数共享：Head 视角没有先后，提升训练效率，节省显存，多 Heads 并行推理；Speedup 1.05x
     - Prune Vocab：Heads 负责高频容易 Tokens，降低 LM Heads 运算量；Speedup 1.04x
- **推理优化**
  1. 重新搜索 TopK Trees；vs. Vicuna Tree Speedup 1.05x
  2. Original Model 采用低 bits 量化部署；Heads Acc.Rate 不影响，但 Speedup 受量化精度影响

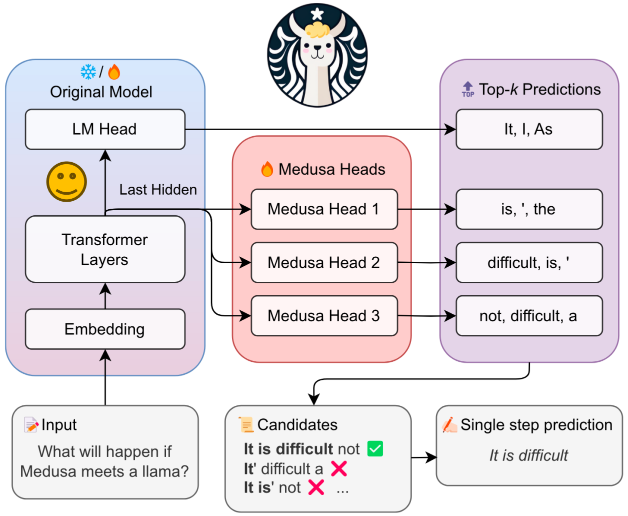




从 FP16 到 INT8 精度，模型显存可节省 50%，推理速度可提升 1.3x；
从 FP16 到 INT4 精度，模型显存可节省 70%，推理速度可提升 1.6x。

Medusa 优化：OmniForce-LLM

动机：Last Hidden 用于 Medusa Heads 输入是否是最佳的？



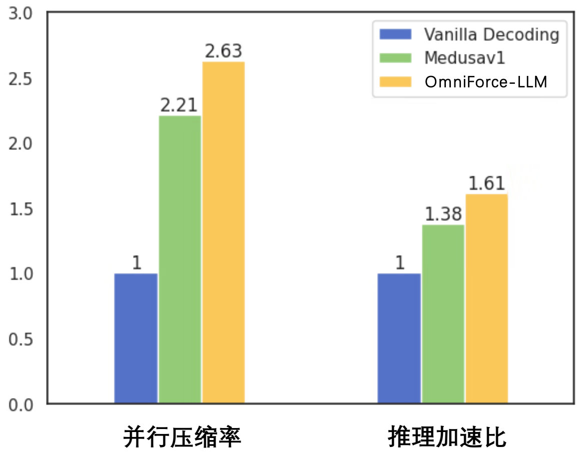
【1】Medusa Fork Decodes 消融实验

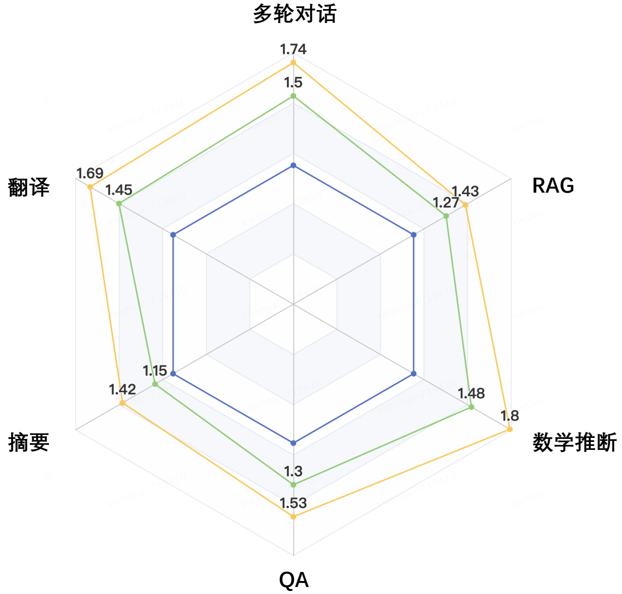
Model	实验组	Medusa Heads Top5 Accuracy (%) 验证集 (2000)					推理性能指标 测试集 (100)			
		head_1	head_2	head_3	head_4	head_5	c_ratio↑	tok (ms)↓	加速比↑	折合比↑
Vicuna-7B	basemodel						1.00	34.51		
	medusa_v1	79.21%	59.39%	46.21%	38.26%	33.37%	2.40	19.07	1.81	75.39%
	medusa_v2	88.90%	78.37%	68.04%	59.08%	50.98%	3.85	12.29	2.81	72.99%
	fork2-decoder2	89.09%	81.08%	73.81%	67.10%	60.01%	3.96	16.05	2.15	54.30%
	fork2-decoder1	87.21%	76.60%	67.20%	58.70%	51.28%	3.45	15.36	2.25	65.15%
	fork1-decoder1	87.14%	76.42%	66.55%	58.14%	50.86%	3.36	15.59	2.21	65.89%
	fork3-decoder1	87.06%	76.44%	66.74%	58.40%	51.13%	3.24	15.98	2.16	66.66%

【2】Spec-Bench 数据集组成

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

【3】ChatRhino-14B 并行解码加速比

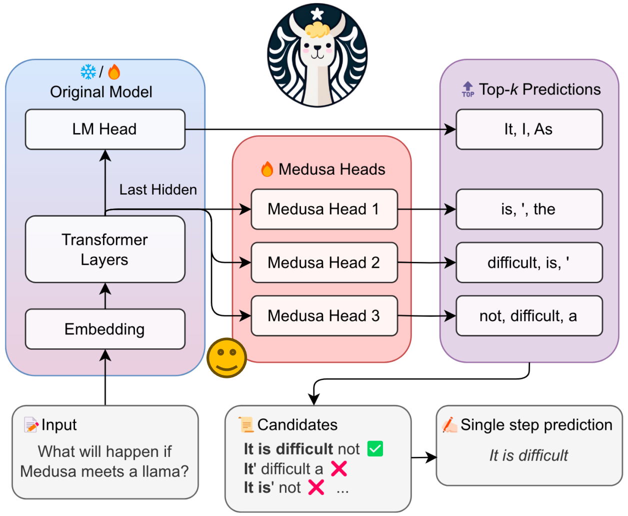




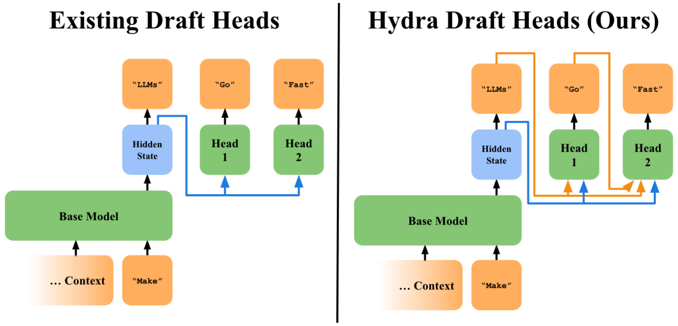
[1] Spec-Bench: A Comprehensive Benchmark and Unified Evaluation Platform for Speculative Decoding. https://sites.google.com/view/spec-bench

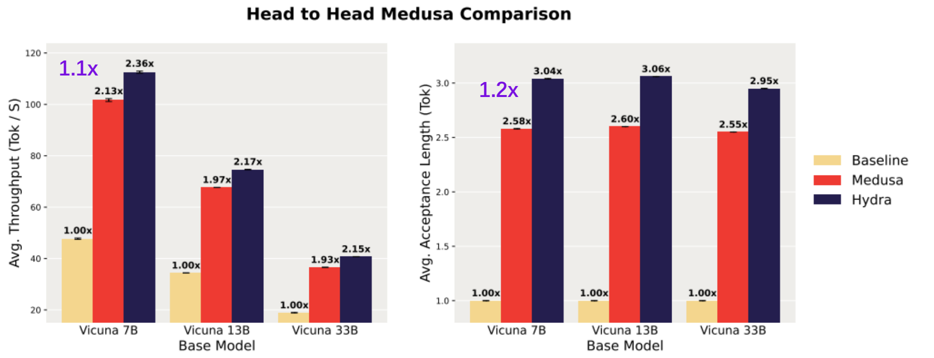
Medusa 优化：Hydra

动机： Medusa Heads 间(低成本)重新引入顺序性是否有帮助?



【1】Hydra Heads 顺序性关联

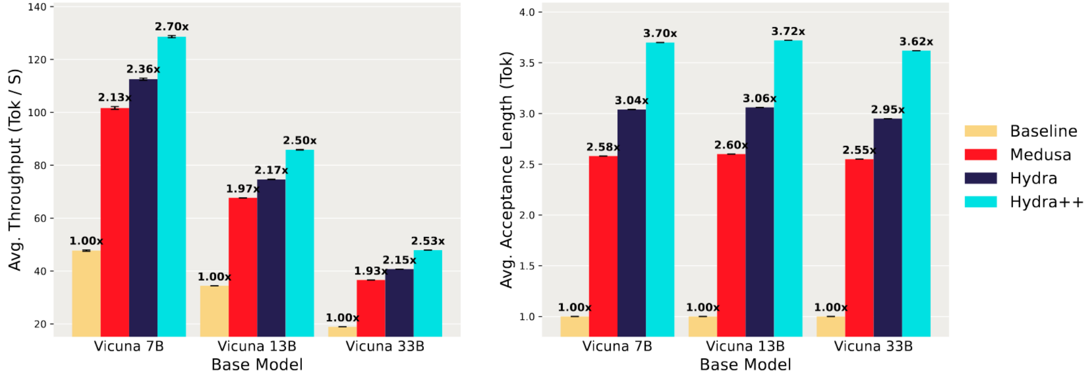


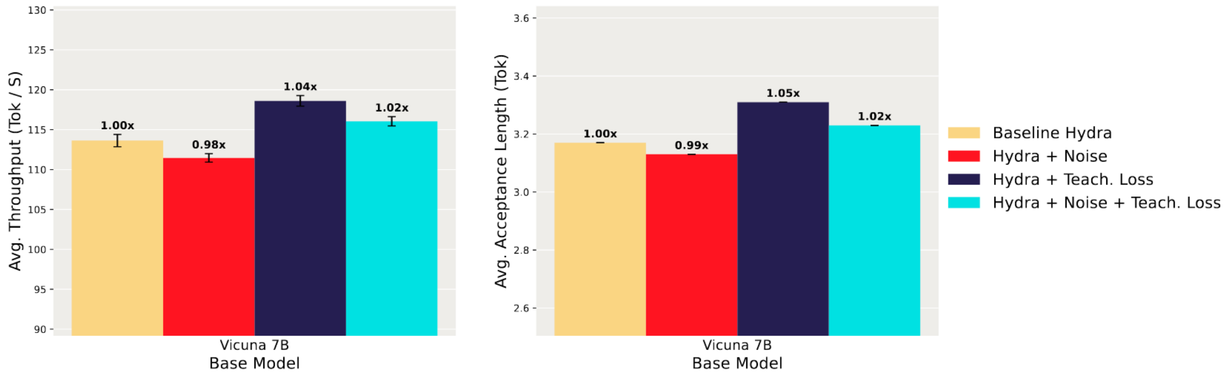


$$p_{\text{draft}}(\hat{x}_{t+i}|x_{\leq t}, \hat{x}_{t+1}, \ldots, \hat{x}_{t+i-1}) = p_{\text{draft}}(\hat{x}_{t+i}|x_{\leq t-1})$$

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

【2】Hydra 加速效果



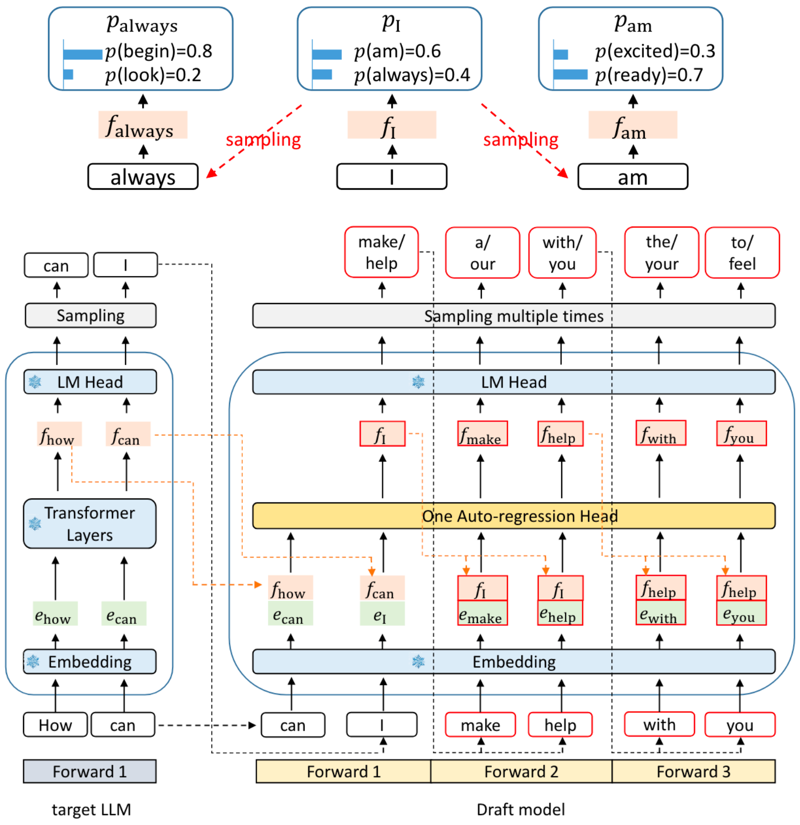


[1] Hydra: Sequentially-Dependent Draft Heads for Medusa Decoding. http://arxiv.org/abs/2402.05109

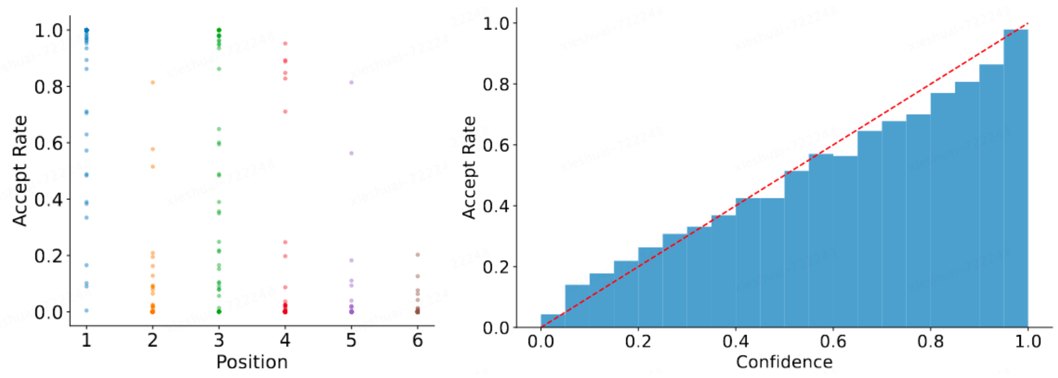
Medusa 优化：EAGLE

动机：① Last Token 可显著降低 Cur Token 预测不确定性，提升 Topk Acc，② 特征自回归更容易，③ TopK Tree 根据 Context 动态剪枝

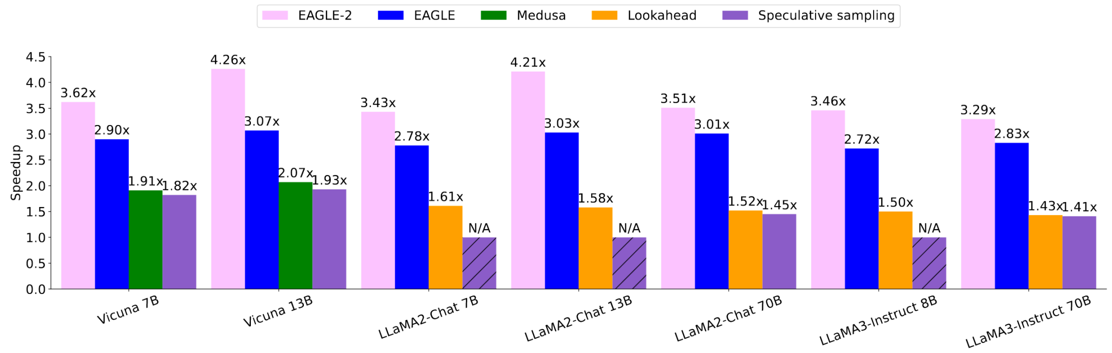
【1】EAGLE Draft Heads

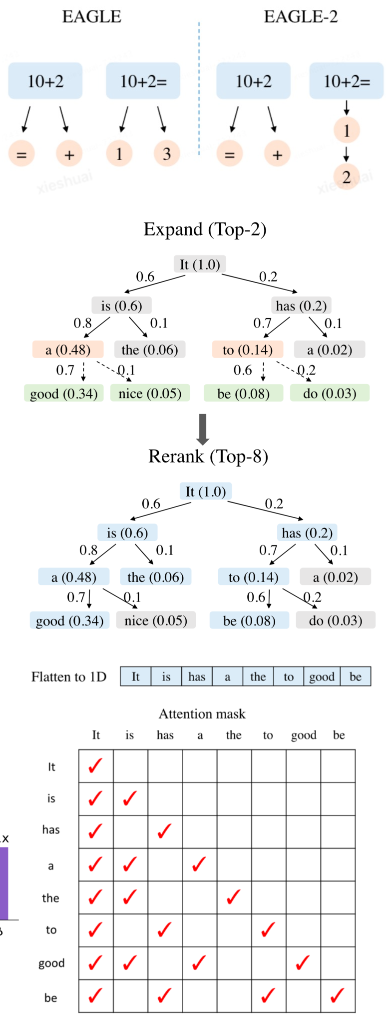


【2】Acc.Rate vs. Position vs. Confidence



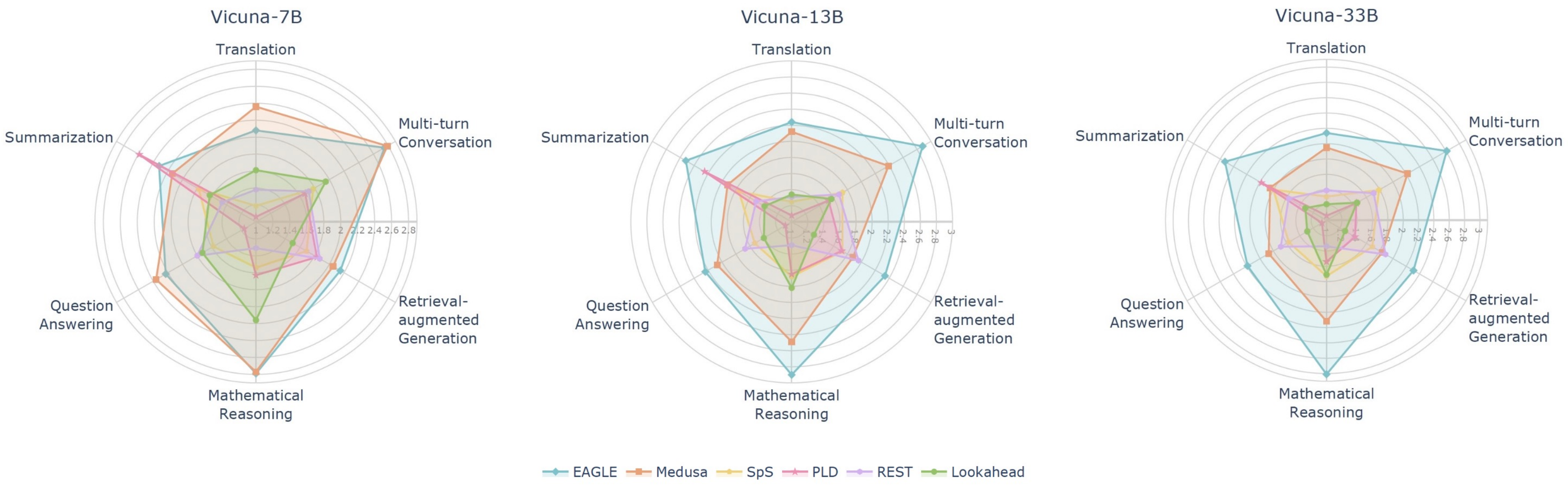
【3】EAGLE v1/v2 加速效果





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

推测解码 Benchmark



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

Summary Of Work

4 未来展望与讨论

- ✓ Medusa
- ✓ 通用推理技术

未来展望与讨论

- **Medusa**
  - Original Model 和 Draft Heads 之间对齐，复用知识蒸馏的小模型对齐经验
  - v2 训推协同 Draft Heads 监督引入大模型原始训练流程 (Pretrain + SFT + RLHF) 原生支持并行解码
  - 长词表 Draft heads 接收率低，分治不同位置 Heads 预测能力与采样对齐
  - 高效的 Draft heads/tokens/continuation 策略，Trade-off btw Acc.Rate and Speed
  - 推理框架算子优化，提升 Medusa 在 Batch 场景下的并行加速能力，与其他加速有效结合
- **通用推理技术**
  - Decode 阶段技术优化已很多，Prefill 阶段大参数模型长文本推理
  - Prefill-Decode 分离式部署，按任务负载特点分别设计优化方法
  - KVCache 高精度量化和中心化管理
  - 超低 bits 大模型量化