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/

2026.03.17

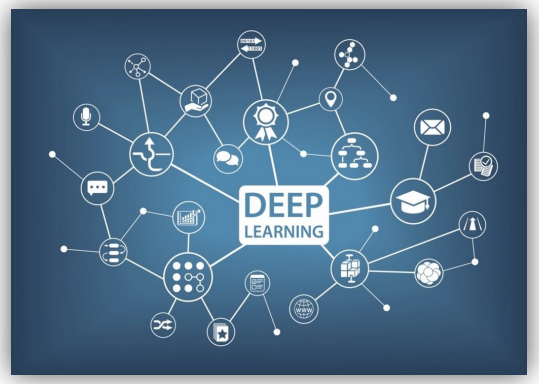
Background

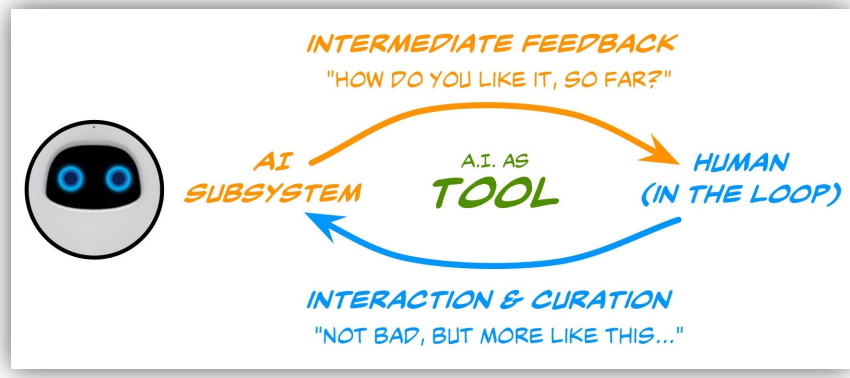
- LLMs have achieved remarkable success across diverse tasks,
  - e.g., ***chatbots***, ***video generation***, and ***agentic tasks***.

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

Background

- LLMs can achieve ***online feedback*** during deployment:





Traditional Machine Learning

- Model remains ***fixed*** after training.
- Single-round inference

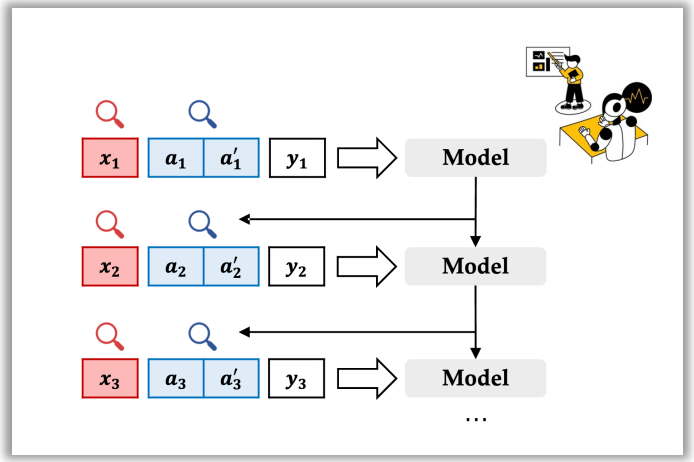
Deployment of ***LLMs***

- A large pre-trained model, later will be ***fine-tuned*** for downstream tasks
- ***Multi-round interaction*** with humans/envs

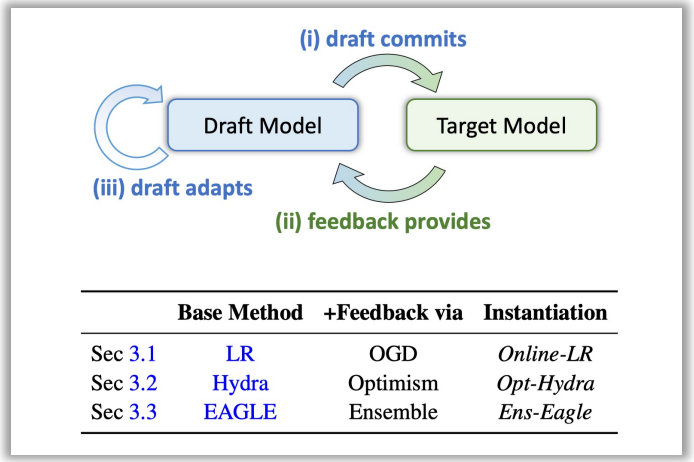
Can we ***(efficiently) leverage*** the interaction feedback to ***accelerate*** LLMs?

Outline

- Goal: ***Efficient LLMs*** via ***Online Feedback:***

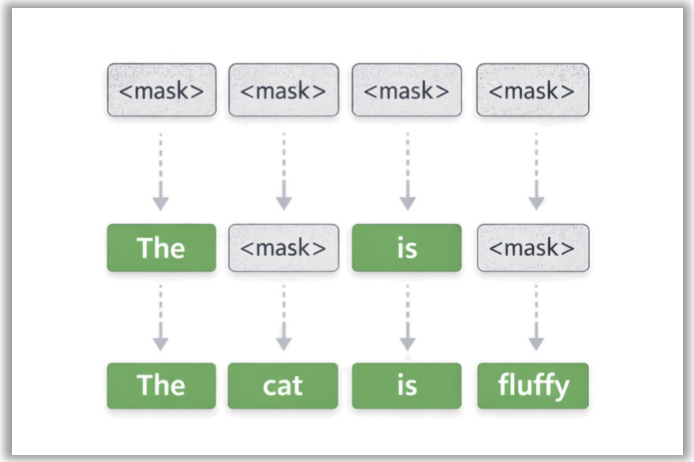


Online (one-pass) ***RLHF***
[NeurIPS'25]



	Base Method	+Feedback via	Instantiation
Sec 3.1	LR	OGD	*Online-LR*
Sec 3.2	Hydra	Optimism	*Opt-Hydra*
Sec 3.3	EAGLE	Ensemble	*Ens-Eagle*

Online ***Speculative Decoding***
[ICML'26]

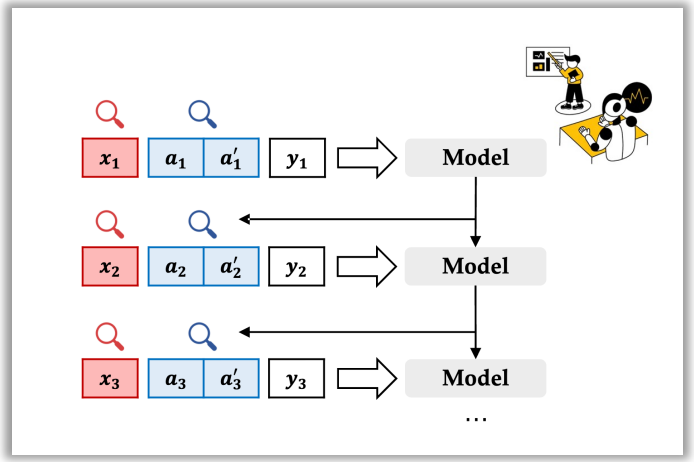


Ultra-Fast ***Diffusion LLMs***
[ICML'26]

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***.

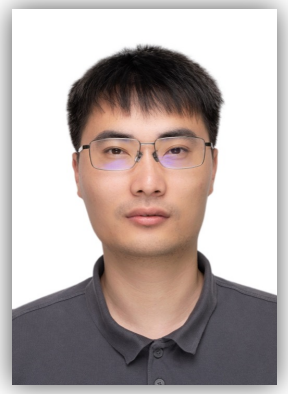
Provably Efficient Online RLHF with *One-Pass* Reward Modeling

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

NANJING UNIVERSITY

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:

Prompt x —SFT model→ $(a, a') \sim \pi_{\text{sft}}(a \mid x)$ —Query→ Which one is better?

- **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

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

r is the latent function.

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

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

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***:

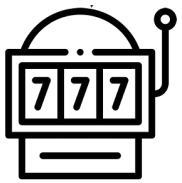
- prompt $\boldsymbol{x}$
- actions a, a'
- ***Goal*:** better reward model $r(\cdot,\cdot)$

Bradley-Terry (BT) Model

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

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

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



Logistic Contextual Bandits

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

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

Method: from *MLE* to *OMD*

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

MLE

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

one-pass update by OMD ⇩ "lookahead" local norm to keep historical information

***implicit* OMD**

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

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

Still no closed-form solution!

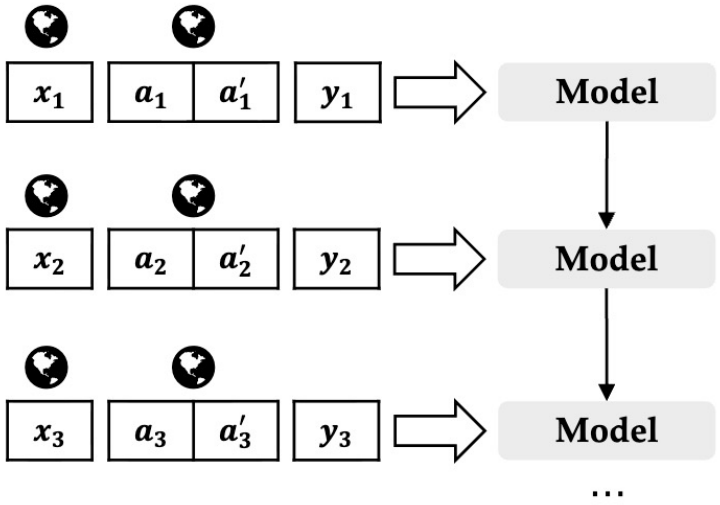
second-order Taylor expansion ⇩ $\widetilde{\ell}_{k,h}(\theta) = \ell_{k,h}(\widetilde{\theta}_{k,h}) + \langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \rangle + \frac{1}{2}\|\theta - \widetilde{\theta}_{k,h}\|_{H_{k,h}(\widetilde{\theta}_{k,h})}^2$

standard OMD

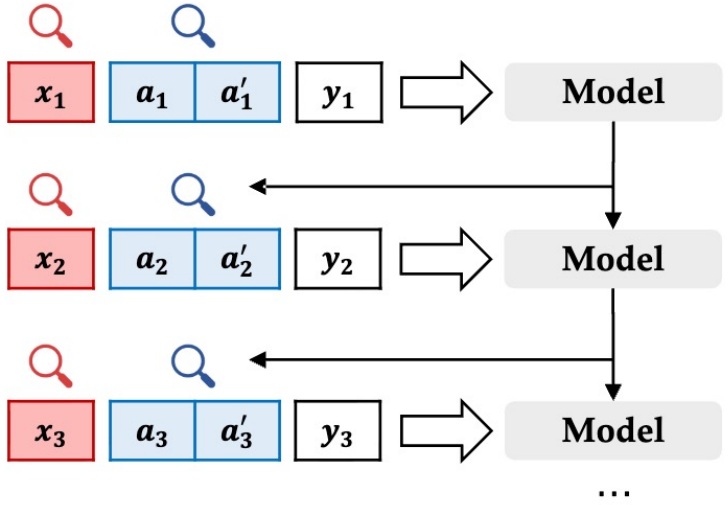
$$\widetilde{\theta}_{k+1,h} = \arg\min_{\theta\in\Theta} \left\{ \left\langle \nabla \ell_{k,h}(\widetilde{\theta}_{k,h}), \theta - \widetilde{\theta}_{k,h} \right\rangle + \frac{1}{2\eta} \left\| \theta - \widetilde{\theta}_{k,h} \right\|_{\widetilde{\mathcal{H}}_{k,h}}^2 \right\},$$

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

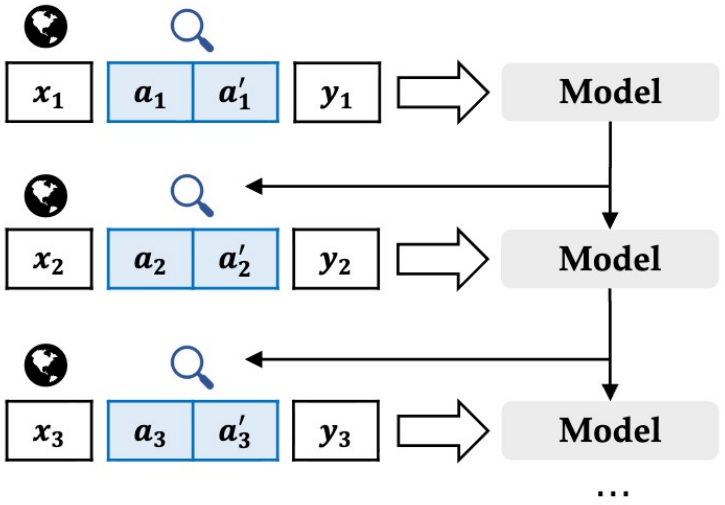
Method: Applied in Three Stages



(a) Passive Data Collection



(b) Active Data Collection



(c) Deployment-Time Adaptation

OMD update:

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

Active select samples:

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

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:

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

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

Theoretical Improvements

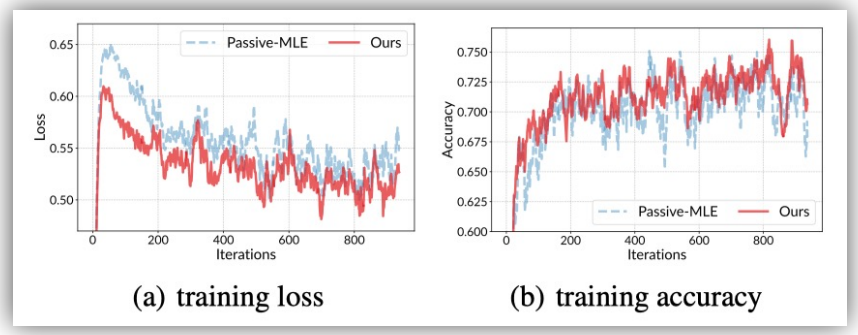
- ***Theoretically:*** improved *statistical* and *computational* efficiency:

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

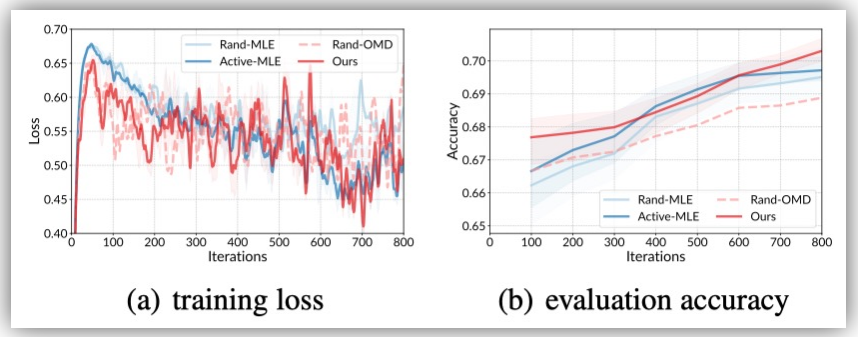
Achieving better regrets with ***less*** time/storage complexity.

Experiments: LLM's RLHF

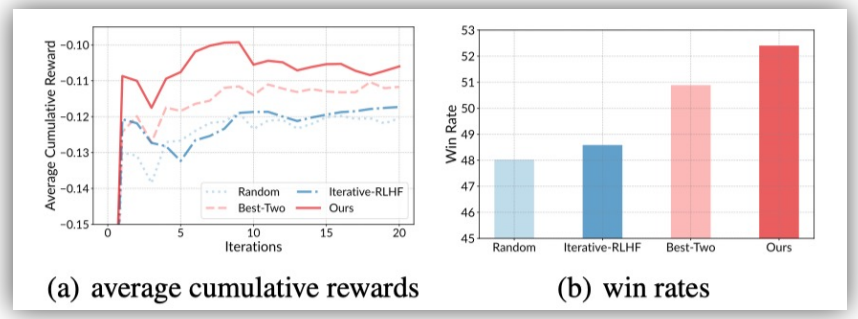
- Foundation Models: ***LLaMA 3 8B*** & ***Qwen 2.5-7B***
- Dataset: ***Ultrafeedback*** & ***Mixture2***



(1) Passive Data Collection



(2) Active Data Collection

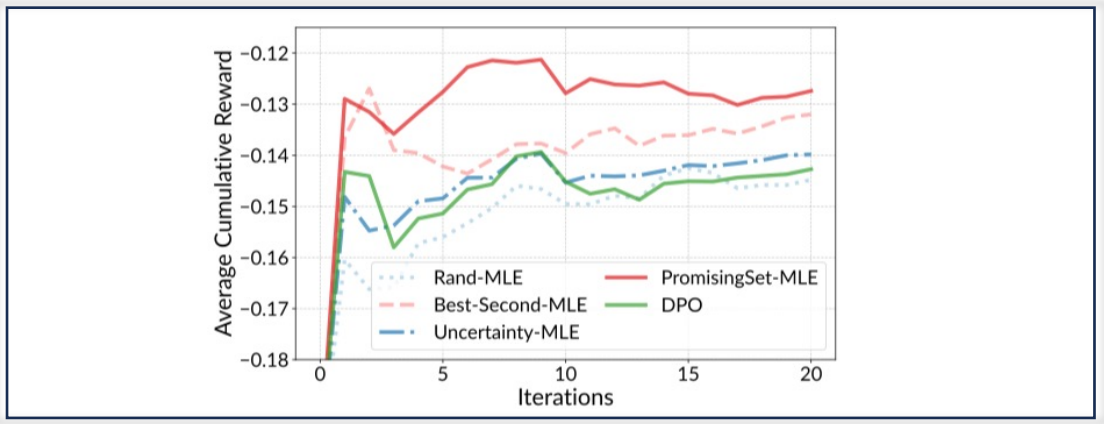


(3) Deployment-time Adaptation

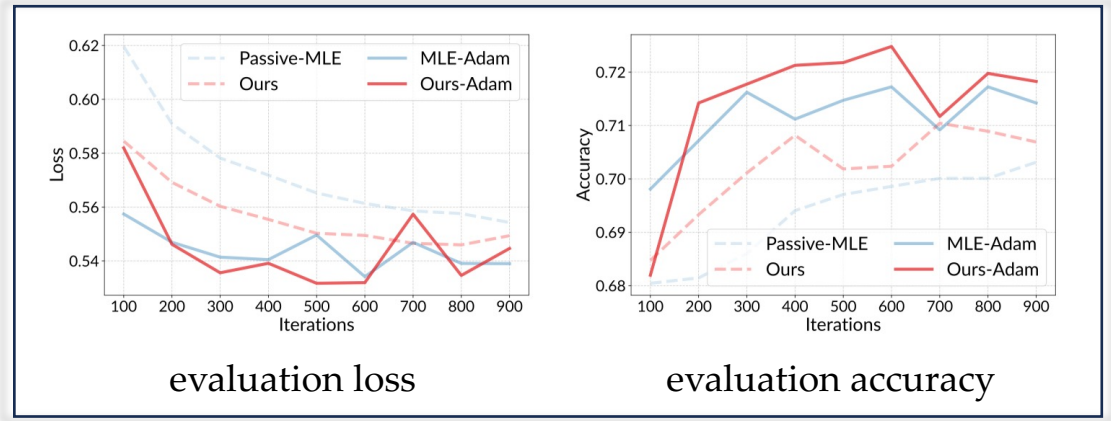
Method	ACC (%)	Time (s)
Rand-MLE	69.51±0.5	4876±47
Active-MLE	69.82±0.4	4982±52
Rand-OMD	68.97±0.6	1456±31
Ours	70.43±0.3	1489±36

(4) Efficiency Comparison

Experiments

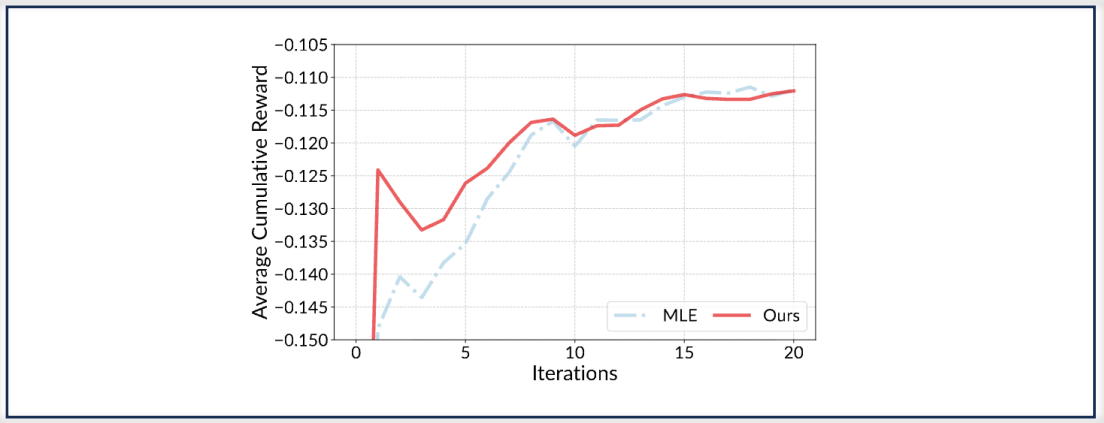


Our method surpass ***DPO*** in deployment stage.



evaluation loss evaluation accuracy

Our method is also comparable with ***Adam***.



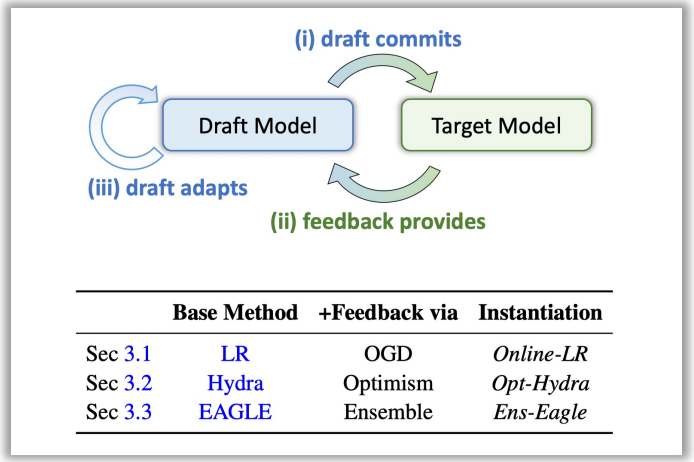
Full parameter update experiment.

Achieve comparable performance with ***improved efficiency***.

Outline: Online *Speculative Decoding*

- Goal: Efficient inference through speculative decoding:

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



	Base Method	+Feedback via	Instantiation
Sec 3.1	LR	OGD	*Online-LR*
Sec 3.2	Hydra	Optimism	*Opt-Hydra*
Sec 3.3	EAGLE	Ensemble	*Ens-Eagle*

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

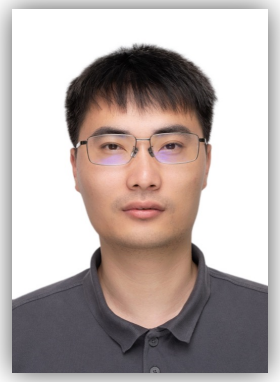
Yu-Yang Qian[1,2] **Hao-Cong Wu**[1,2] **Yichao Fu**[3] **Hao Zhang**[3] **Peng Zhao**[1,2]
[1]National Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Artificial Intelligence, Nanjing University [3]University of California, San Diego
{qianyy,zhaop}@lamda.nju.edu.cn hcwu@smail.nju.edu.cn
{yif034,haozhang}@ucsd.edu


Yu-Yang Qian


Hao-Cong Wu


Yichao Fu


Hao Zhang

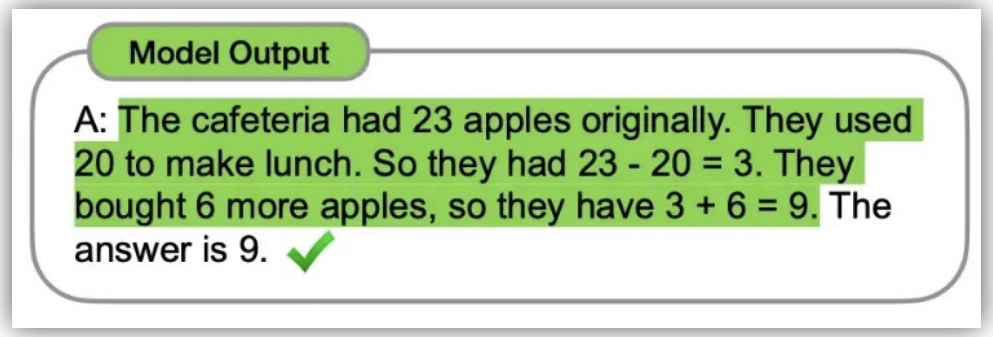

Peng Zhao

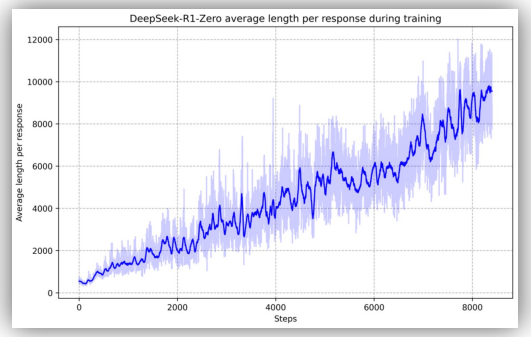
Learning And Mining from DatA

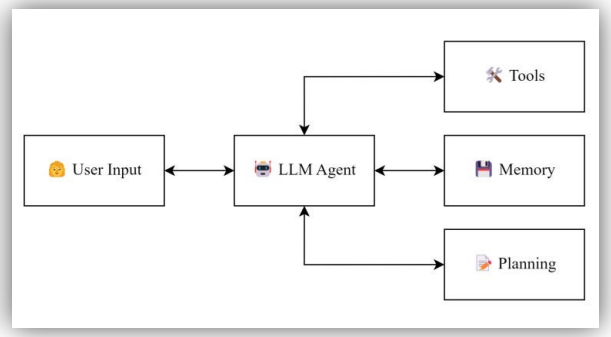
ICML
International Conference On Machine Learning
Seoul, South Korea
[ICML'26]

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



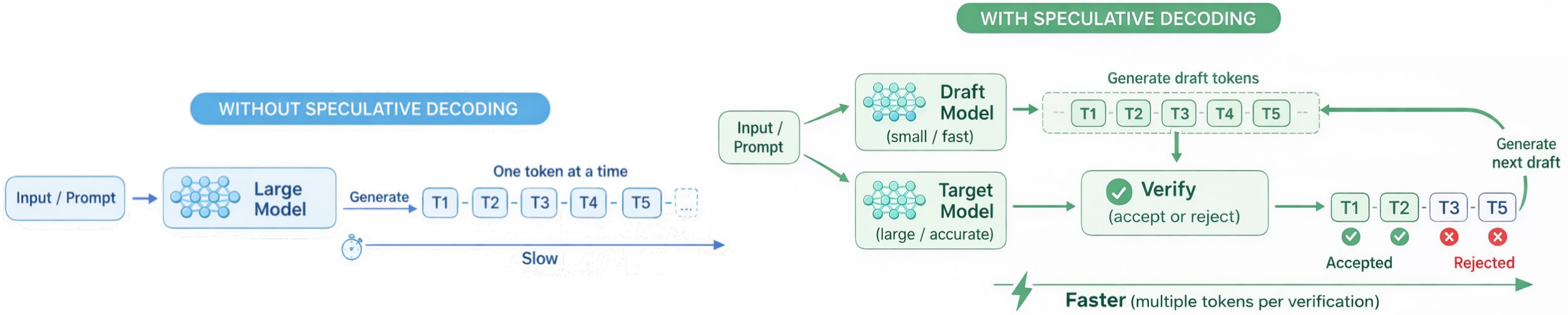




Inference cost and inference latency of LLMs lead to a decline in user experience.

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.



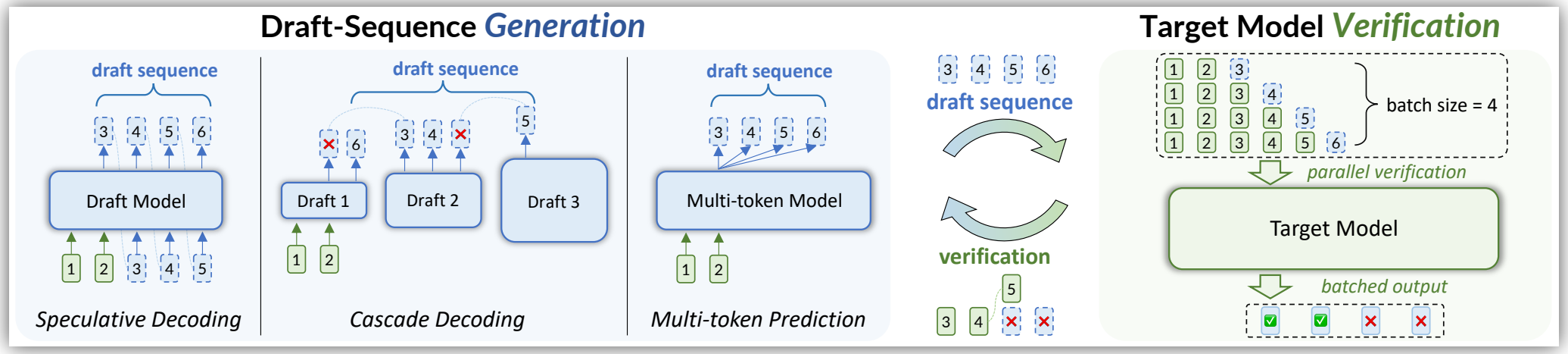
Background

- 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.

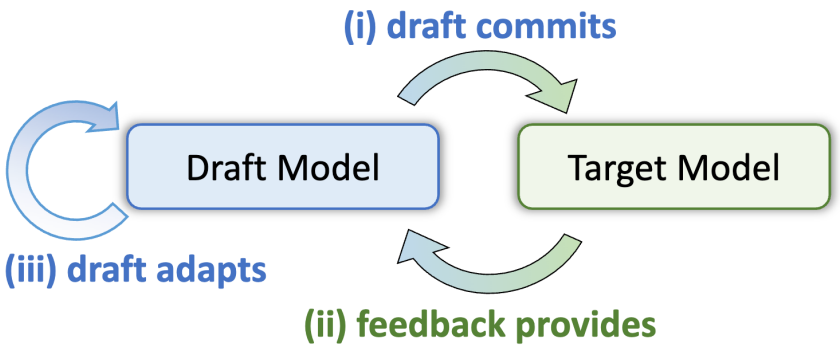
The ***"acceptance rate"*** of the ***draft model*** directly determines speedup rate.

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 $t = 1, \ldots, T$:

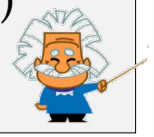
(1) the learner submits draft model(s) $\mathbf{w}_t \in \mathcal{W}$; (draft model generating)

⇩

(2) target model revels 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***

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

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

Algorithm 1 OnlineSPEC Framework

Input: Initial draft model parameter $\mathbf{w}_0 \in \mathcal{W}$ with its predicted distribution $q_{\mathbf{w}_0}(\cdot \mid \mathbf{x})$, target model $\mathbf{v} \in \mathcal{V}$ with its predicted distribution $p_{\mathbf{v}}(\cdot \mid \mathbf{x})$, 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 $t = 1, \ldots, T$ **do**

▷ Generate the draft sequence:

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

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

▷ Verify by target model:

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

▷ Determine the accepted token length n_t:

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

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

▷ Verify and update the draft sequence:

if $n_t < k$ **then**

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

else

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

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

▷ 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 $\hat{\mathbf{x}} \triangleq [x_1, x_2, \ldots]$.

A Unified Framework: *OnlineSPEC*

- Formulated as an *online learning problem:*

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

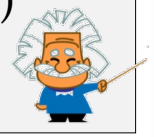
(1) the learner submits draft model(s) $\mathbf{w}_t \in \mathcal{W}$; (draft model generating)

⇩

(2) target model revels 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

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

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 $q_{\mathbf{w}_0}(\cdot \mid \mathbf{x})$, target model $\mathbf{v} \in \mathcal{V}$ with its predicted distribution $p_{\mathbf{v}}(\cdot \mid \mathbf{x})$, 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 $t = 1, \ldots, T$ **do**

▷ Generate the draft sequence:

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

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

▷ Verify by target model:

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

▷ Determine the accepted token length n_t:

Sample $r_1 \sim U(0,1), \ldots, r_k \sim U(0,1)$ 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**

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

else

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

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

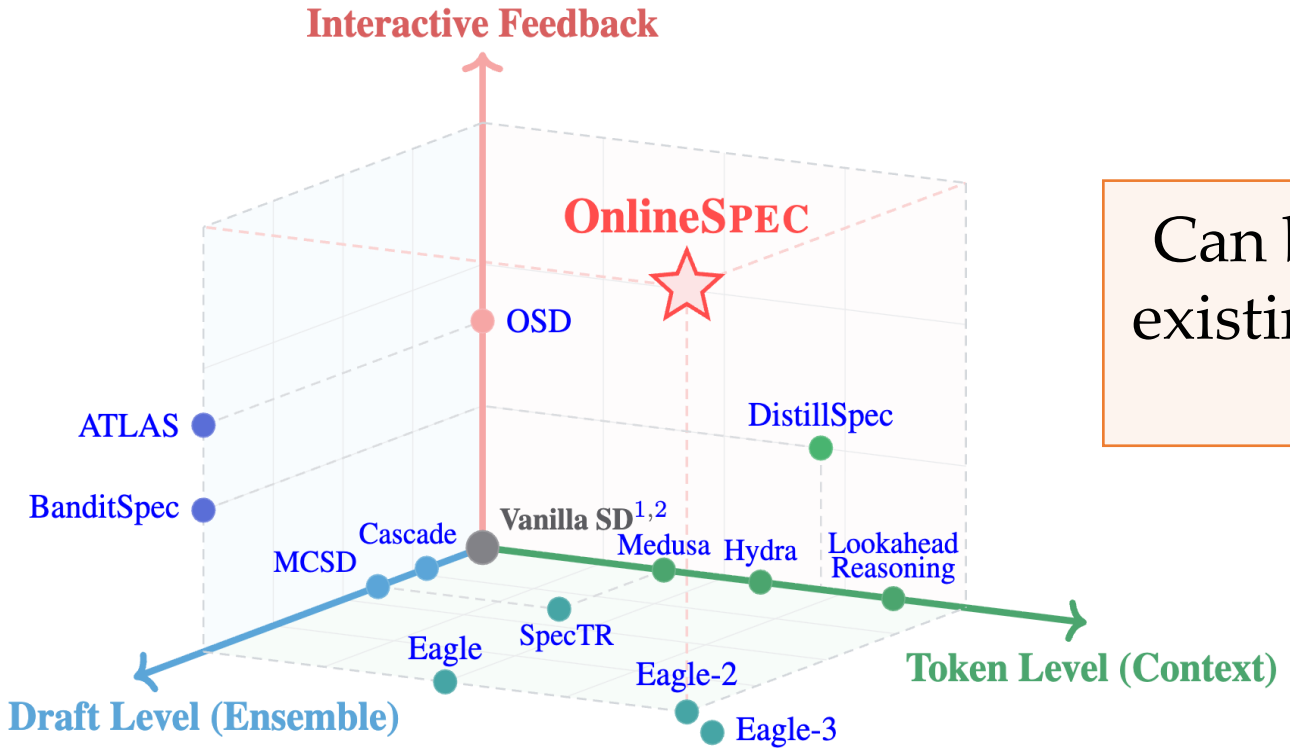
▷ Receive interactive feedback: observe the loss function $f_t : \mathcal{W} \mapsto \mathbb{R}$ from the target model $\mathbf{v}$.

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

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

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 $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left[\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)\right]$	*Online-LR*	$\gamma = \Omega\left(\frac{1}{\alpha k+1}\min\left\{k+1, \frac{T^{1/4}}{\sqrt{1+P_T}}\right\}\right)$
Hydra	Optimistic online learning $\mathbf{w}_t = \Pi_{\mathcal{W}}[\widehat{\mathbf{w}}_t - \eta \mathbf{h}_t];\ \widehat{\mathbf{w}}_{t+1} = \Pi_{\mathcal{W}}[\widehat{\mathbf{w}}_t - \eta \nabla f_t(\mathbf{w}_t)]$	*Opt-Hydra*	$\gamma = \Omega\left(\frac{1}{\alpha k+1}\min\left\{k+1, \frac{\sqrt{T}}{(1+\delta_T)^{1/4}\sqrt{1+P_T}}\right\}\right)$
Eagle/Eagle-3	Online ensemble learning $\mathbf{w}_t = \sum_{i=1}^{N} p_t^i \cdot \mathbf{w}_t^i$	*Ens-Eagle*	$\gamma = \Omega\left(\frac{1}{\alpha k+1}\min\left\{k+1, \frac{T^{1/4}}{(1+P_T)^{1/4}}\right\}\right)$

Experimental Results

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

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

Experimental Results

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

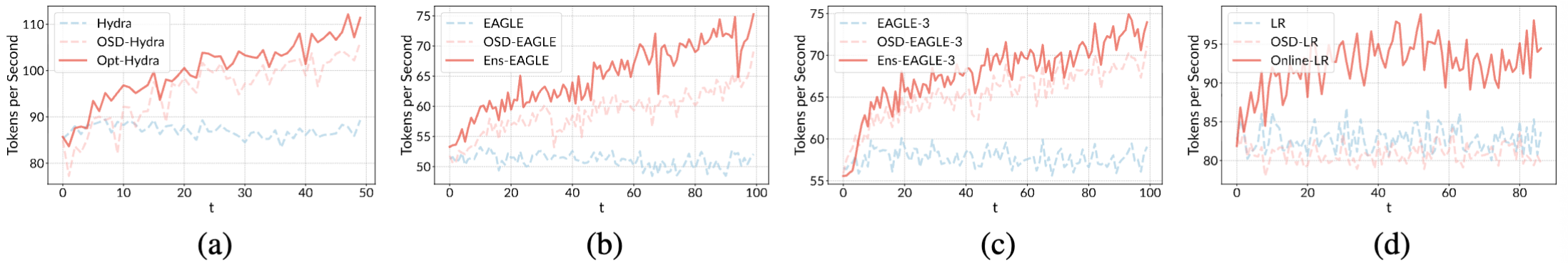


Figure 3. Evolution of tokens per second (TPS) on the *GSM8K* dataset: (a) *Opt-Hydra* with *lmsys/Vicuna-7B-v1.3*, (b) *Ens-EAGLE* with *lmsys/Vicuna-7B-v1.3*, (c) *Ens-EAGLE-3* with *lmsys/Vicuna-7B-v1.3*, and (d) *Online-LR* with *Qwen/Qwen3-8B*. This demonstrates consistent performance improvements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

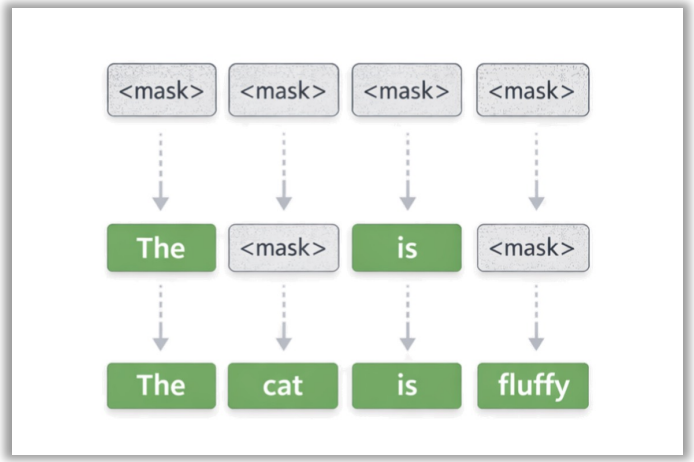
🚀 ***Continuously*** improving the draft model and inference efficiency.

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***.

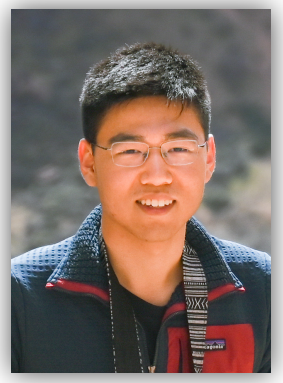
d3LLM: Ultra-Fast Diffusion LLM using Pseudo-Trajectory Distillation

Yu-Yang Qian [1 2] **Junda Su** [1] **Lanxiang Hu** [1] **Peiyuan Zhang** [1] **Zhijie Deng** [4] **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

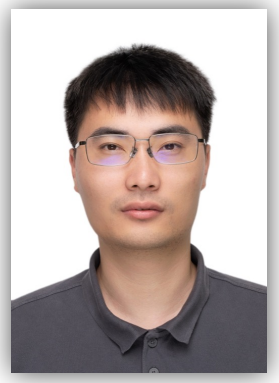

Yu-Yang Qian


Junda Su


Lanxiang Hu


Peiyuan Zhang


Zhijie Deng

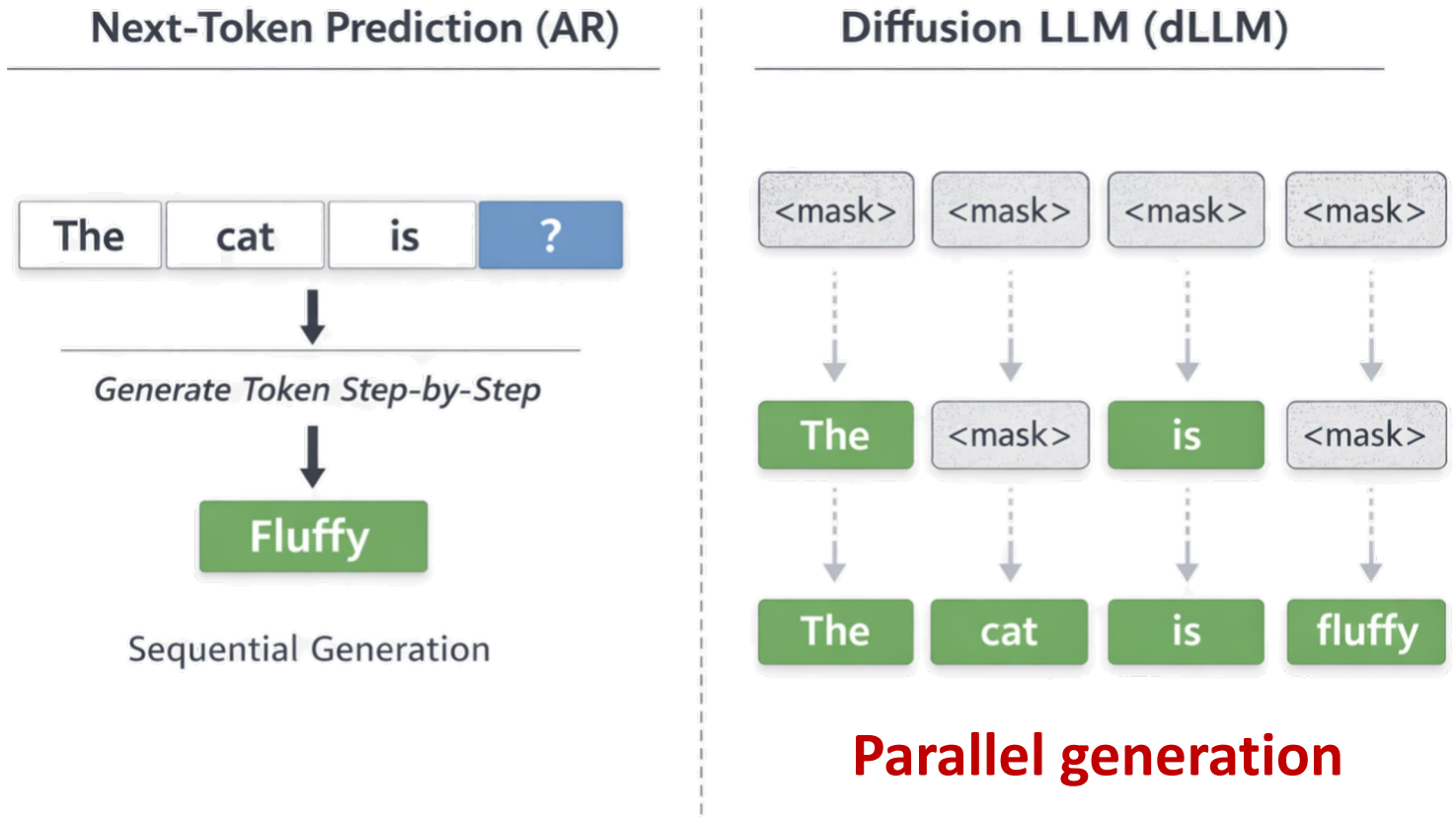

Peng Zhao


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.

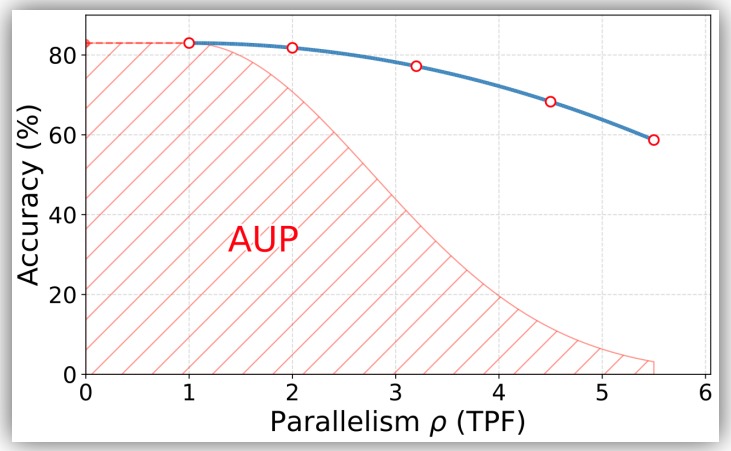
⇨ This motivates us to propose ***a better metric*** for dLLMs.

New Metric: ***AUP***

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

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

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

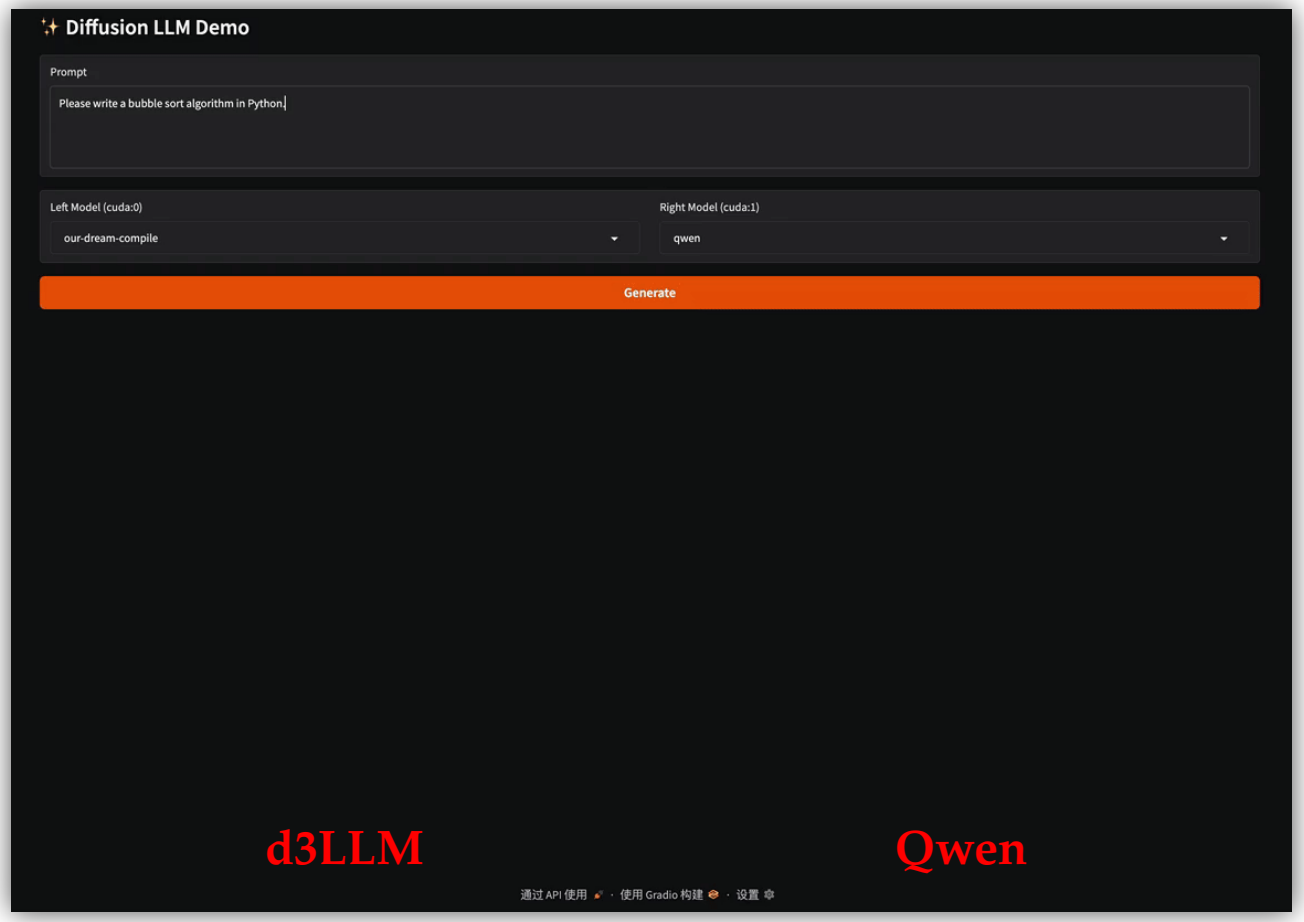
- Guided by AUP, we design ***d3LLM*** (*pseuDo-Distilled Diffusion LLM*):

  - A framework that improves both ***training and inference;***

    (i) Pseudo-trajectory distillation ***(training)***

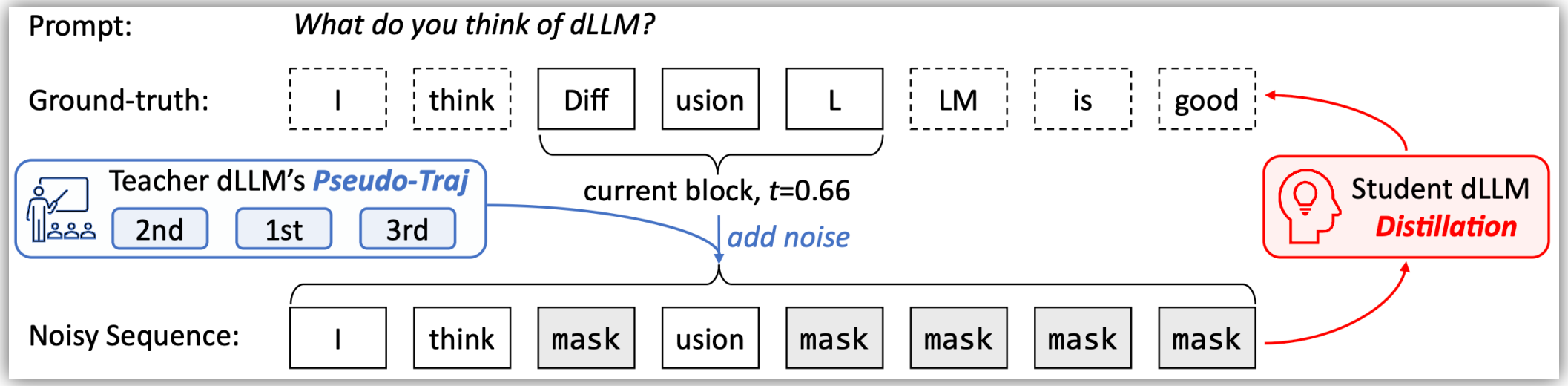
    (ii) Multi-block decoding with KV-refresh ***(inference)***

  - ***Push*** the accuracy–parallelism ***frontier*** of dLLMs.



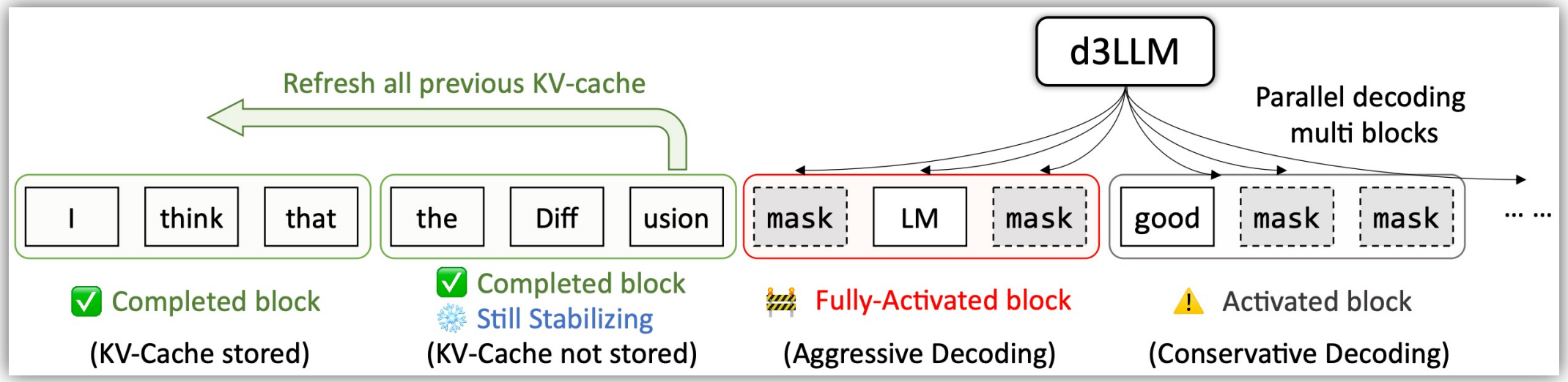
d3LLM: *(i) training recipe*

- 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*

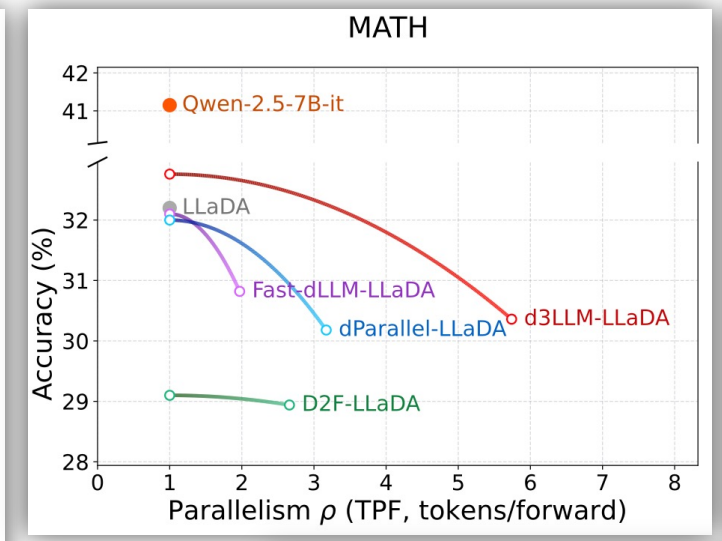
- Regarding the ***decoding*** ⚡:
  - Entropy-Based ***Multi-Block Decoding***: decoding multiple blocks (+30% TPF)
  - KV-Cache with ***Refresh***: refresh KV-caches before the stabilizing (+35% TPS)
  - ***Early Stopping*** on EOS Token: stop when EOS is generated (+5% TPF)

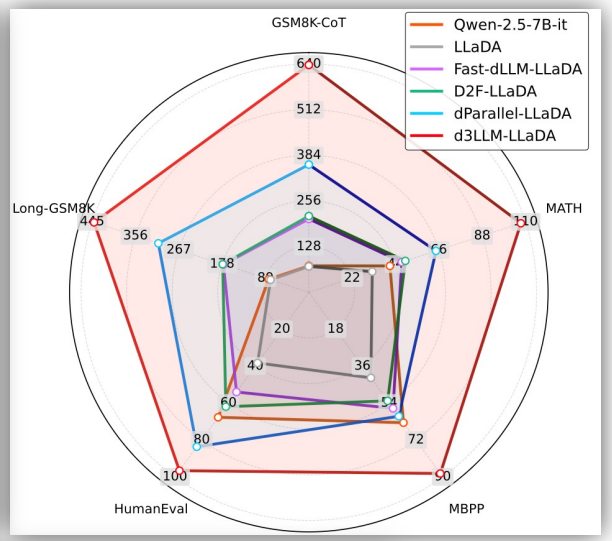


d3LLM: experimental results

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

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





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

🚀 **3.6×–5×** speedup over AR model (Qwen-2.5-7B-it), **10×** 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.