Online Learning and Speculative Decoding

Peng Zhao

LAMDA Group

School of Artificial Intelligence

Nanjing University

Outline

- Online Learning
- Speculative Decoding
- Conclusion

Outline

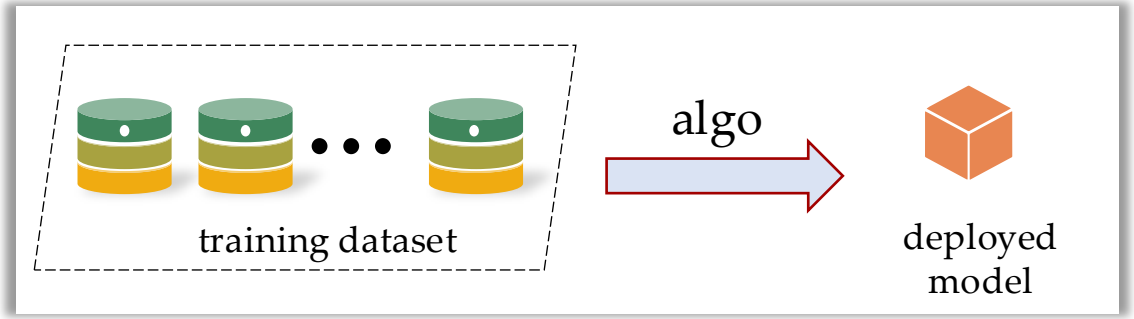
- Online Learning
- Speculative Decoding
- Conclusion

Online Learning

- **Batch Learning**: collect data first, and train model next

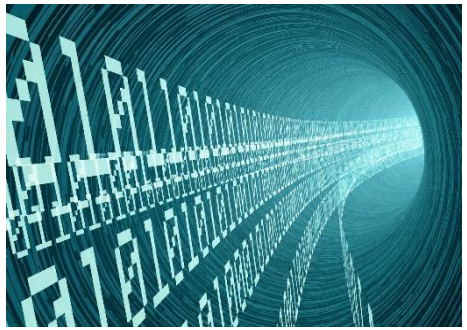


offline data

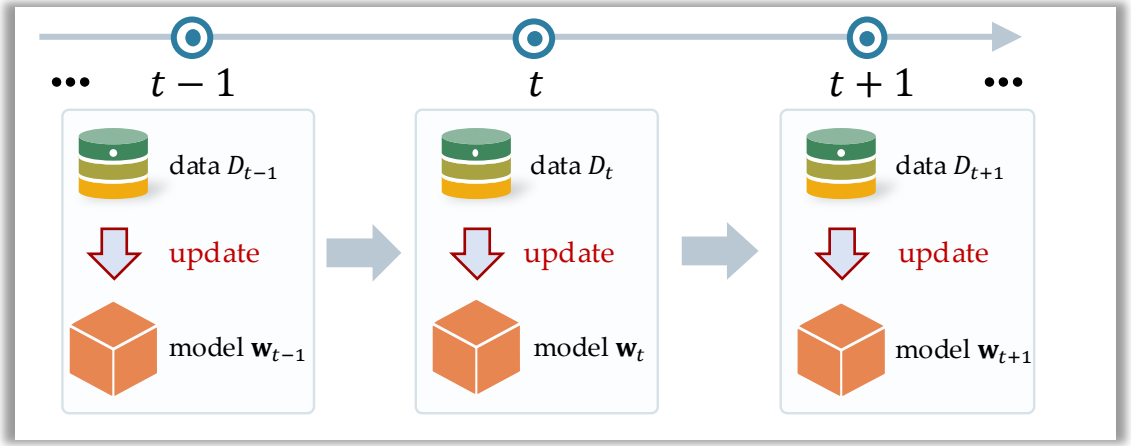


Collect First, Use Later

- **Online Learning**: update the model upon the data arrival



stream data



Learn upon Data Arrives

Online Learning

- It becomes more and more important when the interaction matters.



recommendation



agentic system



self-driving car

- Welcome to era of experience

Welcome to the Era of Experience

David Silver, Richard S. Sutton*

Abstract

We stand on the threshold of a new era in artificial intelligence that promises to achieve an unprecedented level of ability. A new generation of agents will acquire superhuman capabilities by learning predominantly from experience. This note explores the key characteristics that will define this upcoming era.

2025/04/10

The Alberta Plan for AI Research

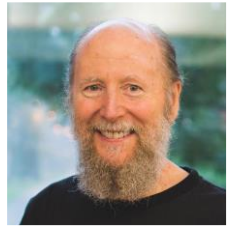
Richard S. Sutton, Michael Bowling, and Patrick M. Pilarski

University of Alberta
Alberta Machine Intelligence Institute
DeepMind Alberta

History suggests that the road to a firm research consensus is extraordinarily arduous.

— Thomas Kuhn, *The Structure of Scientific Revolutions*

arxiv: 2208.11173



Rich Sutton
2024 Turing Award

Non-stationary Online Learning

- **Distribution shift**: data are usually collected in open environments

species monitoring



urban computing



route planning



- When the data is in the form of stream, the distribution changes on the fly.

continuous distribution shift

→

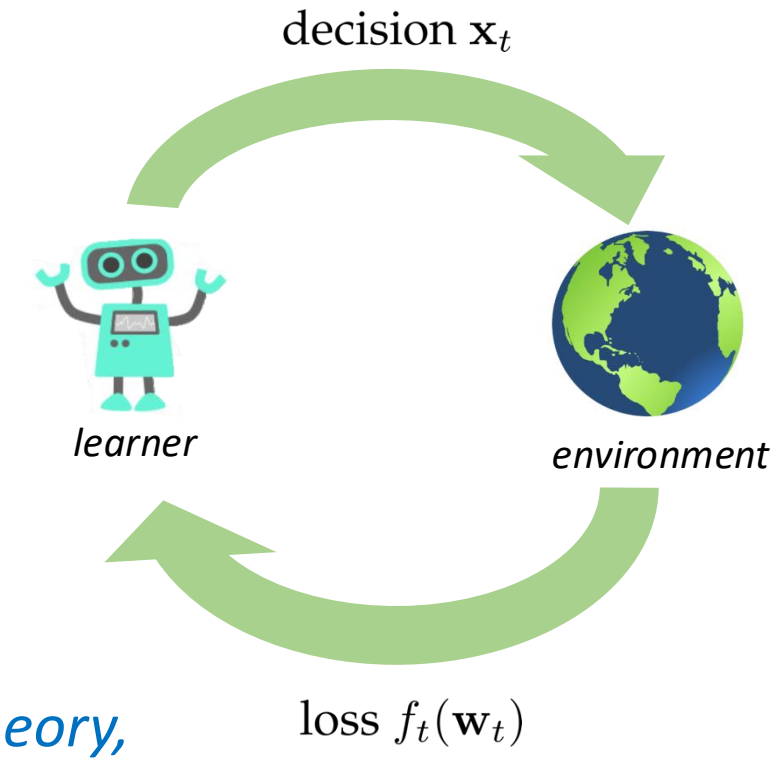
provably robust methods for non-stationary online learning

Formulation

- Optimization view: online learning as an interactive game between ***learner*** and ***environment***.

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

1. The learner chooses a decision $\mathbf{w}_t \in \mathcal{W} \subseteq \mathbb{R}^d$.
2. The environment reveals a convex loss f_t.
3. The learner incurs $f_t(\mathbf{w}_t)$ and uses feedback to update.



⇨ *connecting various areas: optimization, information theory, statistics; also, LLM-RL, agent, attention mechanism, …*

Regret Metric

- The best in hindsight

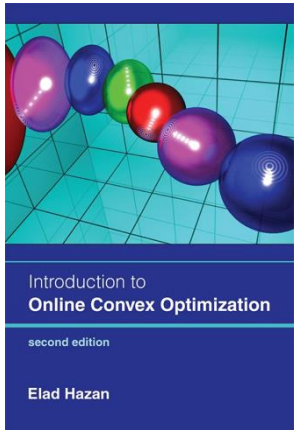
  After T rounds, suppose we look back and imagine that we *did* know all $\{f_t\}_{t=1}^T$.

  Then the best decision in hindsight would be: $\mathbf{w}^\star \in \arg\min_{\mathbf{w}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{w})$.

- To benchmark with the offline model (optimal in hindsight)

Regret: $$\mathrm{REG}_T = \sum_{t=1}^T f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^T f_t(\mathbf{w})$$

"If I had known all the outcomes ahead of time, I would have chosen differently — and I regret the extra loss I've accumulated."



Introduction to Online Convex Optimization

Elad Hazan
MIT Press, 2022

Rethinking the metric

Regret: online prediction as good as the best offline model

$$\textsc{Reg}_T \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

cumulative loss of the best offline model

optimal model changes in non-stationary environments

Dynamic Regret

$$\text{D-}\textsc{Reg}(\mathbf{u}_1, \cdots, \mathbf{u}_T) \triangleq \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t),$$

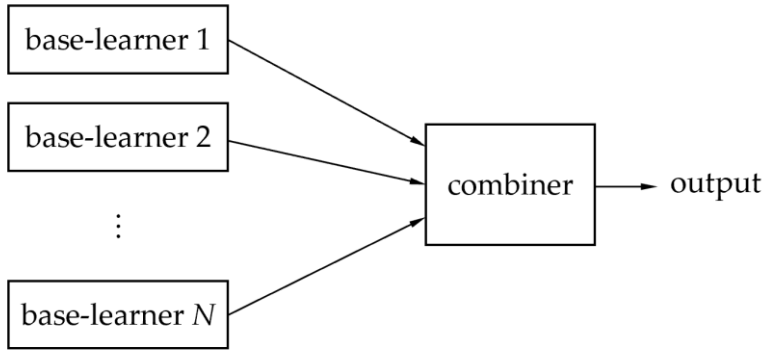
where $\mathbf{u}_1, \ldots, \mathbf{u}_T \in \mathcal{W}$ is a sequence of changing comparators that can be arbitary chosen.

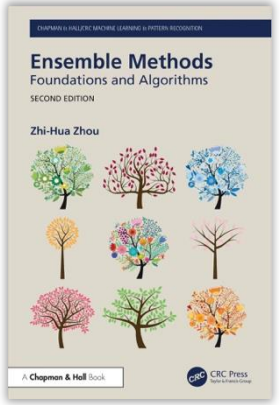
Key difficulty: the ***uncertainty*** due to unknown environmental changes.

Our Solution

Basic idea: Ensemble Method

- *Protocol*: combine multiple base learners to achieve robustness
- *Advantage*: achieve more robust results under uncertain or even changing environments





Zhi-Hua Zhou. Ensemble Methods: Foundations and Algorithms, 2nd edition, 2025.

Online Ensemble （在线集成）：

(1) **base learner**: an online learner to cope with a certain amount of non-stationarity

(2) **schedule**: a set of parameters for initiating base learners that encourage diversity

(3) **meta learner**: an expert-tracking learner that can combine base learners' decision

Deploying Online Ensemble

According to the feedback information of online learning, we have

- Full-information online learning [NeurIPS'20; ICML'22; NerIPS'22; NerIPS'23; JMLR'24; COLT'26]

  the learner can obtain the *gradient information* of the online function

- Partial-info online learning [AISTAST'20; JMLR'21; COLT'22; AISATST'23; ICML'24; NeurIPS'25; ICML'26]

  the learner can obtain the *function value* only, without the gradient information

- Decision-dependent online learning [ICML'22; JMLR'23; NeurIPS'23; AISATST'24; NIPS'24; TIT'26]

  the historical decision may affect the current functions (including gradient and value)

Based on the unified "online ensemble" framework,
we can obtain optimal (or best-known) dynamic regret

Bandit Convex Optimization (BCO)

- BCO with one-point feedback

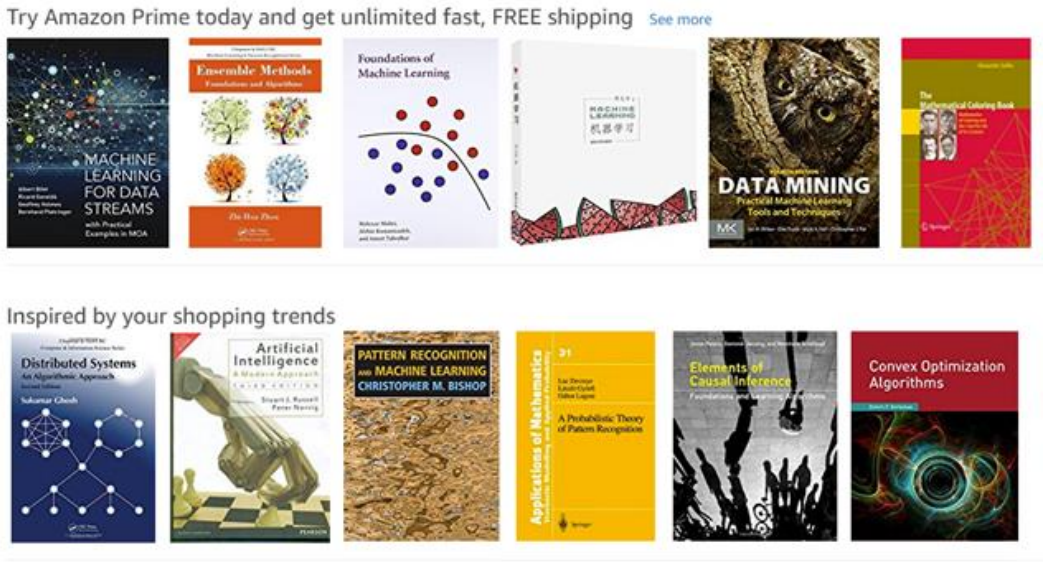
  the learner sends a single point $\mathbf{w}_t \in \mathcal{W}$, and then receives the ***function value*** $f_t(\mathbf{w}_t)$ only

  [Flaxman et al., SODA 2005; Bubeck et al., STOC 2017]

- BCO with two-point feedback

  the learner sends two points $\mathbf{w}_t^1, \mathbf{w}_t^2 \in \mathcal{W}$, and then receives their ***function values***, namely, $f_t(\mathbf{w}_t^1)$ and $f_t(\mathbf{w}_t^2)$, only

  [Agarwal et al., COLT 2010; Shamir, JMLR 2017]



online recommendation

Base Algorithm

- We need to develop base learner and prove its dynamic regret.

Online Gradient Descent (OGD)

for $t = 1$ to T **do**

Play model $\mathbf{w}_t$ and suffer loss $f_t(\mathbf{w}_t)$

Update the model

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)]$$

end for

Challenge: with only bandit feedback, *learner doesn't have gradient*

FKM estimator [Flaxman et al., SODA'05]

construct $\mathbf{w}_t$ using the perturbation technique

$$\mathbf{w}_t \triangleq \widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t$$

$\mathbf{s}_t$ is random vector sampled from ball $\mathbb{B} = \{\mathbf{v} \mid \|\mathbf{v}\| \leq 1\}$

$$\mathbb{E}\left[\frac{d}{\delta} f_t(\mathbf{w}_t) \cdot \mathbf{s}_t\right] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t) \approx \nabla f_t(\mathbf{w}_t)$$

with smoothed function $\widehat{f}_t(\mathbf{w}) \triangleq \mathbb{E}_{\mathbf{v} \in \mathbb{B}}[f_t(\mathbf{w} + \delta \mathbf{v})]$.

⇨ Set $\mathbf{g}_t \triangleq \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$ as gradient estimator

Base Algorithm: Dynamic Regret

Bandit Gradient Descent (BGD)

for $t = 1$ to T **do**

Select a unit vector $\mathbf{s}_t$ uniformly at random

Submit $\mathbf{w}_t = \widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t$ and receive feedback $f_t(\mathbf{w}_t)$

Construct gradient estimator $\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$

Update $\widetilde{\mathbf{w}}_{t+1} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_t - \eta \mathbf{g}_t]$

end for

Theorem 1. With horizon T, step size η, and perturbation radius δ, BGD achieves the following expected dynamic regret bound:

$$\mathbb{E}\left[\text{D-REG}\right] \leq \mathcal{O}\left(\frac{1+P_T}{\eta} + \frac{\eta T}{\delta^2} + \delta T\right),$$

where $P_T = \sum_{t=2}^{T} \|\mathbf{u}_t - \mathbf{u}_{t-1}\|$ measures the non-stationarity level.

Optimal parameter setting is

- step size $\eta_* = \left(\frac{7R^2 + RP_T}{T}\right)^{\frac{3}{4}}$
- perturbation parameter $\delta_* = \eta_*^{\frac{1}{3}}$



Comparators $\mathbf{u}_1, \ldots, \mathbf{u}_T$ can be arbitrary, we cannot know non-stationarity P_T in advance, ***so how to tune the step size ?***

Online Ensemble for BCO

Deploying online ensemble to handle the issue of ***unknown non-stationarity***.

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i} \mathbf{w}_{t+1,i}$$

bandit *feedback*
→ *hard to initiate* ***multiple*** *base learners*

- **Multiple candidates:** to cover uncertainty

  diversity consideration: cover all the possible range using as fewer as possible discretization items



$$\mathcal{H} = \left\{ \eta_i = 2^{i-1} \frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots, N \right\}$$

$$\text{with } N = \left\lceil \log_2(1 + 2T/7) \right\rceil + 1 = \mathcal{O}(\log T).$$

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t^{\eta_i}]$$
$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **Meta algorithm:** provide the weight $\boldsymbol{p}_{t+1} \in \Delta_N$

  increase weight on base-learners with better performance

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i} \exp(-\varepsilon f_t(\mathbf{w}_{t,i}))$$

Multiple base learners in BCO

- A closer look at dynamic regret analysis

$$\widetilde{\mathbf{w}}_{t+1} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_t - \eta \mathbf{g}_t]$$

$$\mathbb{E}[\mathbf{g}_t] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t)$$

$\mathbf{v}_t = (1-\alpha)\mathbf{u}_t$ rescaled comparator

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)$$

$$= \underbrace{\sum_{t=1}^{T} \widehat{f}_t(\widetilde{\mathbf{w}}_t) - \sum_{t=1}^{T} \widehat{f}_t(\mathbf{v}_t)}_{\texttt{term (a)}} + \underbrace{\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} \widehat{f}_t(\widetilde{\mathbf{w}}_t)}_{\texttt{term (b)}} + \underbrace{\sum_{t=1}^{T} \widehat{f}_t(\mathbf{v}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t)}_{\texttt{term (c)}}$$

term (a): *depends on* P_T

term (b): $\leq 2L\delta T$

term (c): $\leq (L\delta + L\alpha R)T$

crucial term, related to non-stationarity measure P_T

not involve the unknown non-stationarity measure P_T *(approximation error due to the perturbation operation)*

Multiple base learners in BCO

- Key quantity: dynamic regret of the smoothed function

$$\sum_{t=1}^{T} \widehat{f}_t(\widetilde{\mathbf{w}}_t) - \sum_{t=1}^{T} \widehat{f}_t(\mathbf{v}_t)$$

surrogate loss $\ell_t(\mathbf{w}) \triangleq \langle \mathbf{g}_t, \mathbf{w} \rangle$

- Key idea: ***surrogate optimization***

Proposition 1. $\mathbb{E}[\widehat{f}_t(\widetilde{\mathbf{w}}_t) - \widehat{f}_t(\mathbf{v}_t)] \leq \mathbb{E}[\langle \mathbf{g}_t, \widetilde{\mathbf{w}}_t - \mathbf{v}_t \rangle]$, where $\mathbf{g}_t = \frac{d}{\delta} f_t(\widetilde{\mathbf{w}}_t + \delta \mathbf{s}_t) \cdot \mathbf{s}_t$ is the gradient estimator (satisfying $\mathbb{E}[\mathbf{g}_t] = \nabla \widehat{f}_t(\widetilde{\mathbf{w}}_t)$).

Proposition 2. $\nabla \ell_t(\mathbf{w}) = \mathbf{g}_t$ holds for any $\mathbf{w} \in \mathcal{W}$.

$\Rightarrow$ All the gradients $\nabla \ell_t(\widetilde{\mathbf{w}}_t^1) = \nabla \ell_t(\widetilde{\mathbf{w}}_t^2) = \cdots = \nabla \ell_t(\widetilde{\mathbf{w}}_t^N) = \mathbf{g}_t$, so they can be obtained by querying the function value of f_t ***only once***.

Online Ensemble for BCO

$$\mathbf{w}_{t+1} = \sum_{i=1}^{N} p_{t+1,i}\mathbf{w}_{t+1,i}$$

- **Multiple candidates:** to cover uncertainty

$$\mathcal{H} = \left\{ \eta_i = 2^{i-1}\frac{\sqrt{7}R}{dCT^{3/4}} \mid i = 1, \ldots, N \right\}$$

surrogate loss makes online ensemble possible in bandit!

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t^{\eta_i}]$$

$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **Meta algorithm:** provide weight $\boldsymbol{p}_{t+1} \in \Delta_N$

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i}\exp(-\varepsilon f_t(\mathbf{w}_{t,i}))$$

bandit feedback → hard to initiate multiple base learners

- **Base learners:** each updated using $\eta_i \in \mathcal{H}$

$$\text{BGD}(\eta_i): \quad \widetilde{\mathbf{w}}_{t+1,i} = \Pi_{(1-\alpha)\mathcal{W}}[\widetilde{\mathbf{w}}_{t,i} - \eta_i \mathbf{g}_t]$$

$$\mathbf{w}_{t+1,i} = \widetilde{\mathbf{w}}_{t+1,i} + \delta \boldsymbol{s}_t$$

- **Meta algorithm:** provide weight $\boldsymbol{p}_{t+1} \in \Delta_N$

$$\text{Hedge:} \quad p_{t+1,i} \propto p_{t,i}\exp(-\varepsilon \ell_t(\mathbf{w}_{t,i}))$$

Dynamic Regret

Theorem 3. *Under certain standard assumptions, with a proper setting of the pool of candidate step sizes $\mathcal{H}$ and the learning rate ϵ for the meta-algorithm, our* PBGD *algorithm enjoys the following expected dynamic regret guarantees.*

- *For the one-point feedback model,* $\mathbb{E}[\text{D-REG}_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)] \leq \mathcal{O}(T^{\frac{3}{4}}(1 + P_T)^{\frac{1}{2}})$.
- *For the two-point feedback model,* $\mathbb{E}[\text{D-REG}_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)] \leq \mathcal{O}(T^{\frac{1}{2}}(1 + P_T)^{\frac{1}{2}})$.

Lower bound: We further establish an $\Omega(\sqrt{TP_T})$ lower bound for BCO.

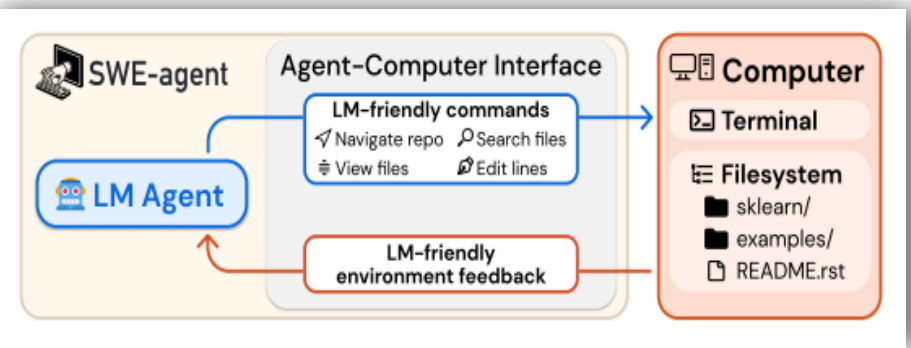
Algorithmic ingredients: online ensemble structure, and optimizing over proper ***surrogate loss*** is essential for deploying it in bandit online problems.

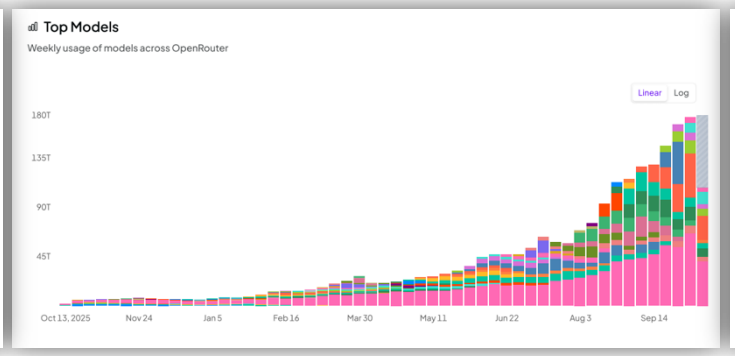
Outline

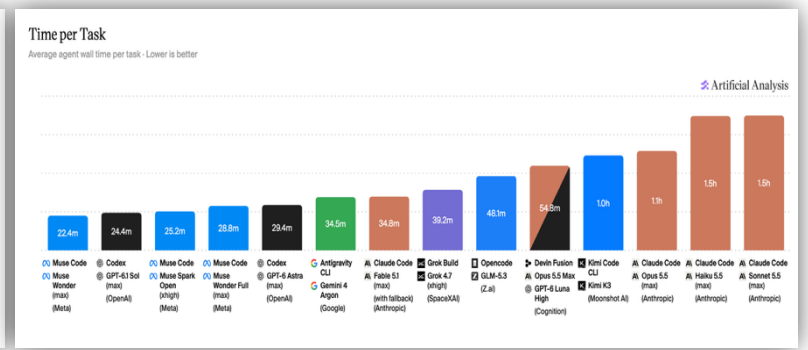
- Online Learning
- Speculative Decoding
- Conclusion

LLM 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



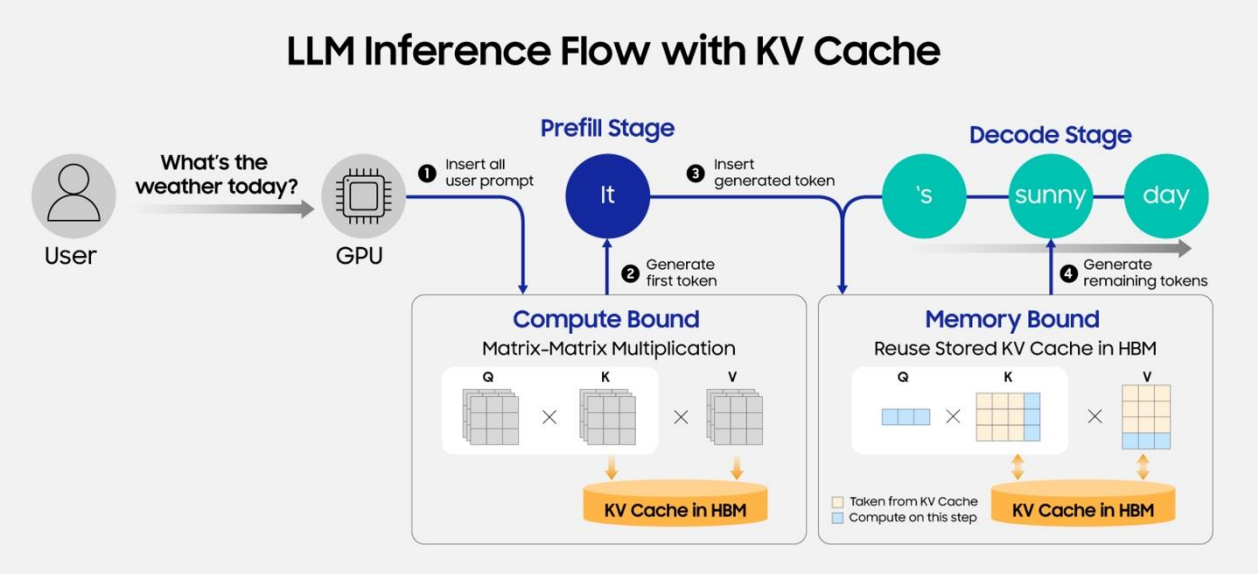


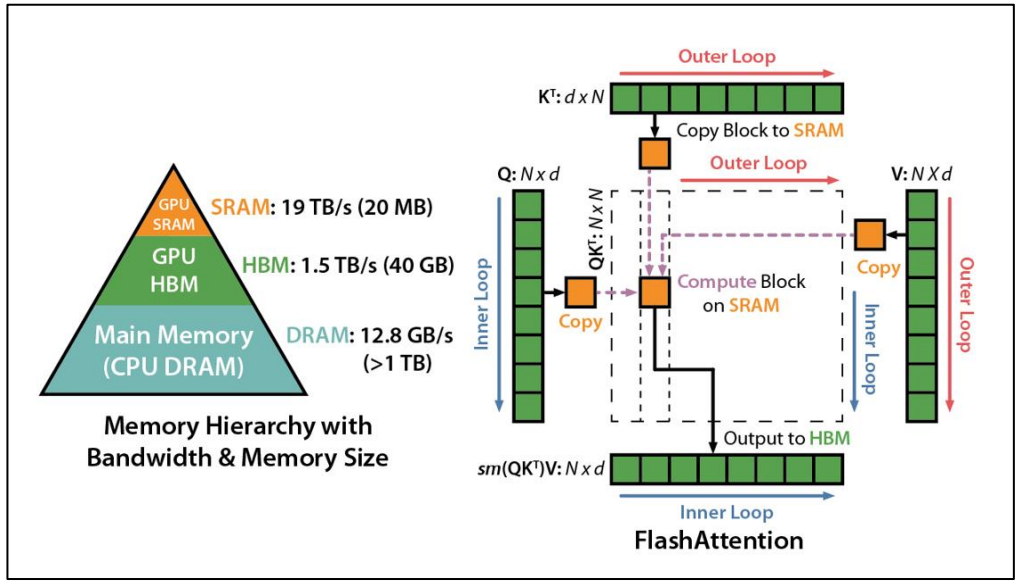


Inference cost and inference latency of LLMs lead to a decline in user experience.

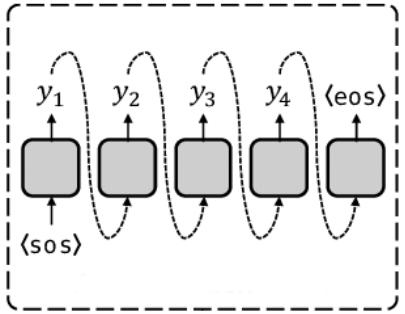
Latency Bottleneck of LLM Inference

- Autoregressive LLMs generate one token per forward pass.
- Decoding phase is ***memory-bound***, leaving GPU compute under-exploited.

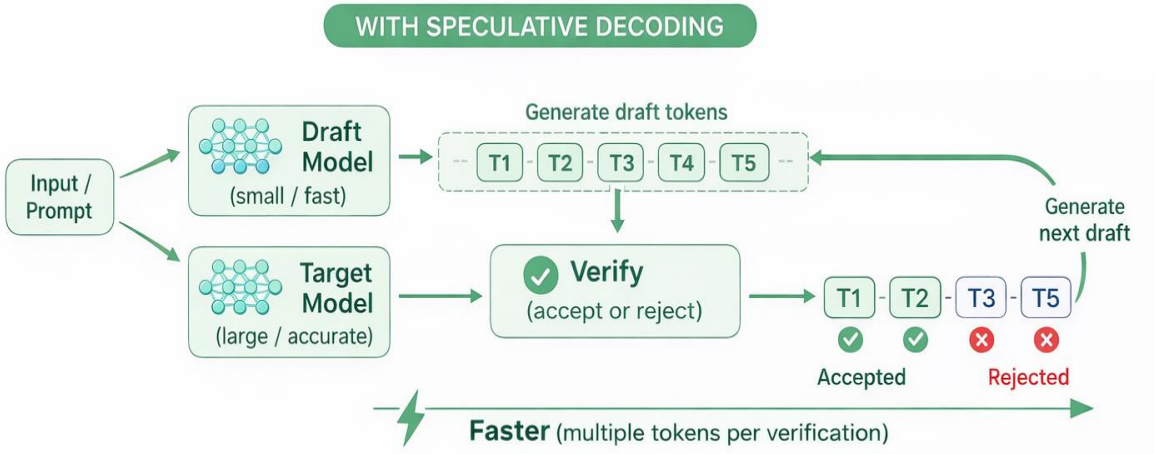


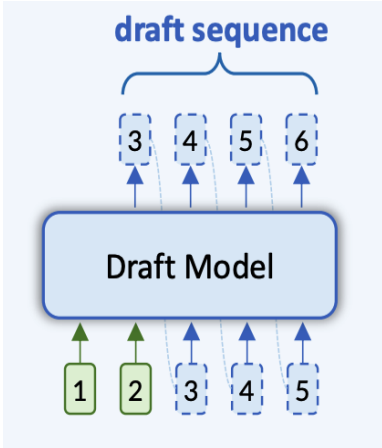


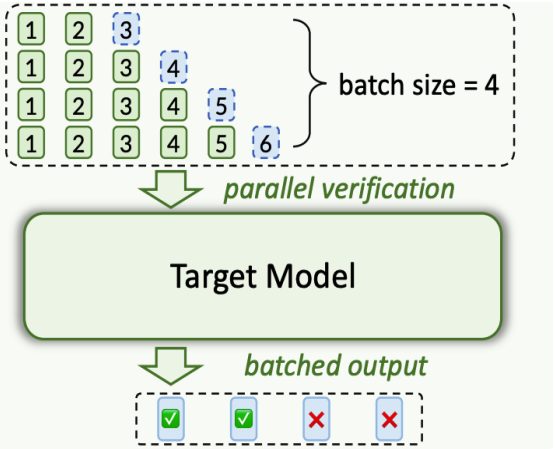
Speculative Decoding



- Autoregressive(AR) models speedup bottleneck
  - *Sequential* dependency
- **Speculative Decoding:** LLM inference acceleration methods :
  - a small *"draft model"*: first quickly generate tokens as drafts;
  - a big *"target model"*: ***parallelly*** verify the drafts.







Speculative Decoding

- The average latency per generated token is

$$L = \frac{T_{\text{draft}} + T_{\text{verify}}}{\tau}$$

$\tau :=$ number of accepted tokens per cycle

$T_{\text{draft}} :=$ wall-clock times of drafting passes per cycle

$T_{\text{verify}} :=$ wall-clock times of verification per cycle

- Improving speedup reduces to ***three*** levers:
  - lowering T_{draft} (***draft faster***)
  - raising τ (***draft better***)
  - reducing the effective T_{verify} (***verify smarter***)

Algorithm 1 Speculative Decoding

Require: Prefix $x_{<t}$, target model p, draft model q, block size K
Ensure: Samples distributed according to p

for $i = 1, \ldots, K$ **do**
  $x_{t+i-1} \sim q(\cdot \mid x_{<t+i-1})$
end for
Evaluate $p_1, \ldots, p_{K+1}$ with one target-model forward pass
for $i = 1, \ldots, K$ **do**
  Draw $r_i \sim U(0,1)$
  if $r_i > \min\left(1, \frac{p_i(x_{t+i-1})}{q_i(x_{t+i-1})}\right)$ **then**
    Sample $x \sim \text{norm}((p_i - q_i)^+)$
    return $x_t, \ldots, x_{t+i-2}, x$
  end if
end for
Sample $x_{t+K} \sim p_{K+1}$
return $x_t, \ldots, x_{t+K}$

Lossless speedup

Leviathan et al., Fast Inference from Transformers via Speculative Decoding. ICML 2023.

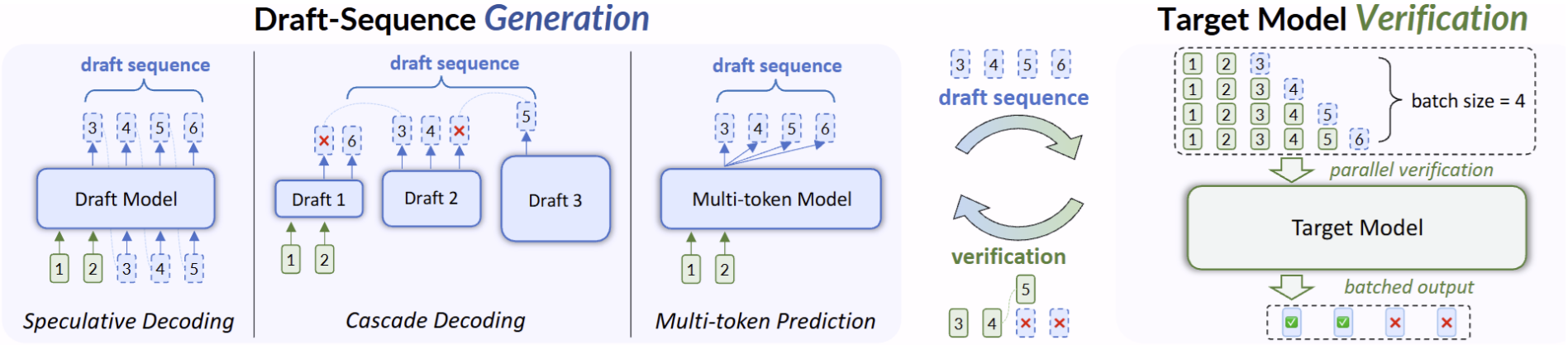
When Drafts Evolve: Speculative Decoding Meets Online Learning

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

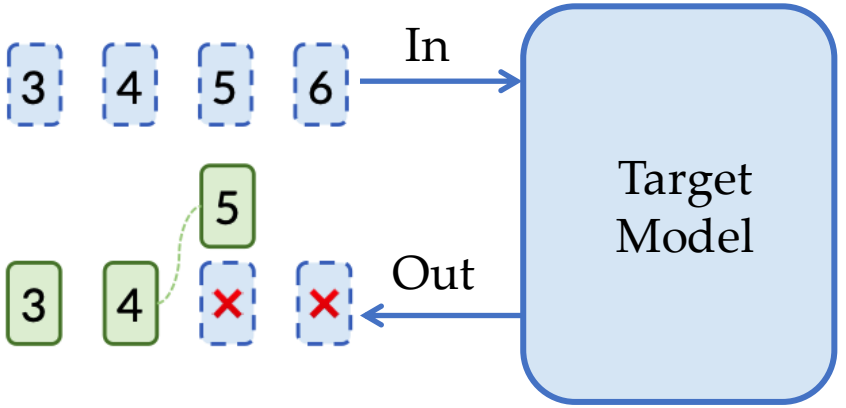
ICML International Conference On Machine Learning
Seoul, South Korea

Observation: Free Feedback

- Feedbacks in speculative decoding framework



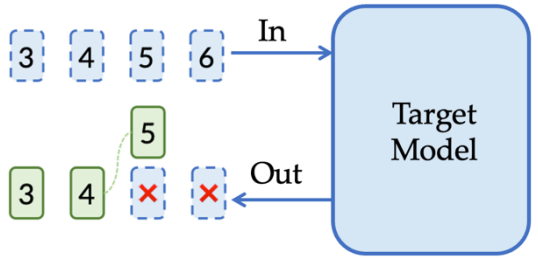
- One round of speculative decoding in *draft model's view*:
  - Submits: *k* tokens
  - Gets: *k* labels denote accept or not



Draft model receives timely feedback ***at no cost!***

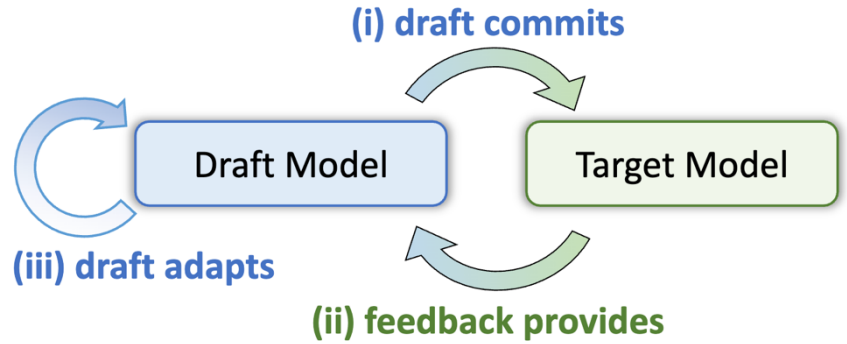
Observation: Generation-Refinement

- One round of speculative decoding in ***draft model's view***:
  - Submits: k tokens
  - Gets: k labels denote accept or not



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

- We can use these feedback to update draft model(s), exactly match ***Online Learning.***



Speculative decoding belongs to ***Generation-Refinement*** framework

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

OnlineSpec: A Unified Framework

- Formulated as an *online learning* problem:

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

(1) 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) 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**

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

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

OnlineSpec: A Unified Framework

- Formulated as an *online learning* problem:

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

(1) 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) 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

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

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**

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

else

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

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

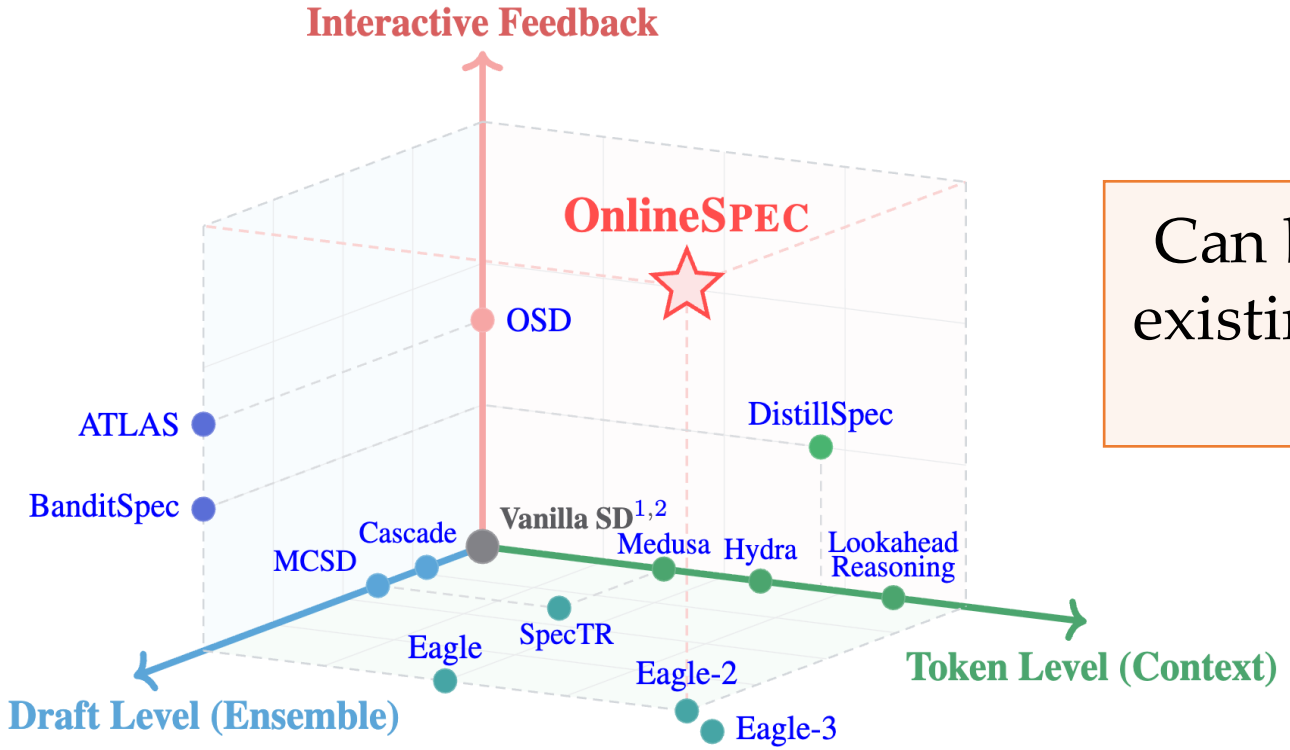
▷ Receive interactive feedback: observe the loss function $f_t : \mathcal{W} \mapsto \mathbb{R}$ from the target model $\mathbf{v}$.

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

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

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***:

Instantiation	+Feedback via	Regret bound	Acceleration rate
Online-LR	Online gradient descent $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)$	$\mathbf{Reg}_T \leq \mathcal{O}\left(\sqrt{T(1+P_T)}\right)$	$\gamma = \Omega\left(\frac{1}{\alpha k+1} \min\left\{k+1, \frac{T^{1/4}}{\sqrt{1+P_T}}\right\}\right)$
Opt-Hydra	Optimistic online learning $\mathbf{w}_t = \widehat{\mathbf{w}}_t - \eta \mathbf{h}_t; \widehat{\mathbf{w}}_{t+1} = \widehat{\mathbf{w}}_t - \nabla f_t(\mathbf{w}_t)$	$\mathbf{Reg}_T \leq \mathcal{O}\left(\sqrt{1+\delta_T} \cdot (1+P_T)\right)$	$\gamma = \Omega\left(\frac{1}{\alpha k+1} \min\left\{k+1, \frac{\sqrt{T}}{(1+\delta_T)^{1/4}\sqrt{1+P_T}}\right\}\right)$
Ens-Eagle	Online ensemble learning $\mathbf{w}_t = \sum_{i=1}^N p_t^i \cdot \mathbf{w}_t^i$	$\mathbf{Reg}_T \leq \mathcal{O}\left(\sqrt{T(1+P_T)} + \sqrt{T \log\log T}\right) = \mathcal{O}\left(\sqrt{T(1+P_T)}\right)$	$\gamma = \Omega\left(\frac{1}{\alpha k+1} \min\left\{k+1, \frac{T^{1/4}}{(1+P_T)^{1/4}}\right\}\right)$

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**

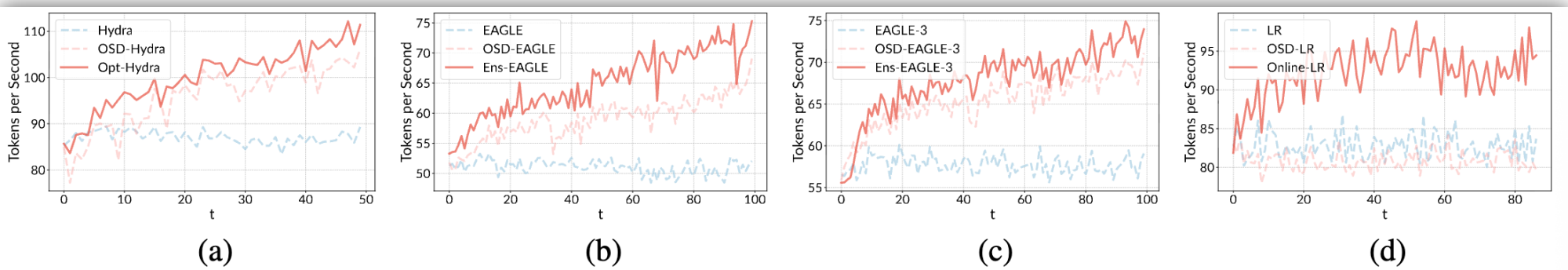


Figure 3. Evolution of tokens per second (TPS) on the *GSM8K* dataset: (a) *Opt-Hydra* with *lmsys/Vicuna-7B-v1.3*, (b) *Ens-EAGLE* with *lmsys/Vicuna-7B-v1.3*, (c) *Ens-EAGLE-3* with *lmsys/Vicuna-7B-v1.3*, and (d) *Online-LR* with *Qwen/Qwen3-8B*. This demonstrates consistent performance improvements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

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

Scaling to larger model

Model	Method	SPEEDUP ↑	AVGLEN ↑	ACC (%)
Vicuna-13B	Standard AR	1.00 (40.40)	1.00	–
	Vanilla SD	1.20 (48.52)	1.93	–
	OSD-EAGLE-3	1.37 (55.52)	2.24	–
	Ens-EAGLE-3	**1.46** (59.03)	**2.35**	–
Vicuna-13B	Standard AR	1.00 (40.40)	1.00	–
	Vanilla SD	1.50 (60.61)	2.22	–
	OSD-Hydra	1.69 (68.36)	2.50	–
	Opt-Hydra	**1.84** (74.37)	**2.73**	–
Qwen3-32B	Standard AR	1.00 (35.53)	1.00	96.08
	LR	1.11 (39.39)	3.66	95.76
	OSD-LR	1.09 (38.63)	3.18	95.44
	Online-LR	**1.31** (46.71)	**9.98**	95.68

🚀 When scale to 13B/32B model, ***OnlineSPEC*** still ***outperforms*** other methods.

AdaFlash

AdaFlash: Adaptive Speculative Decoding via On-Policy Distilled Diffusion Drafters

Yu-Yang Qian[1,2,*] **Hao-Cong Wu**[1,2,*] **Chen Chen**[3] **Jiacheng Sun**[3] **Zhenhua Dong**[3]
Peng Zhao[1,2,†] **Zhi-Hua Zhou**[1,2]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Artificial Intelligence, Nanjing University [3]Huawei Foundation Model Dept



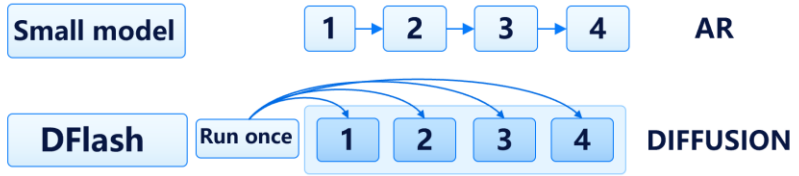




COLM 2026 Workshop on Efficient Reasoning

AdaFlash

DFlash [Chen et al., ICML'26]: lightweight diffusion drafter to speedup the system inference



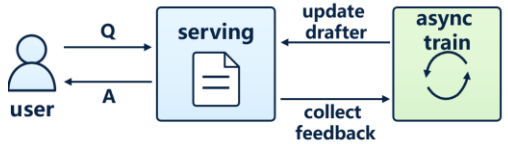
- **Challenge 1: Adaptability Across Tasks**

  No model updates after deployment
  → unstable acceleration on new tasks

- **Challenge 2: High Concurrency Issue**

  concurrent requests increase system load
  → reducing speedup; even worse than MTP

On-Policy Distillation for diffusion drafter



target feedback to update serving drafter; Enable asynchronous training and hot weight loading.

Lightweight Dynamic Verification Length



A lightweight head uses draft features to predict acceptance rate and set prefix length to verify; adjust budget by system load.

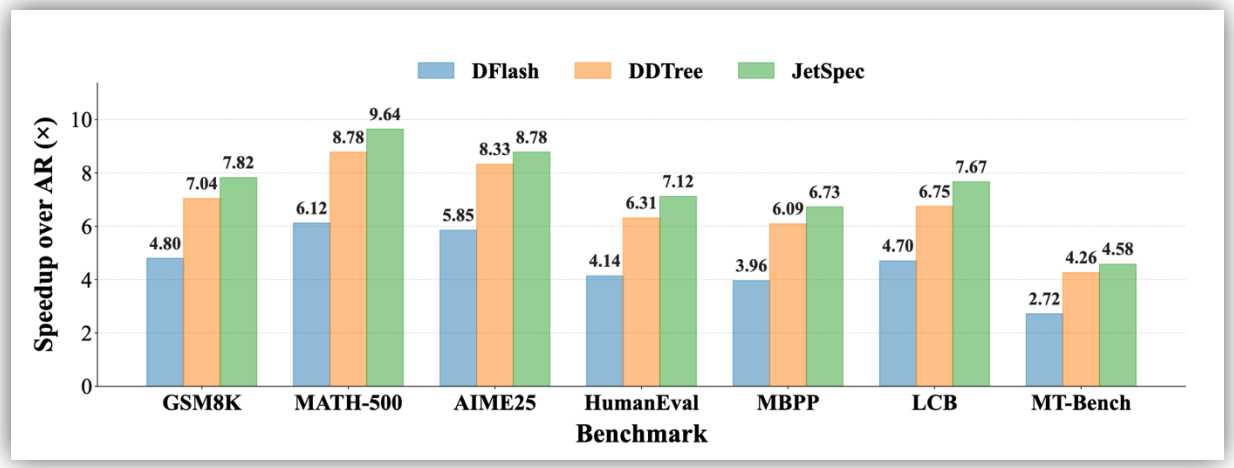
🚀 ***with up to 5.32× speedup over Dflash; Supports Huawei Ascend platform.***

Y.-Y. Qian, H.-C. Wu, C. Chen, J. Sun, Z. Dong, P. Zhao, Z.-H. Zhou. AdaFlash: Adaptive Speculative Decoding via On-Policy Distilled Diffusion Drafters. ***COLM Workshop (Spotlight)***, 2026.

JetSpec/JetSpec++

- [arXiv:2606.18394] JetSpec: Breaking the Scaling Ceiling of Speculative Decoding with Parallel Tree Drafting.

Same time as DSpark (actually 2weeks earlier, 6.25 vs 7.6)



不只DeepSeek，阶跃等开源JetSpec：大模型解码提速近10倍

机器之心 2026年6月30日 15:38 36人

Minds in AI.

机器之心发布

近期，DeepSeek 推出投机解码框架 DSpark，让大模型推理效率再次成为行业焦点。

几乎同一时间，另一大模型基座代表阶跃星辰提出了 JetSpec，也把问题指向了同一个方向：

当模型开始被 Agent 高频调用，智能能不能更快、更稳定输出出来?

Dflash → ”causality” matters, and JetSpec introduces a causal parallel decoding head for tree drafting.

PD Survey

- https://github.com/ZinYY/Awesome-Parallel-Decoding

README

A Survey of Parallel Decoding

Awesome-Parallel-Decoding

awesome | Stars 9 | License CC-BY-NC-ND 4.0 | PRs Welcome | last commit september

A curated list of **Awesome Parallel Decoding** papers to accelerate the inference of (autoregressive) LLMs.

Autoregressive (AR) decoding generates one token per forward pass, leaving the strict token-by-token dependency as the dominant latency bottleneck of modern LLM serving. **Parallel decoding** refers to the broad family of techniques that shorten this critical path by introducing *parallel candidates*, *parallel verification*, *parallel reasoning steps*, or *parallel task orchestration*, while ideally preserving the quality (and, for lossless methods, the exact output distribution) of the original model. This repository organizes the literature along the **granularity of parallelism** — from (i) individual tokens, to (ii) reasoning sequences, to (iii) whole tasks — and additionally tracks the (iv) **model architecture-level** and (v) **system-level** improvements. We also introduce (vi) **hybrid** advances and (vii) **related benchmarks** that make parallel decoding practical.

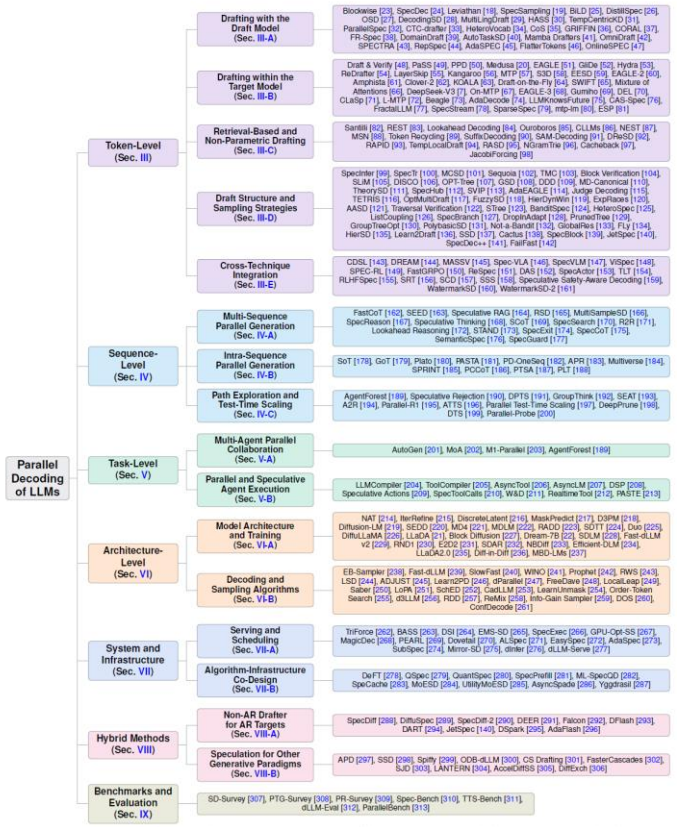


Fig. 1: Taxonomy of parallel decoding methods. We organize existing work by the *granularity* at which parallelism is introduced, i.e., from individual tokens, through sequences and tasks, to model architectures, and further include system-level infrastructure, hybrid methods, and benchmarks. Each granularity level is subdivided into finer categories based on how parallelism is realized within that level.

Conclusion

- **Online Learning**
  - A very powerful framework for interactive system
  - Online Ensemble: an effective theoretical framework (base learners; meta learners; schedule) to handle *uncertainty* in online environments
- **Speculative Decoding**
  - *OnlineSpec*: introduce online learning technique to improve the throughput, connecting regret and acceptance rate
  - *AdaFlash*: on-policy distillation and high-concurrent scenarios
- Future directions: agentic speculative decoding, model router…

References

- Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and Non-stationarity: Problem-dependent Dynamic Regret for Online Convex Optimization. *JMLR 2024*.
- Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit Convex Optimization in Non-stationary Environments. *JMLR 2021*.
- Yu-Yang Qian, Hao-Cong Wu, Yichao Fu, Hao Zhang, and Peng Zhao. When Drafts Evolve: Speculative Decoding Meets Online Learning. *ICML 2026*.
- Yu-Yang Qian*, Hao-Cong Wu*, Chen Chen, Jiacheng Sun, Zhenhua Dong, Peng Zhao, and Zhi-Hua Zhou. AdaFlash: Adaptive Speculative Decoding via On-Policy Distilled Diffusion Drafters. *COLM 2026 Workshop on Efficient Reasoning*.

NANJING UNIVERSITY HUAWEI StepFun

Thanks!