



南京大學
NANJING UNIVERSITY

LAMDA
Learning And Mining from Data



Hao AI Lab
UC San Diego

Efficient LLMs via Online Feedback: Training, Inference, and Architecture

Presented by **Yu-Yang Qian**

School of Artificial Intelligence, Nanjing University, China

<https://www.lamda.nju.edu.cn/qianyy/>

2026.03.18

南

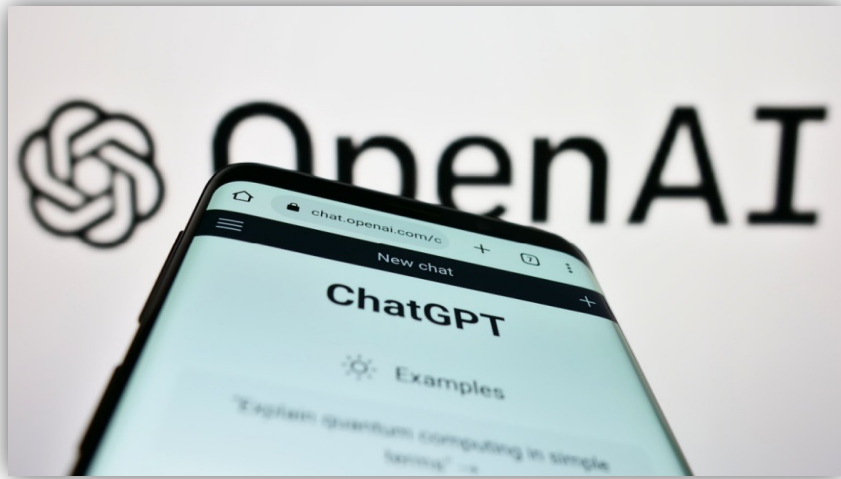
京

大

學

Background

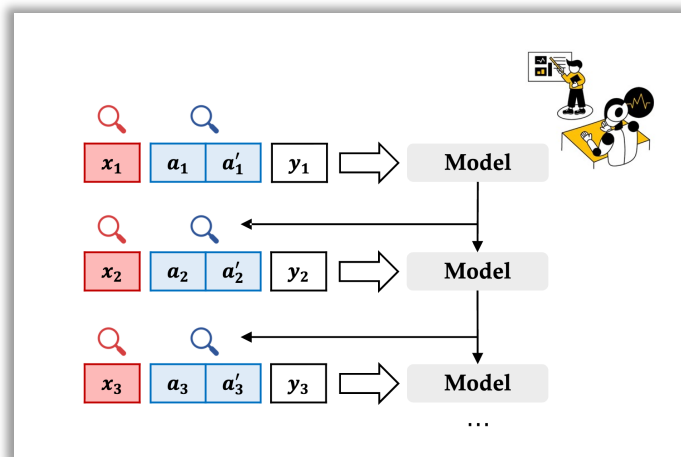
- LLMs have achieved remarkable success across diverse tasks,
 - e.g., *chatbots*, *video generation*, and *agentic tasks*.



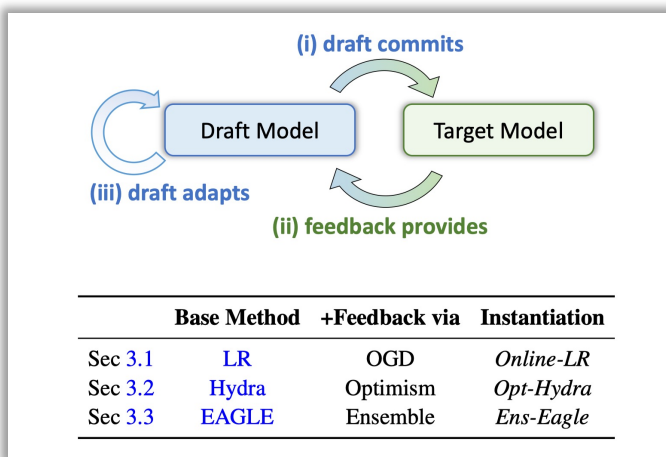
However, the huge parameter size impose *substantial costs* on training & inference

Outline

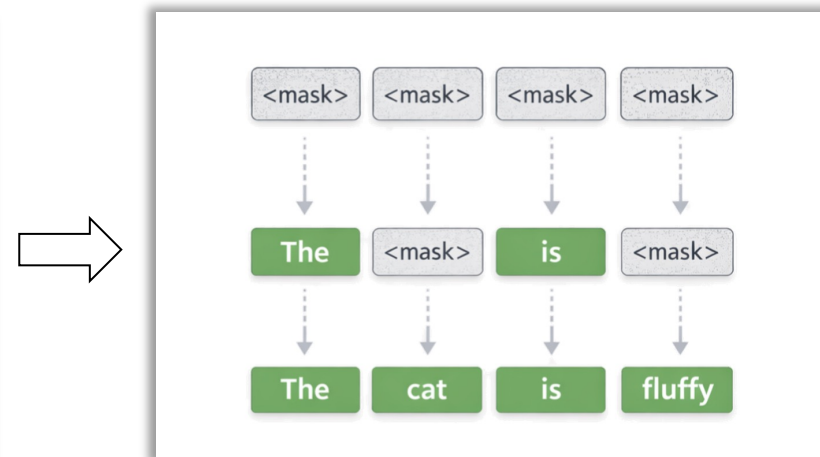
- Goal: *Efficient LLMs via Online Feedback:*



Online (one-pass) *RLHF*
[NeurIPS'25]



Online *Speculative Decoding*
[ICML'26]

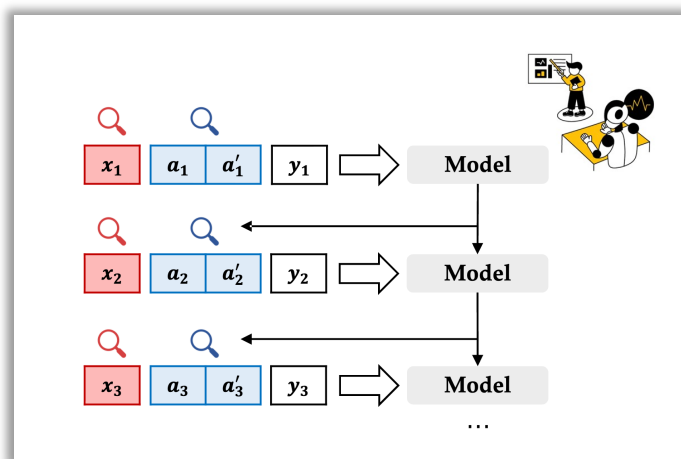


Ultra-Fast *Diffusion LLMs*
[ICML'26]

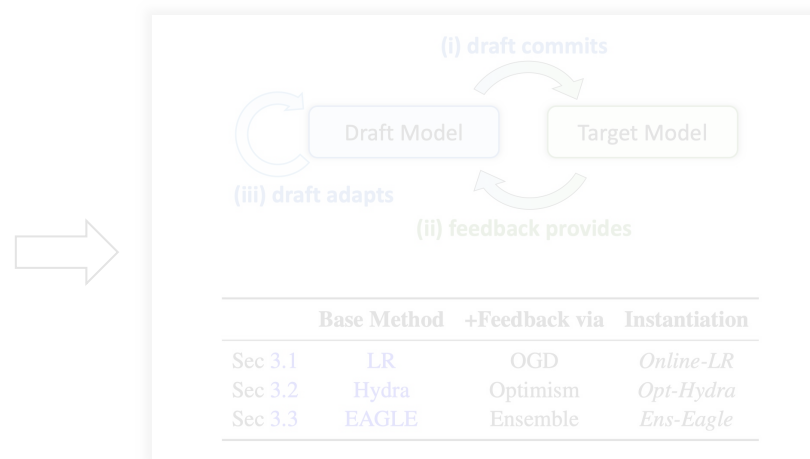
Efficient LLMs across: *training, inference, and architecture.*

Outline: Online *RLHF*

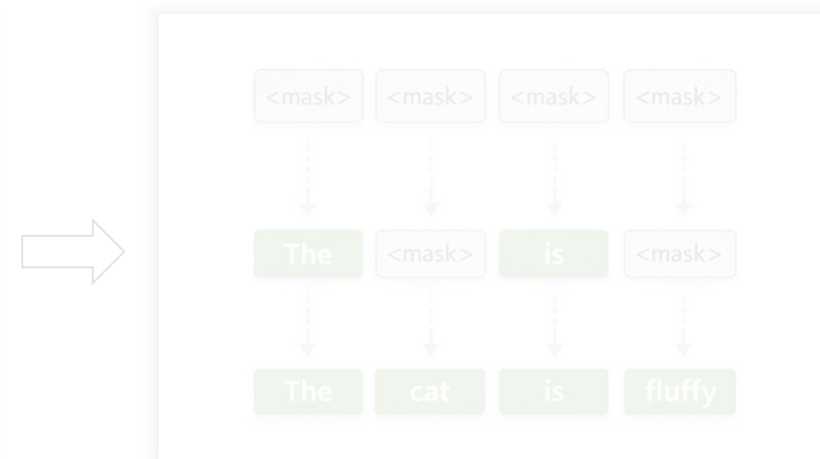
- Goal: Efficiently Online RLHF:



Online (one-pass) *RLHF*
[NeurIPS'25]



Online *Speculative Decoding*
[ICML'26]



Ultra-Fast *Diffusion LLMs*
[ICML'26]

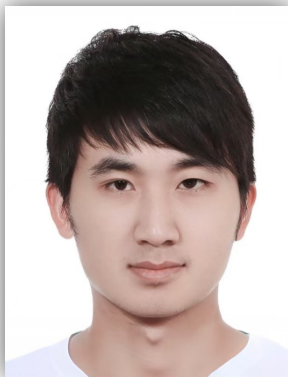
Efficient LLMs across: *training*, *inference*, and *architecture*.

Provably Efficient Online RLHF with *One-Pass* Reward Modeling

Long-Fei Li*, Yu-Yang Qian*, Peng Zhao, Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China,
{lilf, qianyy, zhaop, zhouzh}@lamda.nju.edu.cn

(* indicates equal contribution)



Long-Fei Li*



Yu-Yang Qian*



Peng Zhao



Zhi-Hua Zhou



南京大學
NANJING UNIVERSITY

LAMDA
Learning And Mining from Data

NEURAL INFORMATION
PROCESSING SYSTEMS
SAN DIEGO
[NeurIPS'25]

WarmUp: *RLHF*

- Problem Formulation in RLHF (Reinforcement Learning from Human Feedback):

➤ **Input:** a preference 4-argument tuple (x, a, a', y)

- x : the prompt: "Please write a joke for me."
- a : the first response: "Sorry, I can't."
- a' : the second response: "Here is a joke for you: ..."
- $y \in \{0, 1\}$: the label (human's preference): a'

➤ RLHF wants to use input to improve LLM

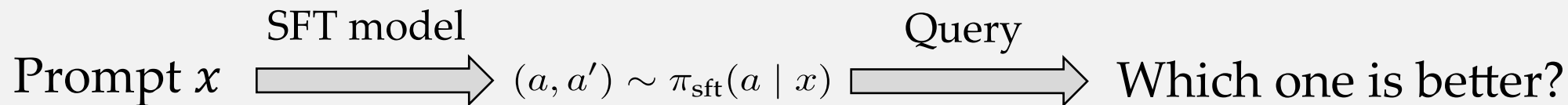
- i.e. *align LLM with human's preference*

➤ **Output:** First train a good *reward model*; then a *fine-tuned LLM*



Previous Method for *RLHF*

- **Step 1:** Sample prompts and queries using SFT model:



- **Step 2:** use *Bradley-Terry Model* (BT Model) for the reward model:

Definition 1 (Bradley-Terry Model). Given a context $x \in \mathcal{X}$ and two actions $a, a' \in \mathcal{A}$, the probability of the human preferring action a over action a' is given by

$$\mathbb{P}(a \succ a' \mid x) = \frac{\exp(r(x, a))}{\exp(r(x, a)) + \exp(r(x, a'))} \quad r \text{ is the latent function.}$$

- **Step 3:** Learning reward model using *Maximum Likelihood Estimation* (MLE):

$$\arg \min_{r_\phi} \mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, a_w, a_l) \sim \mathcal{D}} [\log \sigma(r_\phi(x, a_w) - r_\phi(x, a_l))] \quad \left(\sigma(w) = \frac{1}{1 + e^{-w}} \right)$$

Motivation and Goal: *Efficient* RLHF

- *Computational challenge* of previous MLE-based methods:
 - Need to **store all previous data**, storage expensive;
 - Need to **re-compute all previous data**, costly for online scenarios.

Computational and storage cost at round t is $O(t)$!

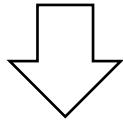
- We aim to take *a unified view* for the *efficient RLHF*:

Goal: **statistically** and **computationally** efficient online RLHF algorithms via *one-pass reward modeling*.

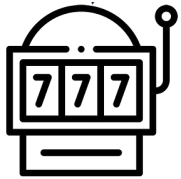
Method: a Unified View from *Contextual Bandits*

- Revisiting RLHF from the perspective of *contextual bandits*:

- prompt x
- actions a, a'
- **Goal:** better reward model $r(\cdot, \cdot)$



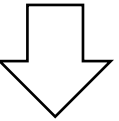
- context x
- arms a, a'
- **Goal:** better model $\tilde{\theta}$



Bradley-Terry (BT) Model

$$\mathbb{P}(a \succ a' \mid x) = \frac{\exp(r(x, a))}{\exp(r(x, a)) + \exp(r(x, a'))}$$

Linear reward model $r(x, a) = \phi(x, a)^\top \theta^*$



Logistic Contextual Bandits

$$\mathbb{P}[y = 1 \mid x, a, a'] = \sigma \left(\phi(x, a)^\top \theta^* - \phi(x, a')^\top \theta^* \right)$$

$\phi(\cdot, \cdot)$ is the known feature mapping

Method: from *MLE* to *OMD*

- Replace MLE with **OMD** inspired by [Zhang and Sugiyama, 2023] :

MLE

$$\hat{\theta}_{k+1,h} = \arg \min_{\theta \in \mathbb{R}^d} \sum_{i=1}^k \sum_{s' \in \mathcal{S}_{i,h}} -y_{i,h}^{s'} \log p_{i,h}^{s'}(\theta) + \frac{\lambda_k}{2} \|\theta\|_2^2$$

$$\underline{\hspace{10em}} \triangleq \ell_{i,h}(\theta)$$

one-pass update by OMD  “lookahead” local norm to keep historical information

implicit OMD

$$\bar{\theta}_{k+1,h} = \arg \min_{\theta \in \Theta} \left\{ \ell_{k,h}(\theta) + \frac{1}{2\eta} \|\theta - \bar{\theta}_{k,h}\|_{\bar{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\bar{\mathcal{H}}_{k,h} \triangleq \sum_{i=1}^{k-1} H_{i,h}(\bar{\theta}_{i+1,h})$ is a particular local norm.

Still no closed-form solution!

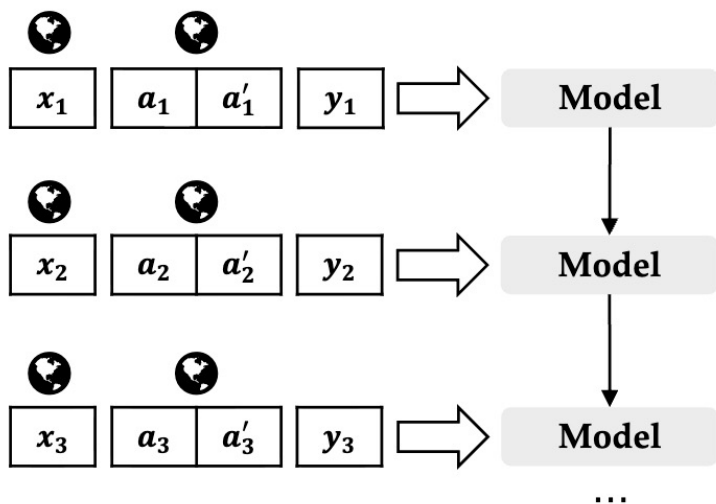
second-order Taylor expansion  $\tilde{\ell}_{k,h}(\theta) = \ell_{k,h}(\tilde{\theta}_{k,h}) + \langle \nabla \ell_{k,h}(\tilde{\theta}_{k,h}), \theta - \tilde{\theta}_{k,h} \rangle + \frac{1}{2} \|\theta - \tilde{\theta}_{k,h}\|_{H_{k,h}(\tilde{\theta}_{k,h})}^2$

standard OMD

$$\tilde{\theta}_{k+1,h} = \arg \min_{\theta \in \Theta} \left\{ \langle \nabla \ell_{k,h}(\tilde{\theta}_{k,h}), \theta - \tilde{\theta}_{k,h} \rangle + \frac{1}{2\eta} \|\theta - \tilde{\theta}_{k,h}\|_{\tilde{\mathcal{H}}_{k,h}}^2 \right\},$$

where $\tilde{\mathcal{H}}_{k,h} = \eta H_{k,h}(\tilde{\theta}_{k,h}) + \sum_{i=1}^{k-1} H_{i,h}(\tilde{\theta}_{i+1,h})$ incorporates additional *second-order quantity*.

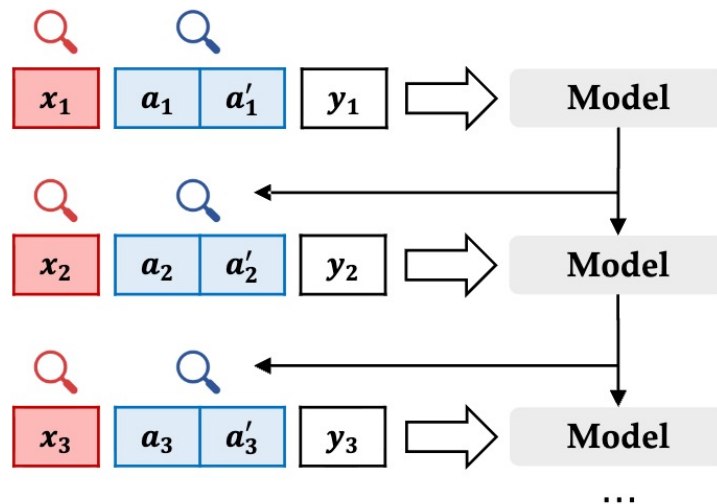
Method: Applied in Three Stages



(a) Passive Data Collection

OMD update:

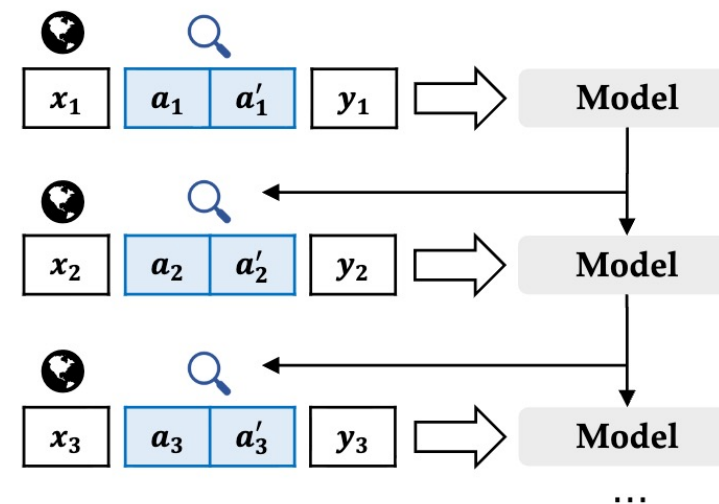
$$\tilde{\theta}_{t+1} = \arg \min_{\theta \in \Theta} \left\{ \langle g_t(\tilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \|\theta - \tilde{\theta}_t\|_{\tilde{\mathcal{H}}_t}^2 \right\}$$



(b) Active Data Collection

Active select samples:

$$(x_t, a_t, a'_t) = \arg \max_{x, a, a' \in \mathcal{X} \times \mathcal{A} \times \mathcal{A}} \left\{ \|\phi(x, a) - \phi(x, a')\|_{\mathcal{H}_t^{-1}} \right\}$$



(c) Deployment-Time Adaptation

Choose dual actions:

$$a_t = \arg \max_{a \in \mathcal{A}} \phi(x_t, a)^\top \tilde{\theta}_t, \text{ and}$$

$$a'_t = \arg \max_{a \in \mathcal{A}} \phi(x_t, a)^\top \tilde{\theta}_t + 2\tilde{\beta} \|\phi(x_t, a) - \phi(x_t, a_t)\|_{\mathcal{H}_t^{-1}}$$

+ OMD update:

$$\tilde{\theta}_{t+1} = \arg \min_{\theta \in \Theta} \left\{ \langle g_t(\tilde{\theta}_t), \theta \rangle + \frac{1}{2\eta} \|\theta - \tilde{\theta}_t\|_{\tilde{\mathcal{H}}_t}^2 \right\}$$

Can be applied to *a variety of* online/offline RLHF scenarios.

Theoretical Improvements

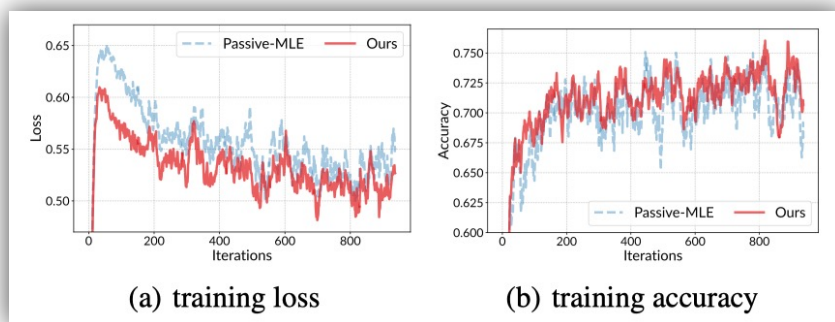
- *Theoretically*: improved *statistical* and *computational* efficiency:

Setting	Context	Action	Gap/Regret	Time	Storage	Reference
Passive			$\tilde{\mathcal{O}}(\sqrt{d} \cdot \kappa \ \Phi\ _{V_T^{-1}})$	$\mathcal{O}(\log T)^*$	$\mathcal{O}(t)$	Zhu et al. [2023]
			$\tilde{\mathcal{O}}(\sqrt{d} \cdot \ \Phi\ _{\mathcal{H}_T^{-1}})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 1)
Active			$\tilde{\mathcal{O}}(d\sqrt{\kappa/T})$	$\mathcal{O}(t \log t)$	$\mathcal{O}(t)$	Das et al. [2024]
			$\tilde{\mathcal{O}}(d\sqrt{\kappa/T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 2)
Deployment			$\tilde{\mathcal{O}}(d\kappa\sqrt{T})$	$\mathcal{O}(t \log t)$	$\mathcal{O}(t)$	Saha et al. [2023]
			$\tilde{\mathcal{O}}(d\sqrt{\kappa T})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Ours (Theorem 3)

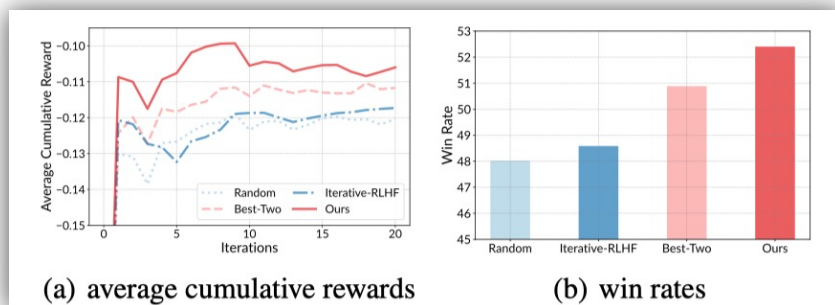
Achieving better regrets with *less* time/storage complexity.

Experiments: LLM's RLHF

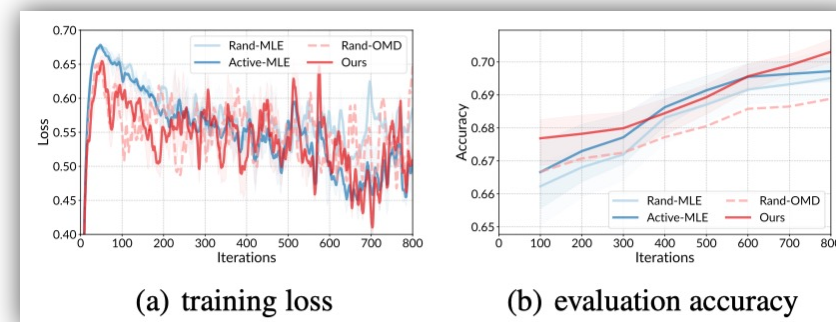
- Foundation Models: *LLaMA 3 8B* & *Qwen 2.5-7B*
- Dataset: *Ultrafeedback* & *Mixture2*



(1) Passive Data Collection



(3) Deployment-time Adaptation

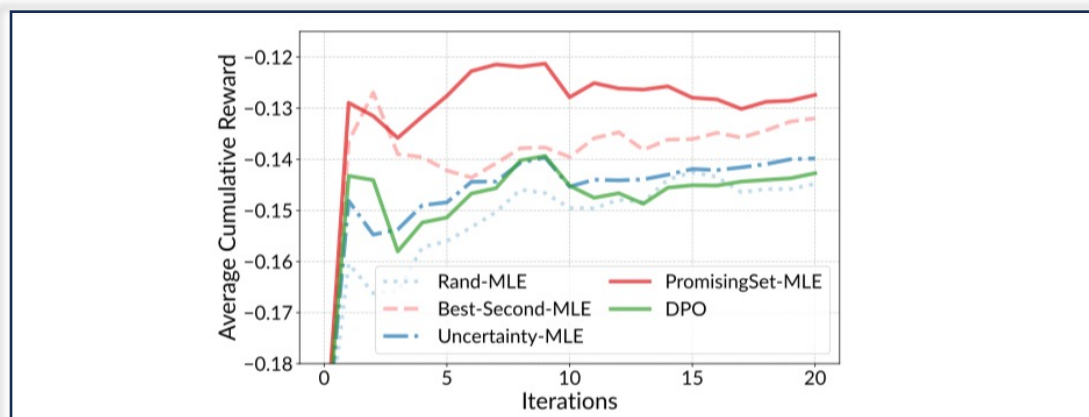


(2) Active Data Collection

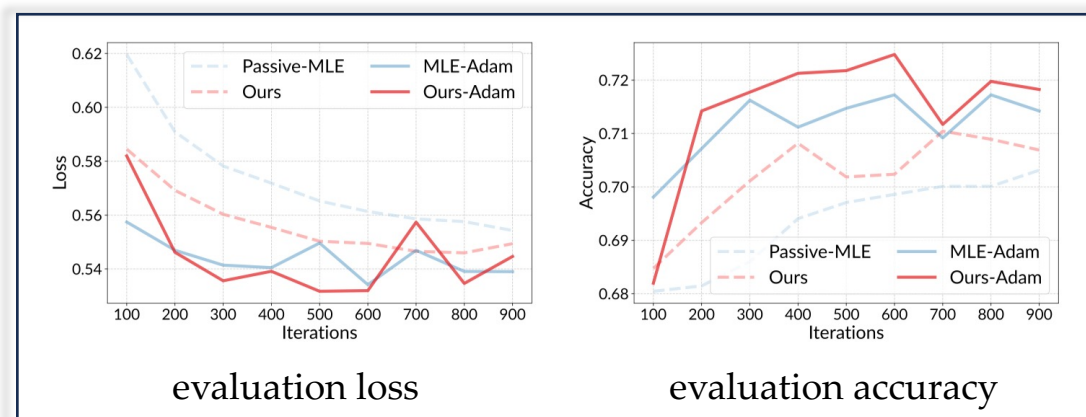
Method	ACC (%)	Time (s)
Rand-MLE	69.51±0.5	4876±47
Active-MLE	69.82±0.4	4982±52
Rand-OMD	68.97±0.6	1456±31
Ours	70.43±0.3	1489±36

(4) Efficiency Comparison

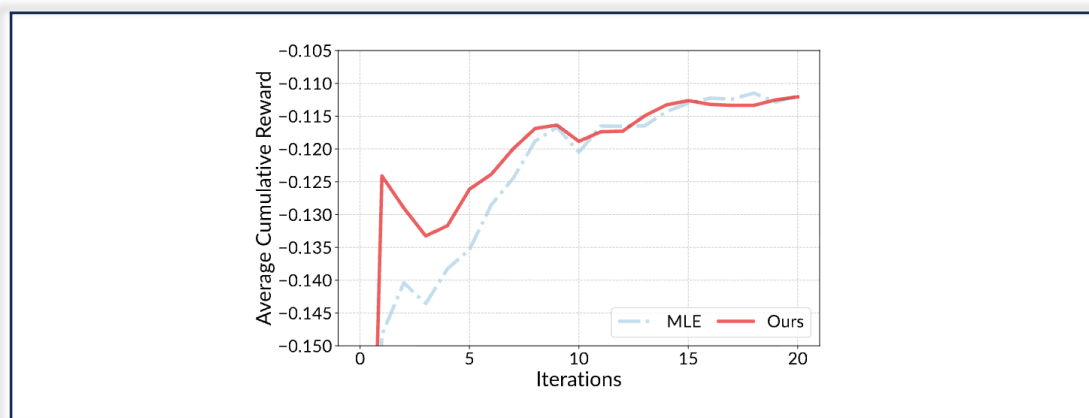
Experiments



Our method surpass *DPO* in deployment stage.



Our method is also comparable with *Adam*.

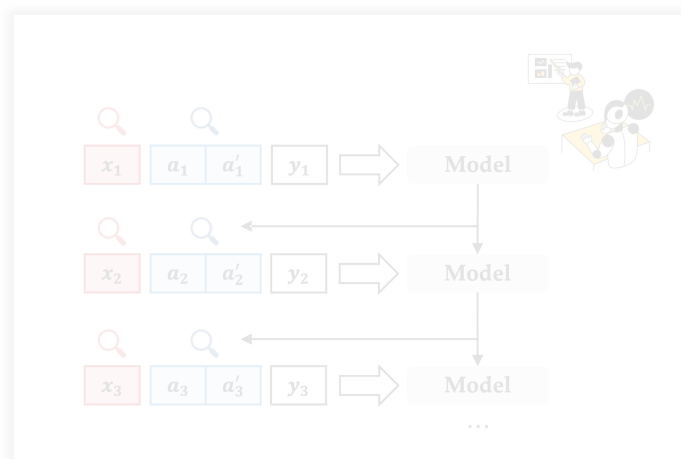


Full parameter update experiment.

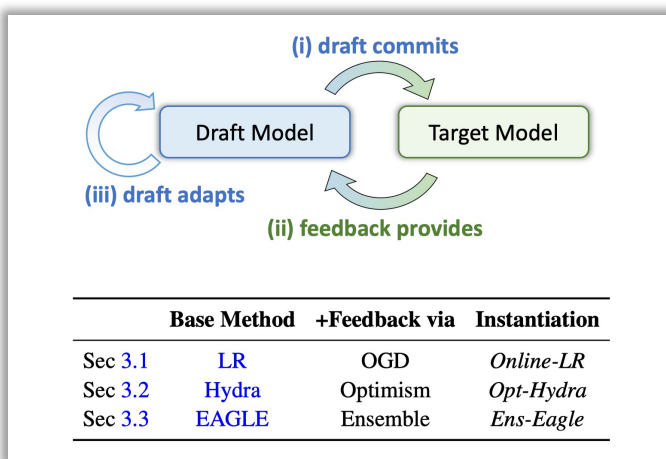
Achieve comparable performance
with *improved efficiency*.

Outline: Online *Speculative Decoding*

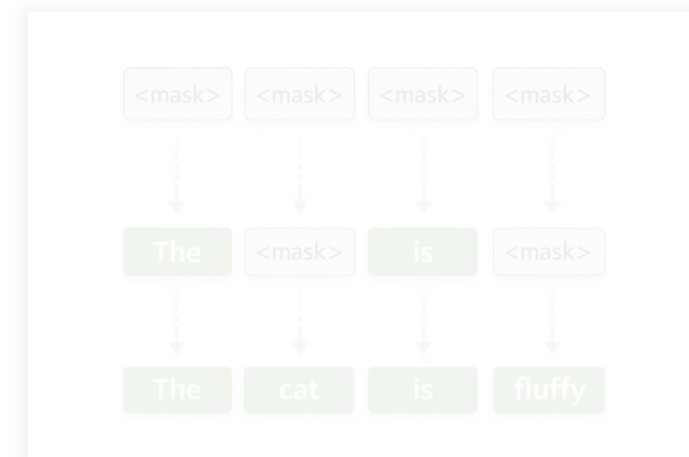
- Goal: Efficient inference through speculative decoding:



Online (one-pass) *RLHF*
[NeurIPS'25]



Online *Speculative Decoding*
[ICML'26]



Ultra-Fast *Diffusion LLMs*
[ICML'26]

Efficient LLMs across: *training*, *inference*, and *architecture*.

WHEN DRAFTS EVOLVE: SPECULATIVE DECODING MEETS ONLINE LEARNING

Yu-Yang Qian^{1,2} Hao-Cong Wu^{1,2} Yichao Fu³ Hao Zhang³ Peng Zhao^{1,2}

¹National Key Laboratory for Novel Software Technology, Nanjing University

²School of Artificial Intelligence, Nanjing University ³University of California, San Diego

{qianyy, zhaop}@lamda.nju.edu.cn hcwu@smail.nju.edu.cn
{yif034, haozhang}@ucsd.edu



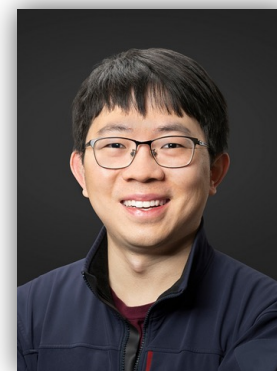
Yu-Yang Qian



Hao-Cong Wu



Yichao Fu



Hao Zhang



Peng Zhao

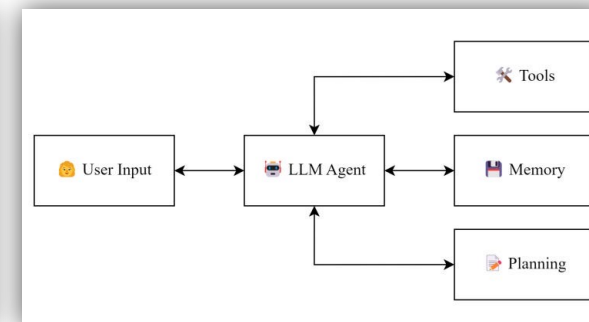
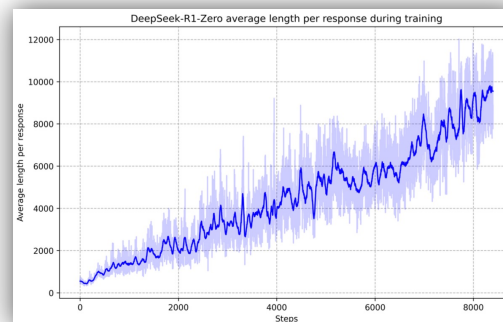


Background: LLM's Inference Cost

- Nowadays, many *LLMs* have been developed and used.
- Not only the model size, but also the *inference cost*, is increasing.
 - Especially with *Chain of Thought* (COT)
 - Recently emerged *agentic LLM* further increase this burden

Model Output

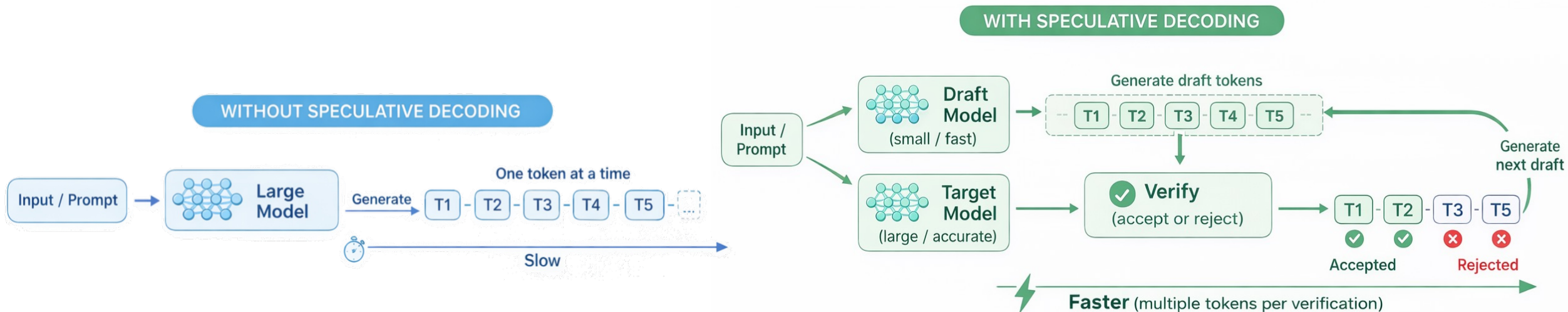
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓



Inference cost and inference latency of LLMs lead to a decline in user experience.

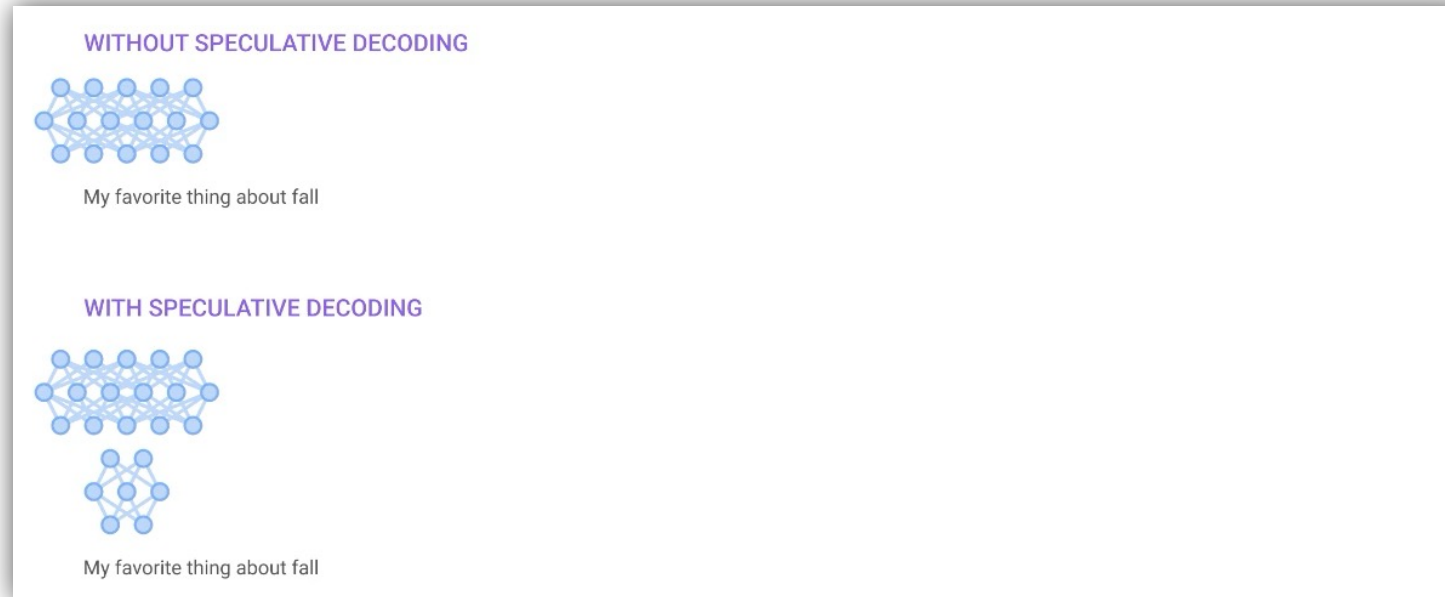
Background

- We investigate the “*generation-refinement*” acceleration methods:
 - e.g., **Speculative Decoding**:
Use a small “*draft model*” to first sequentially generate tokens;
Use a big “*target model*” to parallelly verify the drafts.



Background

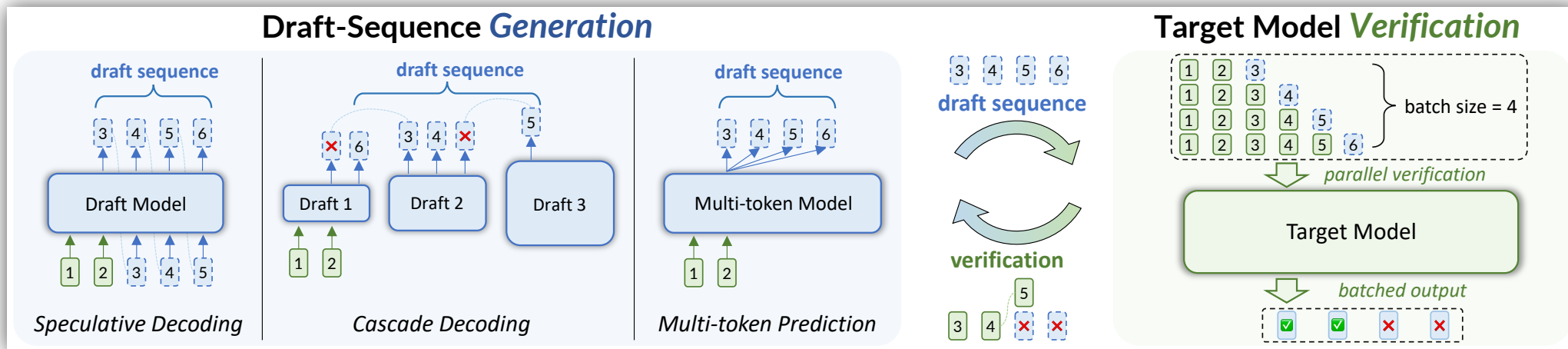
- We investigate the “*generation-refinement*” acceleration methods:
 - e.g., **Speculative Decoding**:
Use a small “*draft model*” to first sequentially generate tokens;
Use a big “*target model*” to parallelly verify the drafts.



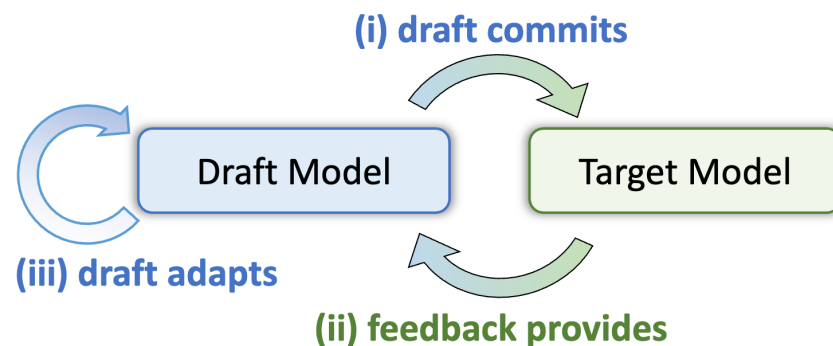
The “*acceptance rate*” of the *draft model* directly determines speedup rate.

Background

- We investigate the “*generation-refinement*” acceleration methods:



- **Key observation:**
 - **Draft model** receives timely feedback **at no cost!** (white box / black box feedback)
 - We can use these feedback to update draft model(s), exactly match **Online Learning**.



Naturally forms a “*draft commits–feedback provides–draft adapts*” **evolving loop**.

A Unified Framework: *OnlineSPEC*

- Formulated as an *online learning problem*:

At each round $t = 1, \dots, T$:

- (1) the learner submits draft model(s) $\mathbf{w}_t \in \mathcal{W}$; (draft model generating)
- ↓
- (2) target model reveals feedback $f_t(\cdot) : \mathcal{W} \mapsto \mathbb{R}$; (parallel verifying)
- ↓
- (3) the learner suffers loss $f_t(\mathbf{w}_t)$ and updates the draft model(s).



- Performance Metric: *Dynamic Regret*

$$\text{Reg}_T \triangleq \sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T f_t(\mathbf{w}_t^*),$$

i.e., the *cumulative gap* between the algorithm and a sequence of time-varying comparators $\{\mathbf{w}_t^*\}_{t=1}^T$.

Algorithm 1 OnlineSPEC Framework

Input: Initial draft model parameter $\mathbf{w}_0 \in \mathcal{W}$ with its predicted distribution $q_{\mathbf{w}_0}(\cdot | \mathbf{x})$, target model $\mathbf{v} \in \mathcal{V}$ with its predicted distribution $p_{\mathbf{v}}(\cdot | \mathbf{x})$, total steps T , candidate length k , input prompt $\mathbf{x}_0$.

Initialize: Input sequence $\mathbf{x} = \mathbf{x}_0$, draft model $\mathbf{w}_t = \mathbf{w}_0$.

for $t = 1, \dots, T$ **do**

 ▷ **Generate the draft sequence:**

for $i = 1$ **to** k **do**

$x_i \sim q_{\mathbf{w}_t}(x | \mathbf{x}_{<i})$, where $\mathbf{x}_{<i} = \{\mathbf{x}, x_1, \dots, x_{i-1}\}$.

 ▷ **Verify by target model:**

 Compute $\{p_{\mathbf{v}}(x_1 | \mathbf{x}_{<1}), \dots, p_{\mathbf{v}}(x_k | \mathbf{x}_{<k})\}$ in parallel.

 ▷ **Determine the accepted token length n_t :**

 Sample $r_1 \sim U(0, 1), \dots, r_k \sim U(0, 1)$ uniformly;

$n_t \leftarrow \min(\{j-1 \mid 1 \leq j \leq k, r_j > \frac{p_{\mathbf{v}}(x_j | \mathbf{x}_{<j})}{q_{\mathbf{w}_t}(x_j | \mathbf{x}_{<j})}\} \cup \{k\})$.

 ▷ **Verify and update the draft sequence:**

if $n_t < k$ **then**

$p'(x) \propto \max(0, p_{\mathbf{v}}(x | \mathbf{x}_{<n_t+1}) - q_{\mathbf{w}_t}(x | \mathbf{x}_{<n_t+1}))$;

else

$p'(x) \leftarrow p_{\mathbf{v}}(x | \mathbf{x}_{<k+1})$.

$x_{n_t+1} \sim p'(x)$; $\mathbf{x} \leftarrow \mathbf{x} + [x_1, \dots, x_{n_t}, x_{n_t+1}]$.

 ▷ **Receive interactive feedback:** observe the loss function $f_t : \mathcal{W} \mapsto \mathbb{R}$ from the target model $\mathbf{v}$.

 ▷ **Update draft model:** $\mathbf{w}_{t+1} \leftarrow \text{Update}(f_t; \mathbf{w}_t)$ as [Sec. 3](#).

Output: Token sequence $\hat{\mathbf{x}} \triangleq [x_1, x_2, \dots]$.

A Unified Framework: *OnlineSPEC*

- Formulated as an *online learning problem*:

At each round $t = 1, \dots, T$:

(1) the learner submits draft model(s) $\mathbf{w}_t \in \mathcal{W}$; (draft model generating)

(2) target model reveals feedback $f_t(\cdot) : \mathcal{W} \mapsto \mathbb{R}$; (parallel verifying)

(3) the learner suffers loss $f_t(\mathbf{w}_t)$ and updates the draft model(s).



- Relationship** between *acceleration rate* and *regret*:

Theorem 1 (Acceleration Rate). Under Assumption 1, for OnlineSpec with a total of T steps, the acceleration rate γ is bounded by

$$\frac{k+1}{(\alpha k+1) \left(1 + (k+1) \sqrt{\text{Reg}_T / (2T)}\right)} \leq \gamma \leq \frac{k+1}{\alpha k+1},$$

where $\alpha \triangleq \frac{a}{A} \ll 1$ is the ratio of the inference times between the draft model and the target model.

Algorithm 1 OnlineSPEC Framework

Input: Initial draft model parameter $\mathbf{w}_0 \in \mathcal{W}$ with its predicted distribution $q_{\mathbf{w}_0}(\cdot | \mathbf{x})$, target model $\mathbf{v} \in \mathcal{V}$ with its predicted distribution $p_{\mathbf{v}}(\cdot | \mathbf{x})$, total steps T , candidate length k , input prompt $\mathbf{x}_0$.

Initialize: Input sequence $\mathbf{x} = \mathbf{x}_0$, draft model $\mathbf{w}_t = \mathbf{w}_0$.

for $t = 1, \dots, T$ **do**

 ▷ **Generate the draft sequence:**

for $i = 1$ **to** k **do**

$x_i \sim q_{\mathbf{w}_t}(x | \mathbf{x}_{<i})$, where $\mathbf{x}_{<i} = \{\mathbf{x}, x_1, \dots, x_{i-1}\}$.

 ▷ **Verify by target model:**

 Compute $\{p_{\mathbf{v}}(x_1 | \mathbf{x}_{<1}), \dots, p_{\mathbf{v}}(x_k | \mathbf{x}_{<k})\}$ in parallel.

 ▷ **Determine the accepted token length n_t :**

 Sample $r_1 \sim U(0, 1), \dots, r_k \sim U(0, 1)$ uniformly;

$n_t \leftarrow \min(\{j-1 \mid 1 \leq j \leq k, r_j > \frac{p_{\mathbf{v}}(x_j | \mathbf{x}_{<j})}{q_{\mathbf{w}_t}(x_j | \mathbf{x}_{<j})}\} \cup \{k\})$.

 ▷ **Verify and update the draft sequence:**

if $n_t < k$ **then**

$p'(x) \propto \max(0, p_{\mathbf{v}}(x | \mathbf{x}_{<n_t+1}) - q_{\mathbf{w}_t}(x | \mathbf{x}_{<n_t+1}))$;

else

$p'(x) \leftarrow p_{\mathbf{v}}(x | \mathbf{x}_{<k+1})$.

$x_{n_t+1} \sim p'(x)$; $\mathbf{x} \leftarrow \mathbf{x} + [x_1, \dots, x_{n_t}, x_{n_t+1}]$.

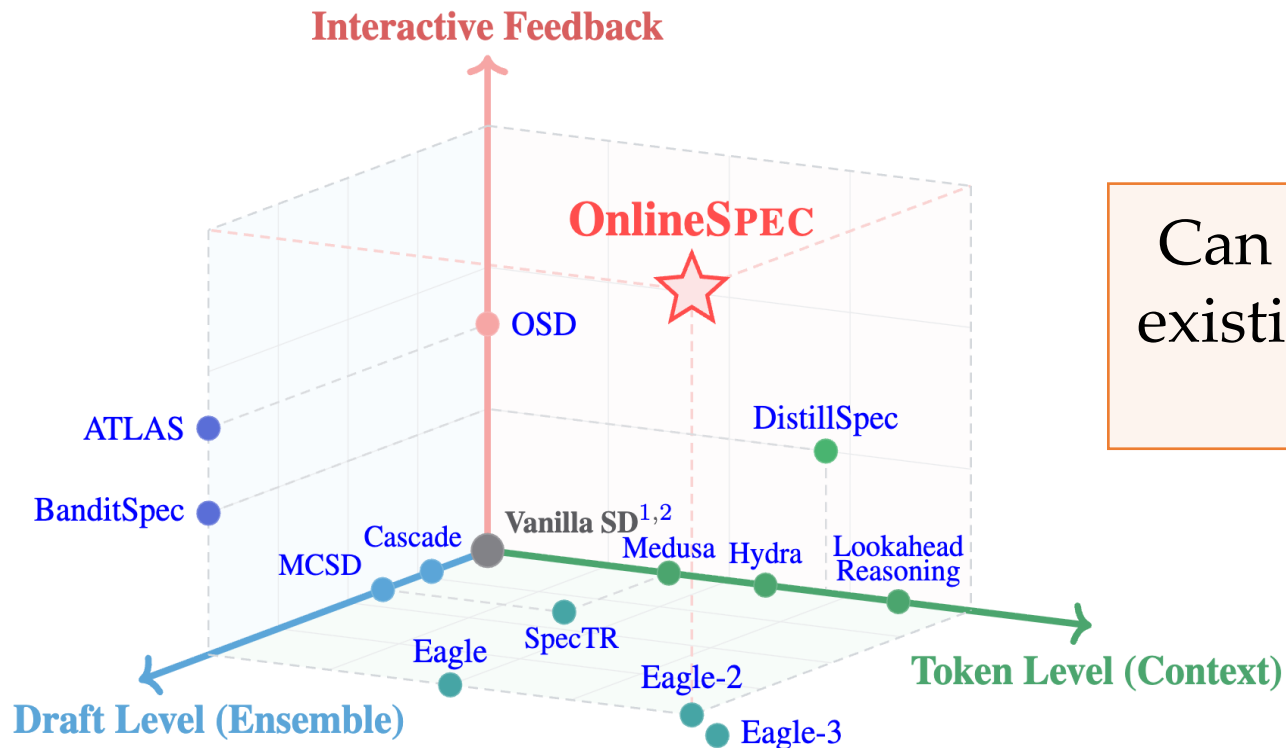
 ▷ **Receive interactive feedback:** observe the loss function $f_t : \mathcal{W} \mapsto \mathbb{R}$ from the target model $\mathbf{v}$.

 ▷ **Update draft model:** $\mathbf{w}_{t+1} \leftarrow \text{Update}(f_t; \mathbf{w}_t)$ as [Sec. 3](#).

Output: Token sequence $\hat{\mathbf{x}} \triangleq [x_1, x_2, \dots]$.

OnlineSPEC with Instantiations

- *OnlineSPEC* framework can guide the design of new algorithms:
 - A *unified framework* to boost speculative decoding
 - By leveraging the rich toolkit from online learning



Can be *seamlessly* combined with existing methods to further enhance the acceleration rate.

OnlineSPEC with Instantiations

- *OnlineSPEC* framework can guide the design of new algorithms:
 - A *unified framework* to boost speculative decoding
 - By leveraging the rich toolkit from online learning
- We demonstrate its generality through three *instantiations*:

Base Method	+Feedback via	Instantiation	Acceleration rate
Lookahead Reasoning	Online gradient descent $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}} [\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)]$	<i>Online-LR</i>	$\gamma = \Omega \left(\frac{1}{\alpha k + 1} \min \left\{ k + 1, \frac{T^{1/4}}{\sqrt{1 + P_T}} \right\} \right)$
Hydra	Optimistic online learning $\mathbf{w}_t = \Pi_{\mathcal{W}} [\hat{\mathbf{w}}_t - \eta \mathbf{h}_t]; \hat{\mathbf{w}}_{t+1} = \Pi_{\mathcal{W}} [\hat{\mathbf{w}}_t - \eta \nabla f_t(\mathbf{w}_t)]$	<i>Opt-Hydra</i>	$\gamma = \Omega \left(\frac{1}{\alpha k + 1} \min \left\{ k + 1, \frac{\sqrt{T}}{(1 + \delta_T)^{1/4} \sqrt{1 + P_T}} \right\} \right)$
Eagle/Eagle-3	Online ensemble learning $\mathbf{w}_t = \sum_{i=1}^N p_t^i \cdot \mathbf{w}_t^i$	<i>Ens-Eagle</i>	$\gamma = \Omega \left(\frac{1}{\alpha k + 1} \min \left\{ k + 1, \frac{T^{1/4}}{(1 + P_T)^{1/4}} \right\} \right)$

Experimental Results

		GSM8K		Spider		Code-Search		Alpaca-Finance	
		AVGLEN $\uparrow$	SPEEDUP $\uparrow$	AVGLEN $\uparrow$	SPEEDUP $\uparrow$	AVGLEN $\uparrow$	SPEEDUP $\uparrow$	AVGLEN $\uparrow$	SPEEDUP $\uparrow$
Target model: <i>lmsys / Vicuna-7B-v1.3</i>									
	Vanilla SD	1.25 $\pm$ 0.01	1.00 $\times$	1.20 $\pm$ 0.03	1.00 $\times$	1.22 $\pm$ 0.02	1.00 $\times$	1.20 $\pm$ 0.01	1.00 $\times$
	OSD	1.52$\pm$0.03	1.23$\times$	1.36$\pm$0.02	1.12$\times$	1.37$\pm$0.02	1.09$\times$	1.30$\pm$0.01	1.08$\times$
	Hydra	2.14 $\pm$ 0.05	1.00 $\times$	2.65 $\pm$ 0.31	1.00 $\times$	1.82 $\pm$ 0.04	1.00 $\times$	1.78 $\pm$ 0.01	1.00 $\times$
	OSD-Hydra	2.56 $\pm$ 0.06	1.19 $\times$	3.11 $\pm$ 0.31	1.11 $\times$	2.11 $\pm$ 0.05	1.16 $\times$	2.19 $\pm$ 0.03	1.25 $\times$
(OnlineSPEC)	Opt-Hydra	2.69$\pm$0.08	1.26$\times$	3.27$\pm$0.32	1.18$\times$	2.37$\pm$0.07	1.31$\times$	2.70$\pm$0.03	1.55$\times$
	EAGLE	1.48 $\pm$ 0.03	1.00 $\times$	1.31 $\pm$ 0.03	1.00 $\times$	1.41 $\pm$ 0.03	1.00 $\times$	1.39 $\pm$ 0.01	1.00 $\times$
	OSD-EAGLE	1.92 $\pm$ 0.04	1.28 $\times$	1.53 $\pm$ 0.03	1.21 $\times$	1.54 $\pm$ 0.04	1.07 $\times$	1.51 $\pm$ 0.02	1.09 $\times$
(OnlineSPEC)	Ens-EAGLE	2.01$\pm$0.05	1.41$\times$	1.60$\pm$0.03	1.32$\times$	1.58$\pm$0.04	1.15$\times$	1.61$\pm$0.04	1.14$\times$
	EAGLE-3	1.78 $\pm$ 0.03	1.00 $\times$	1.85 $\pm$ 0.07	1.00 $\times$	1.67 $\pm$ 0.04	1.00 $\times$	1.62 $\pm$ 0.01	1.00 $\times$
	OSD-EAGLE-3	2.07 $\pm$ 0.04	1.20 $\times$	2.10 $\pm$ 0.05	1.07 $\times$	1.97 $\pm$ 0.04	1.13 $\times$	2.00 $\pm$ 0.04	1.16 $\times$
(OnlineSPEC)	Ens-EAGLE-3	2.16$\pm$0.03	1.26$\times$	2.24$\pm$0.07	1.12$\times$	2.01$\pm$0.07	1.18$\times$	2.07$\pm$0.04	1.27$\times$
Target model: <i>meta-llama / Llama-2-7B-Chat</i>									
	Vanilla SD	1.25 $\pm$ 0.01	1.00 $\times$	1.07 $\pm$ 0.01	1.00 $\times$	1.09 $\pm$ 0.01	1.00 $\times$	1.20 $\pm$ 0.01	1.00 $\times$
	OSD	1.40$\pm$0.03	1.06$\times$	1.47$\pm$0.05	1.22$\times$	1.34$\pm$0.02	1.26$\times$	1.31$\pm$0.01	1.12$\times$
	Hydra	1.52 $\pm$ 0.02	1.00 $\times$	1.48 $\pm$ 0.08	1.00 $\times$	1.66 $\pm$ 0.01	1.00 $\times$	1.64 $\pm$ 0.01	1.00 $\times$
	OSD-Hydra	2.55 $\pm$ 0.03	1.67 $\times$	3.10 $\pm$ 0.18	1.91 $\times$	2.39 $\pm$ 0.04	1.43 $\times$	2.25 $\pm$ 0.03	1.36 $\times$
(OnlineSPEC)	Opt-Hydra	2.94$\pm$0.03	1.90$\times$	3.28$\pm$0.16	2.03$\times$	2.71$\pm$0.05	1.61$\times$	2.78$\pm$0.05	1.68$\times$
	EAGLE	1.19 $\pm$ 0.01	1.00 $\times$	1.15 $\pm$ 0.03	1.00 $\times$	1.17 $\pm$ 0.01	1.00 $\times$	1.14 $\pm$ 0.01	1.00 $\times$
	OSD-EAGLE	1.45 $\pm$ 0.01	1.18 $\times$	1.72 $\pm$ 0.06	1.46 $\times$	1.47 $\pm$ 0.02	1.28 $\times$	1.43 $\pm$ 0.01	1.20 $\times$
(OnlineSPEC)	Ens-EAGLE	1.58$\pm$0.01	1.33$\times$	1.82$\pm$0.08	1.61$\times$	1.62$\pm$0.01	1.46$\times$	1.56$\pm$0.02	1.41$\times$
	EAGLE-3	1.82 $\pm$ 0.03	1.00 $\times$	1.61 $\pm$ 0.03	1.00 $\times$	1.69 $\pm$ 0.04	1.00 $\times$	1.70 $\pm$ 0.02	1.00 $\times$
	OSD-EAGLE-3	2.25 $\pm$ 0.02	1.23 $\times$	2.42 $\pm$ 0.12	1.43 $\times$	2.09 $\pm$ 0.10	1.27 $\times$	2.13 $\pm$ 0.02	1.23 $\times$
(OnlineSPEC)	Ens-EAGLE-3	2.33$\pm$0.02	1.27$\times$	2.54$\pm$0.14	1.57$\times$	2.18$\pm$0.10	1.33$\times$	2.15$\pm$0.03	1.26$\times$



Up to **24%** *speedup* over previous SOTA methods.

Experimental Results

	GSM8K		MBPP		MATH		MMLU	
	AVGLen $\uparrow$	ACC (%) $\uparrow$	AVGLen $\uparrow$	ACC (%) $\uparrow$	AVGLen $\uparrow$	ACC (%) $\uparrow$	AVGLen $\uparrow$	ACC (%) $\uparrow$
Target model: <i>Qwen / Qwen3-8B</i> , Draft model: <i>Qwen / Qwen3-0.6B-Base</i>								
Target Model	1.00 (1.00 $\times$)	94.32	1.00 (1.00 $\times$)	53.56	1.00 (1.00 $\times$)	91.54	1.00 (1.00 $\times$)	83.81
Draft Model	1.00 (3.60 $\times$)	53.84	1.00 (3.56 $\times$)	14.15	1.00 (3.54 $\times$)	60.66	1.00 (3.56 $\times$)	55.71
LR	13.25 (1.26 $\times$)	91.04	5.76 (1.09 $\times$)	50.65	9.56 (1.20 $\times$)	92.84	9.40 (1.21 $\times$)	82.86
OSD-LR	12.57 (1.20 $\times$)	93.84	5.66 (1.09 $\times$)	50.54	6.21 (1.10 $\times$)	89.87	6.67 (1.14 $\times$)	82.14
(OnlineSPEC) Online-LR	14.71 (1.41$\times$)	92.88	7.16 (1.14$\times$)	51.19	10.63 (1.24$\times$)	91.37	10.62 (1.26$\times$)	84.52

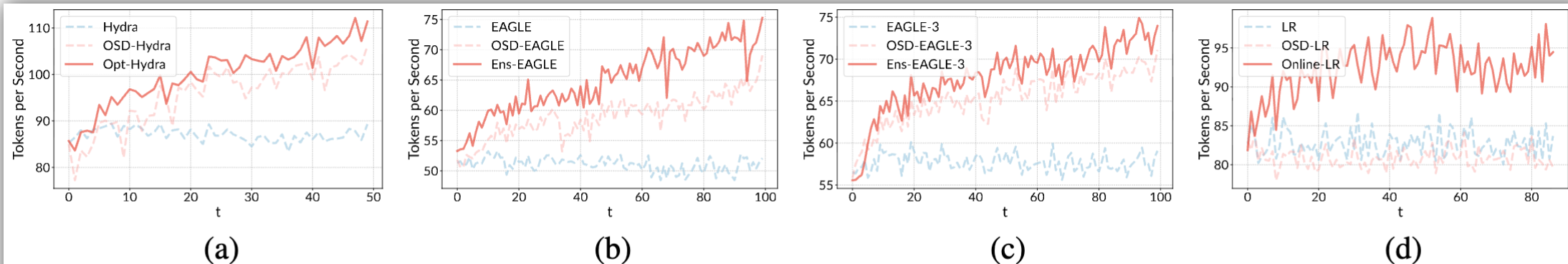


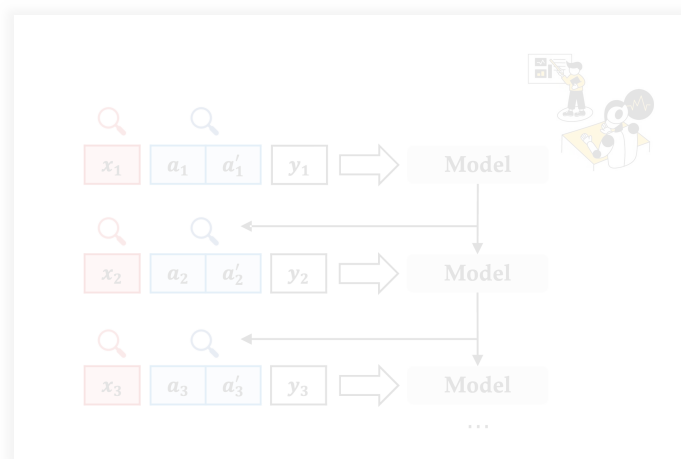
Figure 3. Evolution of tokens per second (TPS) on the *GSM8K* dataset: (a) *Opt-Hydra* with *lmsys/Vicuna-7B-v1.3*, (b) *Ens-EAGLE* with *lmsys/Vicuna-7B-v1.3*, (c) *Ens-EAGLE-3* with *lmsys/Vicuna-7B-v1.3*, and (d) *Online-LR* with *Qwen/Qwen3-8B*. This demonstrates consistent performance improvements via online learning, validating the effectiveness of our *ONLINESPEC* during deployment.



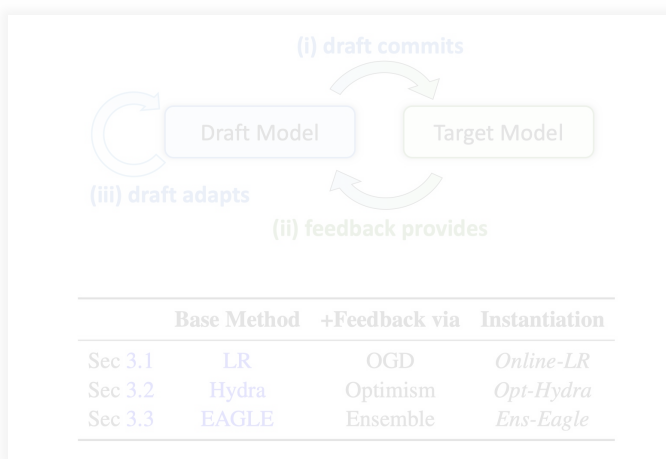
Continuously improving the draft model and inference efficiency.

Outline: *Diffusion* LLMs

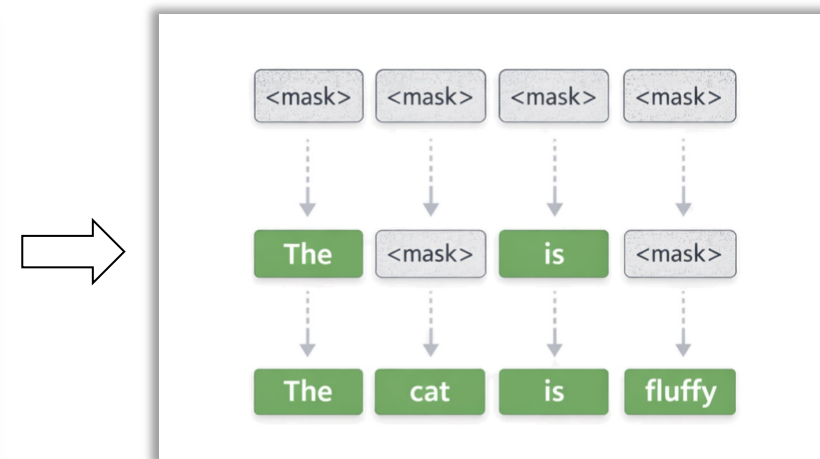
- Goal: Diffusion large language models (*dLLMs*)



Online (one-pass) *RLHF*
[NeurIPS'25]



Online *Speculative Decoding*
[ICML'26]



Ultra-Fast *Diffusion LLMs*
[ICML'26]

Efficient LLMs across: *training*, *inference*, and *architecture*.

d3LLM: Ultra-Fast Diffusion LLM using Pseudo-Trajectory Distillation

Yu-Yang Qian^{1 2} Junda Su¹ Lanxiang Hu¹ Peiyuan Zhang¹ Zhijie Deng⁴ Peng Zhao^{† 2 3} Hao Zhang^{† 1}

¹ University of California, San Diego ² School of Artificial Intelligence, Nanjing University

³ National Key Laboratory for Novel Software Technology, Nanjing University ⁴ Shanghai Jiao Tong University

[†] Correspondence to: Peng Zhao zhaop@lamda.nju.edu.cn, Hao Zhang haozhang@ucsd.edu



Yu-Yang Qian



Junda Su



Lanxiang Hu



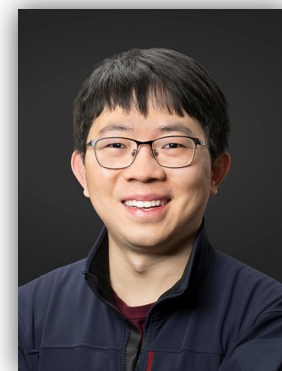
Peiyuan Zhang



Zhijie Deng



Peng Zhao



Hao Zhang



ICML

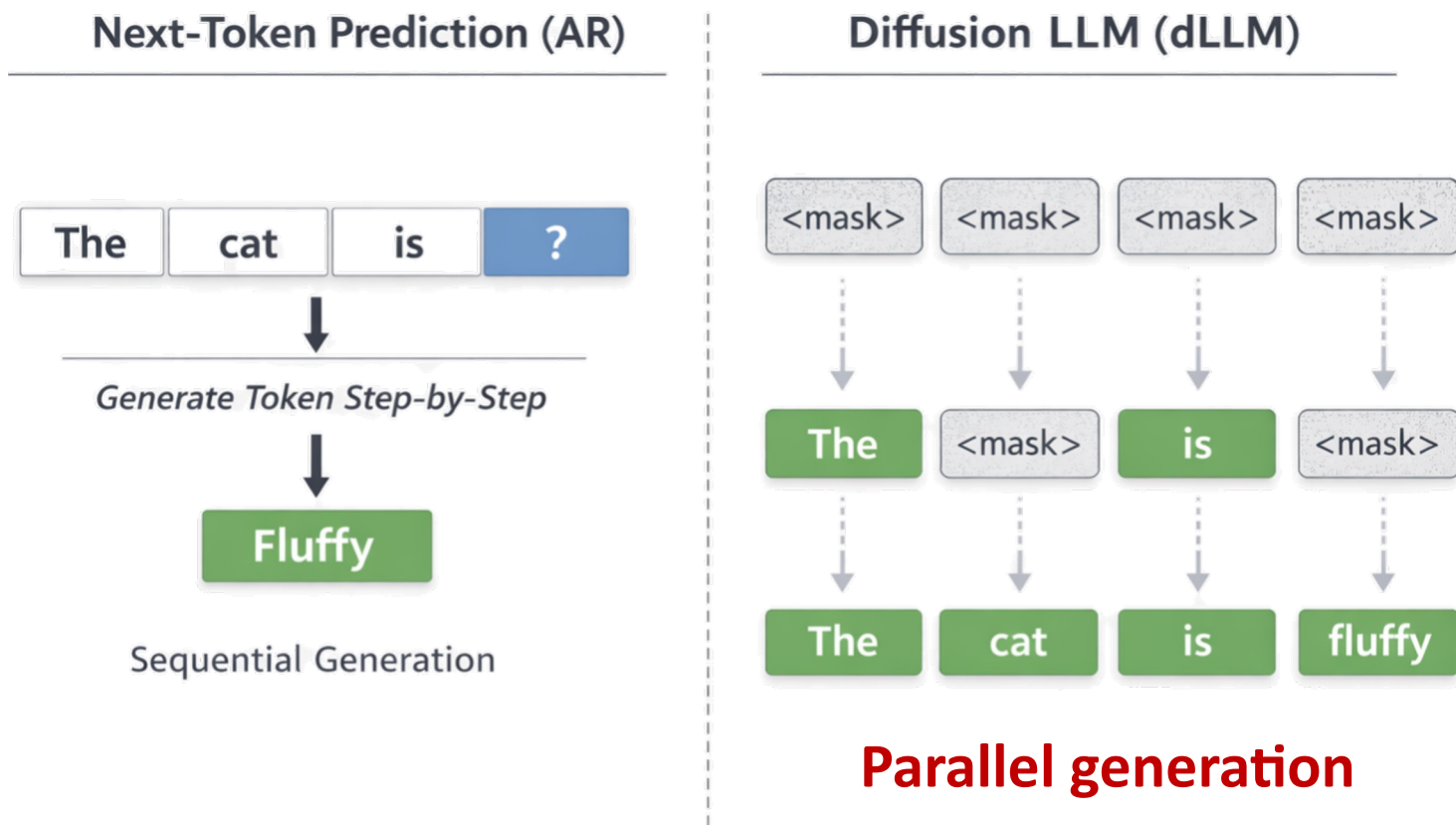
International Conference
On Machine Learning

Seoul, South Korea

[ICML'26]

Background: *Diffusion large language models (dLLMs)*

- *dLLMs* have shown promising capabilities, such as:
 - Parallel generation
 - Error correction
 - Random-order generation



Background: *Diffusion* large language models (dLLMs)

- *dLLMs* have shown promising capabilities, such as:
 - Parallel generation
 - Error correction
 - Random-order generation

- *Closed-source* dLLMs showed effectiveness:



Inception - *Mercury*



Google - *Gemini Diffusion*



ByteDance - *Seed Diffusion*



Mercury®: 1109 tokens/s

- However, *open-source* dLLMs exhibited **lower speed and accuracy**:

• *LLaDA-8B*



• *Dream-7B*



• *SDAR*



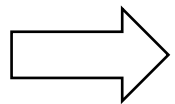
Previous Work: *Improving dLLMs*

- *Acceleration* of dLLMs:
 - *Fast-dLLM*: efficient *training-free* decoding method;
 - *dKV-Cache*: KV-Cache for dLLMs;
 - *dParallel*: distilling dLLM to get a faster one;
 - *D2F*: block-wise causal semi-AR-Diffusion;
 - *dInfer*: systematic implementation of *Efficient dLLM*;
 - *Refusion*: adapt from AR, slot-level parallel decoding.
- Improving the *performance* of dLLMs:
 - *MMaDA*: multimodal diffusion foundation models;
 - *TraDo*: dLLMs with *reasoning* ability;
 - *Fast-dLLM-v2*: directly post-training from an AR.



Previous Work: *One-side of the Coin*

- However, previous methods use *single, isolated metrics*:
 - **Efficiency-only metrics**: tokens per second (TPS) or tokens per forward (TPF);
 - **Performance-only metrics**: accuracy (solve rate / pass@1).
- Key insight: a fundamental *trade-off* for dLLMs:
 - dLLM naturally lives on an *accuracy–parallelism curve*;
 - Increasing parallelism almost always reduces accuracy, and vice versa;
 - Therefore, single metrics become misleading.



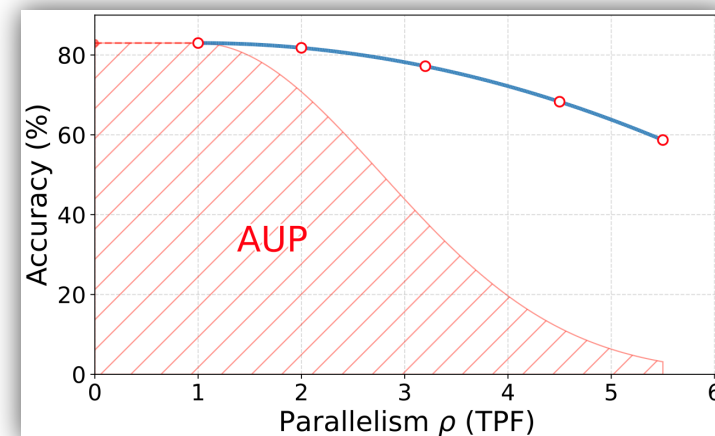
This motivates us to propose *a better metric* for dLLMs.

New Metric: *AUP*

Collect many parallelism-accuracy pairs $\{(\rho_i, y_i)\}_{i=1}^m$:

$$AUP \triangleq \rho_1 y_1 + \sum_{i=2}^m (\rho_i - \rho_{i-1}) \left(\frac{y_i W(y_i) + y_{i-1} W(y_{i-1})}{2} \right)$$

where $W(y) = \min(e^{-\alpha(1-y/y_{\max})}, 1)$ is weight function.

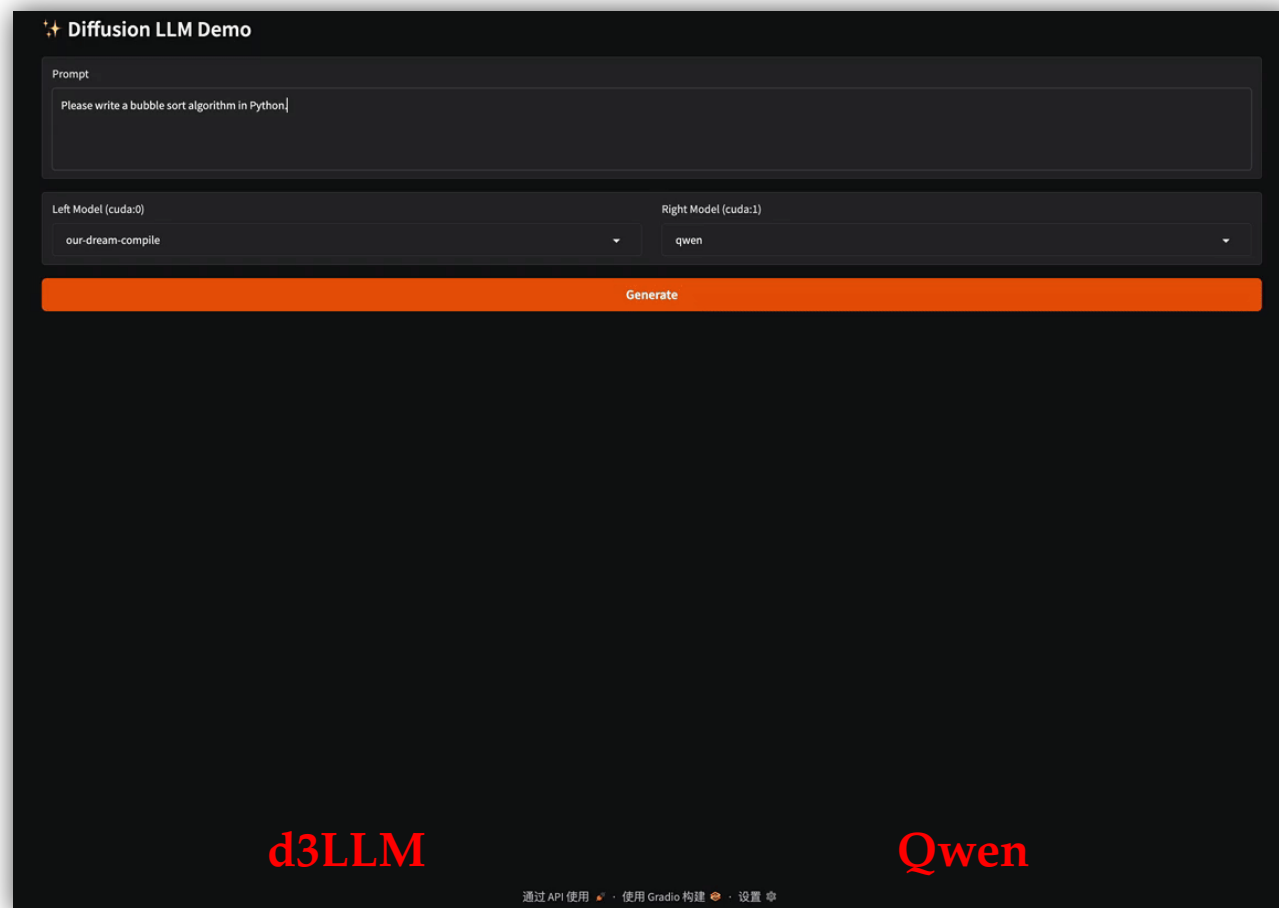


- We introduce *AUP* (*Accuracy Under Parallelism*):
 - A weighted area under the accuracy–parallelism curve;
 - *Jointly* measures both accuracy and parallelism;
 - Exponentially *penalizes* accuracy drops.

Encourage dLLMs to *strike a balance* between accuracy and parallelism.

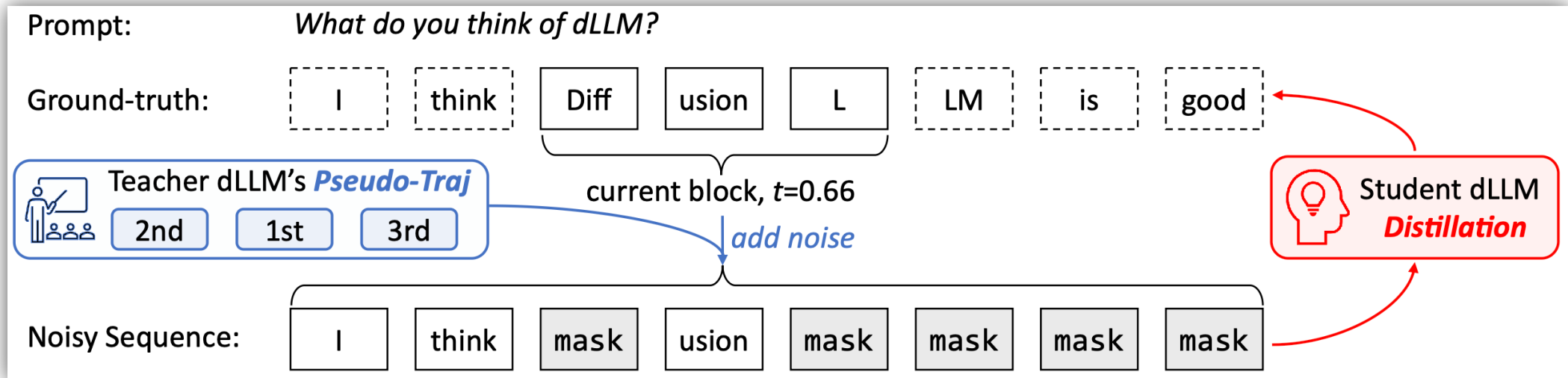
Approach: d3LLM

- Guided by AUP, we design *d3LLM* (*pseuDo-Distilled Diffusion LLM*):
 - A framework that improves both *training and inference*;
 - (i) Pseudo-trajectory distillation (*training*)
 - (ii) Multi-block decoding with KV-refresh (*inference*)
 - *Push* the accuracy–parallelism *frontier* of dLLMs.



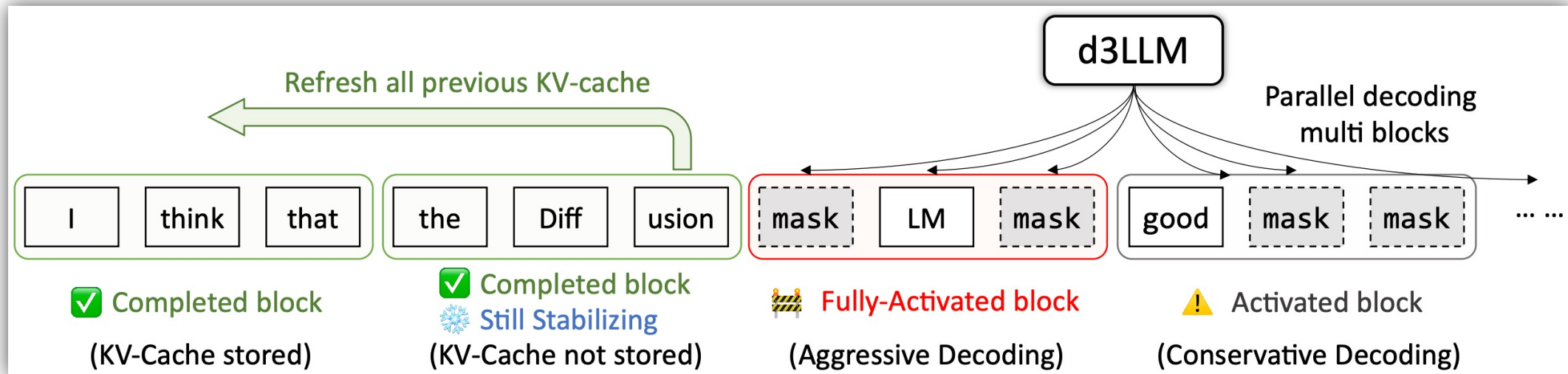
d3LLM: (i) training recipe

- Regarding the (post-)training 🏋️:
 - Utilizing the teacher's *pseudo-trajectory*: which tokens can be confidently decoded earlier (+18% TPF)
 - *Curriculum noise level*: gradually increase mask ratio (+12% TPF)
 - *Curriculum window size*: gradually increase window size (+8% TPF)



d3LLM: (ii) decoding strategy

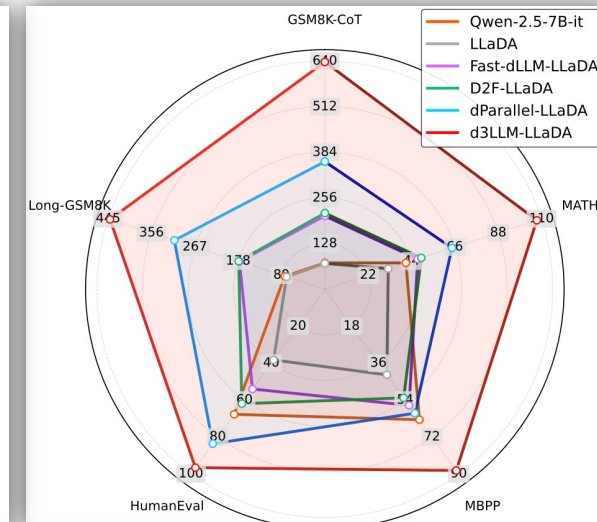
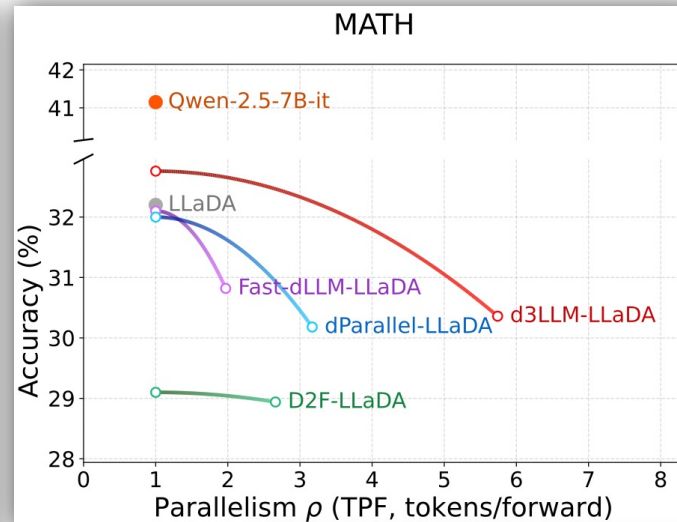
- Regarding the *decoding* ⚡:
 - Entropy-Based *Multi-Block Decoding*: decoding multiple blocks (+30% TPF)
 - KV-Cache with *Refresh*: refresh KV-caches before the stabilizing (+35% TPS)
 - *Early Stopping* on EOS Token: stop when EOS is generated (+5% TPF)



d3LLM: experimental results

- On *LLaDA*, *Dream* and *Dream-Coder*:

Benchmark	Method	TPF $\uparrow$	Acc (%) $\uparrow$	AUP Score $\uparrow$
GSM8K-CoT (0-shot)	LLaDA	1.00 $\pm$ 0.0	72.6 $\pm$ 0.2	72.6 $\pm$ 0.2
	Fast-dLLM-LLaDA	2.77 $\pm$ 0.1	74.7 $\pm$ 0.2	205.8 $\pm$ 6.4
	D2F-LLaDA	2.88 $\pm$ 0.1	73.2 $\pm$ 0.3	209.7 $\pm$ 6.1
	dParallel-LLaDA	5.14 $\pm$ 0.1	72.6 $\pm$ 0.2	358.1 $\pm$ 6.2
	d3LLM-LLaDA	9.11 $\pm$ 0.1	73.1 $\pm$ 0.3	637.7 $\pm$ 6.8
MATH (4-shot)	LLaDA	1.00 $\pm$ 0.0	32.2 $\pm$ 0.4	32.2 $\pm$ 0.4
	Fast-dLLM-LLaDA	1.97 $\pm$ 0.1	30.8 $\pm$ 0.3	47.2 $\pm$ 2.9
	D2F-LLaDA	2.38 $\pm$ 0.1	28.7 $\pm$ 0.2	45.5 $\pm$ 2.8
	dParallel-LLaDA	3.17 $\pm$ 0.1	30.2 $\pm$ 0.2	64.5 $\pm$ 3.1
	d3LLM-LLaDA	5.74 $\pm$ 0.1	30.4 $\pm$ 0.3	107.6 $\pm$ 3.2
MBPP (3-shot)	LLaDA	1.00 $\pm$ 0.0	41.7 $\pm$ 0.3	41.7 $\pm$ 0.3
	Fast-dLLM-LLaDA	2.13 $\pm$ 0.1	38.6 $\pm$ 0.3	56.6 $\pm$ 3.7
	D2F-LLaDA	1.94 $\pm$ 0.1	38.0 $\pm$ 0.2	50.0 $\pm$ 3.6
	dParallel-LLaDA	2.35 $\pm$ 0.1	40.0 $\pm$ 0.3	60.5 $\pm$ 3.9
	d3LLM-LLaDA	4.21 $\pm$ 0.1	40.6 $\pm$ 0.2	88.4 $\pm$ 4.0
HumanEval (0-shot)	LLaDA	1.00 $\pm$ 0.0	38.3 $\pm$ 0.5	38.3 $\pm$ 0.5
	Fast-dLLM-LLaDA	2.56 $\pm$ 0.1	37.8 $\pm$ 0.4	54.0 $\pm$ 2.9
	D2F-LLaDA	2.69 $\pm$ 0.1	36.6 $\pm$ 0.5	62.0 $\pm$ 2.7
	dParallel-LLaDA	4.93 $\pm$ 0.2	39.0 $\pm$ 0.4	83.7 $\pm$ 4.8
	d3LLM-LLaDA	5.95 $\pm$ 0.1	39.6 $\pm$ 0.6	96.6 $\pm$ 3.2
Long-GSM8K (5-shot)	LLaDA	1.00 $\pm$ 0.0	78.6 $\pm$ 0.2	78.6 $\pm$ 0.2
	Fast-dLLM-LLaDA	2.45 $\pm$ 0.1	78.0 $\pm$ 0.3	175.4 $\pm$ 6.4
	D2F-LLaDA	2.70 $\pm$ 0.1	73.7 $\pm$ 0.2	168.5 $\pm$ 6.0
	dParallel-LLaDA	4.49 $\pm$ 0.1	76.7 $\pm$ 0.3	309.1 $\pm$ 6.2
	d3LLM-LLaDA	6.95 $\pm$ 0.1	74.2 $\pm$ 0.3	441.1 $\pm$ 6.5



Method	H100 TPS $\uparrow$	A100 TPS $\uparrow$	Acc (%) $\uparrow$
Qwen-2.5-7B-it	57.3 (1.0 $\times$)	50.4 (1.0 $\times$)	74.1
LLaDA	27.9 (0.5 $\times$)	19.2 (0.4 $\times$)	72.6
Fast-dLLM-LLaDA	114.3 (2.0 $\times$)	79.1 (1.6 $\times$)	74.7
D2F-LLaDA	102.1 (1.8 $\times$)	76.2 (1.5 $\times$)	73.2
dParallel-LLaDA	172.2 (3.0 $\times$)	105.9 (2.1 $\times$)	72.6
d3LLM-LLaDA	288.9 (5.0$\times$)	183.3 (3.6$\times$)	73.1

🚀 **3.6 $\times$ –5 $\times$ speedup** over AR model (Qwen-2.5-7B-it), **10 $\times$ speedup** compared to vanilla LLaDA/Dream, with *negligible acc degradation*.

Incorporating *SGLang* engine

- Incorporating d3LLM models into *SGLang engine* 🚀:

			B200	H800/H100	A800/A100	Result		
	阈值	batch size	TPS	TPS	TPS	TPF	Acc	对齐HF
Qwen2.5-7B-Instruct	/	1	274.7	108.6	96.8	1	74.1	✓
Qwen3-8B	/	1	234.2	98.3	90	1	93.63	/
d3LLM-LLaDA (8B dense)	0.5	1	1240.99	545.31	251.61	9.91	75.36	✓
	0.5	4	1310.18	551.87	249.98	8.56	75.12	/
d3LLM-Dream (7B dense)	0.4	1	586.77	280.48	125.57	4.89	80.89	✓
	0.4	4	676.81	281.82	127.85	4.22	80.76	/
LLaDA 2.0 mini (16B A1B)	0.95	1	401.27	241.94	179.86	2.42	92.49	✓
	0.95	4	507.22	275.25	218.57	2.25	92.95	/
LLaDA 2.1 mini (16B A1B)	0.95	1	520.81	270.24	247.66	3.04	92.65	/
	0.95	4	516.89	390.04	310.55	2.49	92.87	/



3×–4.5× speedup over AR model (Qwen) with SGLang engine.

dLLM Leaderboard

- [dLLM Leaderboard - Hugging Face Space](#)

Rank	Method	Type	Foundation Model	GSM8K-CoT †	MATH †	MBPP †	HumanEval †	Long-GSM8K †	Avg AUP †
1	EAGLE-3	AR	Llama-3.1-8B-it	319.0 TPF:5.12 Acc:76.6	142.1 TPF:5.72 Acc:39.8	298.6 TPF:5.69 Acc:60.2	344.8 TPF:5.98 Acc:67.6	422.2 TPF:5.57 Acc:80.5	305.3
2	d3LLM-LLaDA	dLLM	LLaDA-8B-it	637.7 TPF:9.11 Acc:73.1	107.6 TPF:5.74 Acc:30.4	88.4 TPF:4.21 Acc:40.6	96.6 TPF:5.95 Acc:39.6	441.1 TPF:6.95 Acc:74.2	274.3
3	d3LLM-Dream	dLLM	Dream-v0-it-7B	391.3 TPF:4.94 Acc:81.4	97.5 TPF:3.92 Acc:38.2	141.4 TPF:2.96 Acc:55.6	129.5 TPF:3.20 Acc:57.1	348.6 TPF:4.80 Acc:77.2	221.7
4	dParallel-LLaDA	dLLM	LLaDA-8B-it	358.1 TPF:5.14 Acc:72.6	64.5 TPF:3.17 Acc:30.2	60.5 TPF:2.35 Acc:40.0	83.7 TPF:4.93 Acc:39.0	309.1 TPF:4.49 Acc:76.7	175.2
5	dParallel-Dream	dLLM	Dream-v0-it-7B	245.7 TPF:3.02 Acc:82.1	77.9 TPF:2.94 Acc:38.7	108.0 TPF:2.24 Acc:55.4	98.8 TPF:2.57 Acc:54.3	262.4 TPF:3.49 Acc:78.6	158.6
6	Fast-dLLM-v2	dLLM	Qwen-2.5-7B-it	176.1 TPF:2.21 Acc:81.5	126.7 TPF:2.61 Acc:48.7	114.1 TPF:2.04 Acc:59.1	128.9 TPF:2.58 Acc:61.7	207.2 TPF:2.58 Acc:81.0	150.6
7	D2F-LLaDA	dLLM	LLaDA-8B-it	213.8 TPF:2.88 Acc:74.4	49.0 TPF:2.66 Acc:28.9	53.0 TPF:2.13 Acc:39.0	62.0 TPF:2.69 Acc:40.6	176.9 TPF:2.70 Acc:75.7	110.9
8	Fast-dLLM-LLaDA	dLLM	LLaDA-8B-it	205.8 TPF:2.77 Acc:74.7	47.2 TPF:1.97 Acc:30.8	56.6 TPF:2.13 Acc:38.6	54.0 TPF:2.56 Acc:37.8	175.4 TPF:2.45 Acc:78.0	107.8

Summary

- (Efficiently) Leverage the interaction feedback to accelerate LLMs
- Across: training, inference, and model architecture.
 - [OnlineRLHF]: Explore *Online RLHF* (one-pass update) method for LLMs
 - [OnlineSPEC]: Continuously *evolve draft model* to speed LLM inference
 - [d3LLM]: Explore *diffusion LLM* for parallel generation
- Future work: investigate more complicated scenarios, such as agent

Thanks!

References

-  Long-Fei Li, Yu-Yang Qian, Peng Zhao, and Zhi-Hua Zhou. Provably efficient online RLHF with one-pass reward modeling. In Advances in Neural Information Processing Systems 38 (**NeurIPS**), pp. to appear, 2025.
-  Yu-Yang Qian, Hao-Cong Wu, Yichao Fu, Hao Zhang, and Peng Zhao. When drafts evolve: Speculative decoding meets online learning. **ArXiv preprint**, arXiv:2603.12617, 2026.
-  Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. d3llm: Ultra-fast diffusion LLM using pseudo trajectory distillation. **ArXiv preprint**, arXiv:2601.07568, 2026.

References

-  Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In Advances in Neural Information Processing Systems 38 (**NeurIPS**), 2025.
-  Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. In Proceedings of the 41st International Conference on Machine Learning (**ICML**), pp. 31131–31146, 2024.
-  Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q. Weinberger. Online adaptation to label distribution shift. In Advances in Neural Information Processing Systems 34 (**NeurIPS**), pages 11340–11351, 2021.

References

-  Yong Bai*, Yu-Jie Zhang*, Peng Zhao, Masashi Sugiyama, and Zhi-Hua Zhou. Adapting to Online Label Shift with Provable Guarantees. In: Advances in Neural Information Processing Systems 35 (**NeurIPS**), page 29960-29974, 2022.
-  Dheeraj Baby, Saurabh Garg, Thomson Yen, Sivaraman Balakrishnan, Zachary C. Lipton, and Yu-Xiang Wang. Online Label Shift: Optimal Dynamic Regret meets Practical Algorithms. In Advances in Neural Information Processing Systems 36 (**NeurIPS**), pages 65703–65742, 2023.
-  Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and Non-stationarity: Problem-dependent Dynamic Regret for Online Convex Optimization, Journal of Machine Learning Research (**JMLR**), 25(98):1–52, 2024.