# BanditSpec: Adaptive Speculative Decoding via Bandit Algorithms (ICML 2025)

Yunlong Hou*, Fengzhuo Zhang*, Cunxiao Du*, Xuan Zhang*, Jiachun Pan, Tianyu Pang, Chao Du, **Vincent Y. F. Tan**, Zhuoran Yang

August 25, 2025
CNI Seminar, IISc

# Background: Canonical Decoding

- **Canonical Decoding:** tokens are generated in an autoregressive manner: each token requires a full forward pass through the massive target model $P : \mathcal{X}^* \to \Delta_{\mathcal{X}}$.

# Background: Canonical Decoding

- **Canonical Decoding:** tokens are generated in an autoregressive manner: each token requires a full forward pass through the massive target model $P : \mathcal{X}^* \to \Delta_{\mathcal{X}}$.

**Algorithm 2** CANONICAL DECODING

**Inputs:** initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, target model $P$.

**Procedures:**

1: Set $t = 0$.
2: **while** $t \neq 0$ and $x_t \neq \mathrm{EOS}$ **do**
3: $t = t + 1$.
4: $x_t \sim P(\cdot \mid \mathrm{pt}_{t-1})$.
5: $\mathrm{pt}_t = \mathrm{concat}(\mathrm{pt}_{t-1}, x_t)$.
6: **end while**
7: **return** $\tau_c = t$, $\mathrm{pt}_{\tau_c} = \mathrm{pt}_t$

# Background: Canonical Decoding

0 : [BOS]
1 : [BOS] Adaptive
2 : [BOS] Adaptive speculative
3 : [BOS] Adaptive speculative decoding
4 : [BOS] Adaptive speculative decoding via
5 : [BOS] Adaptive speculative decoding via bandit
6 : [BOS] Adaptive speculative decoding via bandit algorithm
7 : [BOS] Adaptive speculative decoding via bandit algorithm is
8 : [BOS] Adaptive speculative decoding via bandit algorithm is more
9 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient
10 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient .
11 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient . [EOS]

- **Total Time:**

$$T_{\text{total}} = T_{\text{target}} \times \tau_{\text{c}} = 30\,\text{ms} \times 11 = 330\,\text{ms}.$$

How can we accelerate this one-by-one token generation process?

[0] $T_{\text{target}} \approx 30$ ms for LLaMA3-8B.

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

- **Step** $1$**:** **Draft** the next $L$ tokens via draft model $Q$ *autoregressively*

$$Q\left(\cdot \mid \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right), \quad j = 0, \ldots, L-1.$$

  - Draft models are small and fast (e.g., T5-small (77M)).
  - Maybe inaccurate.

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** → **Verify** → **Accept**

- **Step** 1: **Draft** the next $L$ tokens via draft model $Q$ *autoregressively*

$$Q\left(\cdot \;\middle|\; \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right), \quad j = 0, \ldots, L-1.$$

  - Draft models are small and fast (e.g., T5-small (77M)).
  - Maybe inaccurate.

[BOS] Adaptive speculative decoding via
[BOS] Adaptive speculative decoding via suffix
[BOS] Adaptive speculative decoding via suffix tree
[BOS] Adaptive speculative decoding via suffix tree way

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** → **Verify** → **Accept**

- **Step $2$: Verify** the drafted tokens: compute the probabilities of the outputs via the target LLM *in parallel*

$$P\left(\tilde{x}_{t+j+1} \,\middle|\, \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right), \quad j = 0, \ldots, L-1.$$

[BOS] Adaptive speculative decoding via
[BOS] Adaptive speculative decoding via suffix
[BOS] Adaptive speculative decoding via suffix tree
[BOS] Adaptive speculative decoding via suffix tree way

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

- **Step** $3$**: Accept** the drafted tokens **sequentially**.
  - Accept $\tilde{x}_{t+j+1}$ with probability

$$\min\left\{1\ ,\ \frac{P\left(\tilde{x}_{t+j+1} \mid \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right)}{Q\left(\tilde{x}_{t+j+1} \mid \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right)}\right\},\ j = 0, \ldots, L-1.$$

[BOS] Adaptive speculative decoding via

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

- **Step** $3$**: Accept** the drafted tokens **sequentially**.
  - Accept $\tilde{x}_{t+j+1}$ with probability

$$\min\left\{1\ ,\ \frac{P\left(\tilde{x}_{t+j+1} \mid \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right)}{Q\left(\tilde{x}_{t+j+1} \mid \mathrm{pt}_{t-1}, \tilde{x}_{t:t+j}\right)}\right\},\ j = 0, \ldots, L-1.$$

[BOS] Adaptive speculative decoding via ✓

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

- **Step** $3$**:** **Accept** the drafted tokens **sequentially**.
  - Accept the draft tokens until the first rejected one (e.g. $\tilde{x}_i$).

[BOS] Adaptive speculative decoding via ✓

[BOS] Adaptive speculative decoding via suffix

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** $\rightarrow$ **Verify** $\rightarrow$ **Accept**

- **Step** $3$**:** **Accept** the drafted tokens **sequentially**.
  - Accept the draft tokens until the first rejected one (e.g. $\tilde{x}_i$).

[BOS] Adaptive speculative decoding via ✓
[BOS] Adaptive speculative decoding via suffix ✗

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** → **Verify** → **Accept**

- **Step** $3$**:** **Accept** the drafted tokens **sequentially**.
  - Accept the draft tokens until the first rejected one (e.g. $\tilde{x}_i$).
  - Correct $\tilde{x}_i$ by $x_i \sim \text{Norm}\left(\left[P(\cdot \mid \text{pt}, \tilde{x}_{1:i-1}) - Q(\cdot \mid \text{pt}, \tilde{x}_{1:i-1})\right]_+\right)$.

[BOS] Adaptive speculative decoding via ✓

[BOS] Adaptive speculative decoding via suffix ×

[BOS] Adaptive speculative decoding via ~~suffix~~ bandit

~~[BOS] Adaptive speculative decoding via suffix tree~~

~~[BOS] Adaptive speculative decoding via suffix tree way~~

# Preliminary: Speculative Decoding

**Speculative Decoding:** Accelerate inference of LLMs while maintaining high generation quality (Chen et al., 2023; Leviathan et al., 2023).

**Draft** → **Verify** → **Accept**

Input: [BOS] Adaptive speculative decoding

[BOS] Adaptive speculative decoding via suffix tree way

---

[BOS] Adaptive speculative decoding via

[BOS] Adaptive speculative decoding via suffix

[BOS] Adaptive speculative decoding via suffix tree

[BOS] Adaptive speculative decoding via suffix tree way

---

[BOS] Adaptive speculative decoding via ✓

[BOS] Adaptive speculative decoding via suffix ✗

[BOS] Adaptive speculative decoding via ~~suffix~~ bandit

⇒ Output: [BOS] Adaptive speculative decoding via bandit

# Preliminary: Speculative Decoding

**Canonical Decoding:**

0 : [BOS]
1 : [BOS] Adaptive
2 : [BOS] Adaptive speculative
3 : [BOS] Adaptive speculative decoding
4 : [BOS] Adaptive speculative decoding via
5 : [BOS] Adaptive speculative decoding via bandit
6 : [BOS] Adaptive speculative decoding via bandit algorithm
7 : [BOS] Adaptive speculative decoding via bandit algorithm is
8 : [BOS] Adaptive speculative decoding via bandit algorithm is more
9 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient
10 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient .
11 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient . [EOS]

**Speculative Decoding:**

0 : [BOS]
1 : [BOS] Adaptive speculative ~~sampling~~ ~~is~~ decoding
2 : [BOS] Adaptive speculative decoding via ~~suffix~~ ~~tree~~ ~~way~~ bandit
3 : [BOS] Adaptive speculative decoding via bandit algorithm is ~~definitely~~ more
4 : [BOS] Adaptive speculative decoding via bandit algorithm is more efficient . [EOS]

# Preliminary: Speculative Decoding

- For one round of speculative decoding:

$$T_{\text{spec}} = \tau_{\text{draft}} \times L + T_{\text{target}} + \tau_{\text{accept}} \times n_{\text{accepted}} \approx T_{\text{target}}$$

# Preliminary: Speculative Decoding

- For one round of speculative decoding:

$$T_{\text{spec}} = \tau_{\text{draft}} \times L + T_{\text{target}} + \tau_{\text{accept}} \times n_{\text{accepted}} \approx T_{\text{target}}$$

- Total time saved:

$$T_{\text{target}} \times \mathbb{E}\left[ \tau_{\text{c}} - \tau_{\text{spec}} \right]$$

$\tau_{\text{c}}$ : the number of canonical decoding rounds.

$\tau_{\text{spec}}$ : the number of speculative decoding rounds.

# Preliminary: Speculative Decoding

- For one round of speculative decoding:

$$T_{\text{spec}} = \tau_{\text{draft}} \times L + T_{\text{target}} + \tau_{\text{accept}} \times n_{\text{accepted}} \approx T_{\text{target}}$$

- Total time saved:

$$T_{\text{target}} \times \mathbb{E}\left[\tau_{\text{c}} - \tau_{\text{spec}}\right]$$

$\tau_{\text{c}}$ : the number of canonical decoding rounds.

$\tau_{\text{spec}}$ : the number of speculative decoding rounds.

- The distribution of the generated token sequence is unbiased (Leviathan et al., 2023; Chen et al., 2023; Yin et al., 2024):

$$\text{pt}_{\tau_{\text{spec}}} \overset{d}{=} \text{pt}_{\tau_c}.$$

# Preliminary: Speculative Decoding

- Most existing speculative decoding methods use a fixed configuration.
  - A single draft model or a fixed set of hyperparameters is used for all tasks.
  - This is suboptimal because the ideal configuration depends heavily on the specific context.

# Preliminary: Speculative Decoding

- Most existing speculative decoding methods use a fixed configuration.
  - A single draft model or a fixed set of hyperparameters is used for all tasks.
  - This is suboptimal because the ideal configuration depends heavily on the specific context.

| **Task** | **Best approach** |
|---|---|
| Code Debugging | Requires precision; a retrieval-based method like Suffix Tree (Oliaro et al., 2024) might be best. |
| Story Generation | Requires creativity; a smaller draft LLM like Eagle (Li et al., 2024b) might be better. |

# Preliminary: Speculative Decoding

- Most existing speculative decoding methods use a fixed configuration.
  - A single draft model or a fixed set of hyperparameters is used for all tasks.
  - This is suboptimal because the ideal configuration depends heavily on the specific context.

| **Task** | **Best approach** |
|---|---|
| Code Debugging | Requires precision; a retrieval-based method like Suffix Tree (Oliaro et al., 2024) might be best. |
| Story Generation | Requires creativity; a smaller draft LLM like Eagle (Li et al., 2024b) might be better. |

Is there any *training-free* method that can *adaptively* choose the hyperparameters such that the latency of speculative decoding can be minimized?

# Preliminary: Bandit Model

**Algorithm 3** Dynamics of Multi-Armed Bandits

**Inputs:** $K$ arms, time horizon $T$.
$\mathcal{H}_0 = \emptyset$.
**for** $t = 1, 2, \ldots, T$ **do**
  Agent adopts an algorithm `ALG` to select arm $I_t$ based on $\mathcal{H}_{t-1}$.
  Environment reveals the reward $X_{I_t,t}$ to the agent.
  Agent updates the history $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, X_{I_t,t}))$.
**end for**

**Goal:** Devise an algorithm `ALG` to minimize the cumulative regret

$$\max_{i \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t}\right] - \mathbb{E}_{\texttt{ALG}}\left[\sum_{t=1}^{T} X_{I_t,t}\right].$$

# Preliminary: Bandit Model

**Algorithm 4** Dynamics of Multi-Armed Bandits

**Inputs:** $K$ arms, time horizon $T$.
$\mathcal{H}_0 = \emptyset$.
**for** $t = 1, 2, \ldots, T$ **do**
  Agent adopts an algorithm `ALG` to select arm $I_t$ based on $\mathcal{H}_{t-1}$.
  Environment reveals the reward $X_{I_t,t}$ to the agent.
  Agent updates the history $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, X_{I_t,t}))$.
**end for**

**Goal:** Devise an algorithm `ALG` to minimize the cumulative regret

$$\max_{i \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t}\right] - \mathbb{E}_{\texttt{ALG}}\left[\sum_{t=1}^{T} X_{I_t,t}\right].$$

# Preliminary: Bandit Model

**Algorithm 5** Dynamics of Multi-Armed Bandits

1: **Inputs:** $K$ arms, time horizon $T$.
2: $\mathcal{H}_0 = \emptyset$.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Agent adopts an algorithm `ALG` to select arm $I_t$ based on $\mathcal{H}_{t-1}$.
5: Environment reveals the reward $X_{I_t,t}$ to the agent.
6: Agent updates the history $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, X_{I_t,t}))$.
7: **end for**

**Goal:** Devise an algorithm `ALG` to minimize the cumulative regret

$$\max_{i \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t}\right] - \mathbb{E}_{\texttt{ALG}}\left[\sum_{t=1}^{T} X_{I_t,t}\right].$$

# Preliminary: Bandit Model

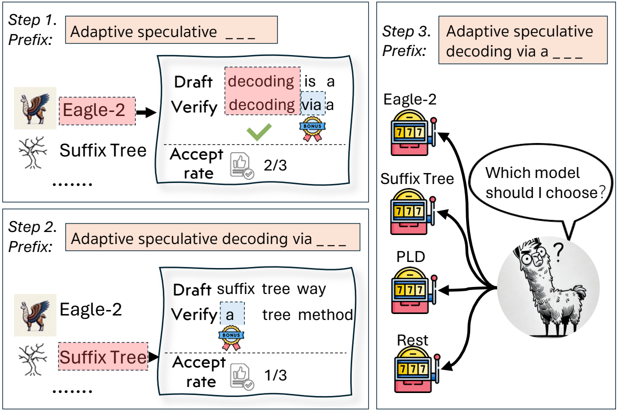
**Algorithm 6** Dynamics of Multi-Armed Bandits

```
Inputs: K arms, time horizon T.
H_0 = ∅.
for t = 1, 2, ..., T do
    Agent adopts an algorithm ALG to select arm I_t based on H_{t-1}.
    Environment reveals the reward X_{I_t,t} to the agent.
    Agent updates the history H_t = concat(H_{t-1}, (I_t, X_{I_t,t})).
end for
```

**Goal:** Devise an algorithm ALG to minimize the cumulative regret

$$\max_{i \in [K]} \mathbb{E}\left[\sum_{t=1}^{T} X_{i,t}\right] - \mathbb{E}_{\texttt{ALG}}\left[\sum_{t=1}^{T} X_{I_t,t}\right].$$

# Problem Formulation




**Hyperparameter Selection Problem:** Each candidate hyperparameter configuration (e.g., draft model) is treated as an "arm".

# Bandit Framework for Speculative Decoding: BanditSpec

BanditSpec: A *training-free and adaptive* online learning framework for hyperparameter selection in speculative decoding.

---

**Inputs:** arm selection algorithm ALG, initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$.

**Procedures:**

$t = 0, \mathcal{H}_0 = \emptyset, I_0 = 1, \mathbf{x}_{I_0,0} = \emptyset$.
**while** $\mathrm{EOS} \notin \mathbf{x}_{I_t,t}$ **do**
    $t = t + 1$.
    Select a hyperparameter index $I_t = \mathtt{ALG}(\mathcal{H}_{t-1})$.
    Receive the accepted tokens $\mathbf{x}_{I_t,t} = \textsc{SpecDecSub}(\mathrm{pt}_{t-1}, P, S_{I_t}, L)$.
    Update $\mathrm{pt}_t = \mathrm{concat}(\mathrm{pt}_{t-1}, \mathbf{x}_{I_t,t})$, $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, \mathbf{x}_{I_t,t}))$.
**end while**
**return** $\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) = t,\ \mathrm{pt}_{\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu)} = \mathrm{pt}_t$.

---

$\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu)$: Number of calls to SpecDecSub.

# Bandit Framework for Speculative Decoding: BanditSpec

BanditSpec: A *training-free and adaptive* online learning framework for hyperparameter selection in speculative decoding.

**Inputs:** arm selection algorithm $\texttt{ALG}$, initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i\in[K]}, L)$.

**Procedures:**

$t = 0, \mathcal{H}_0 = \emptyset, I_0 = 1, \mathbf{x}_{I_0,0} = \emptyset$.
**while** $\mathrm{EOS} \notin \mathbf{x}_{I_t,t}$ **do**
    $t = t + 1$.
    Select a hyperparameter index $I_t = \texttt{ALG}(\mathcal{H}_{t-1})$.
    Receive the accepted tokens $\mathbf{x}_{I_t,t} = \textsc{SpecDecSub}(\mathrm{pt}_{t-1}, P, S_{I_t}, L)$.
    Update $\mathrm{pt}_t = \mathrm{concat}(\mathrm{pt}_{t-1}, \mathbf{x}_{I_t,t})$, $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, \mathbf{x}_{I_t,t}))$.
**end while**
**return** $\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = t$, $\mathrm{pt}_{\mathrm{ST}(\texttt{ALG},\mathrm{pt},\nu)} = \mathrm{pt}_t$.

$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu)$: Number of calls to SpecDecSub.

# Bandit Framework for Speculative Decoding: BanditSpec

BanditSpec: A *training-free and adaptive* online learning framework for hyperparameter selection in speculative decoding.

---

**Inputs:** arm selection algorithm ALG, initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$.

**Procedures:**

```
t = 0, H_0 = ∅, I_0 = 1, x_{I_0,0} = ∅.
while EOS ∉ x_{I_t,t} do
    t = t + 1.
    Select a hyperparameter index I_t = ALG(H_{t-1}).
    Receive the accepted tokens x_{I_t,t} = SpecDecSub(pt_{t-1}, P, S_{I_t}, L).
    Update pt_t = concat(pt_{t-1}, x_{I_t,t}), H_t = concat(H_{t-1}, (I_t, x_{I_t,t})).
end while
return ST(ALG, pt, ν) = t, pt_{ST(ALG,pt,ν)} = pt_t.
```

---

$\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu)$: Number of calls to SpecDecSub.

# Bandit Framework for Speculative Decoding: BanditSpec

BanditSpec: A *training-free and adaptive* online learning framework for hyperparameter selection in speculative decoding.

---

**Inputs:** arm selection algorithm $\texttt{ALG}$, initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$.

**Procedures:**

```
t = 0, H_0 = ∅, I_0 = 1, x_{I_0,0} = ∅.
while EOS ∉ x_{I_t,t} do
    t = t + 1.
    Select a hyperparameter index I_t = ALG(H_{t-1}).
    Receive the accepted tokens x_{I_t,t} = SpecDecSub(pt_{t-1}, P, S_{I_t}, L).
    Update pt_t = concat(pt_{t-1}, x_{I_t,t}), H_t = concat(H_{t-1}, (I_t, x_{I_t,t})).
end while
return ST(ALG, pt, ν) = t, pt_{ST(ALG,pt,ν)} = pt_t.
```

---

$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu)$: Number of calls to SpecDecSub.

# Bandit Framework for Speculative Decoding: BanditSpec

BanditSpec: A *training-free and adaptive* online learning framework for hyperparameter selection in speculative decoding.

---

**Inputs:** arm selection algorithm ALG, initial prompt $\mathrm{pt}_0 = \mathrm{pt} \in \mathcal{X}^*$, bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i\in[K]}, L)$.

**Procedures:**

$t = 0, \mathcal{H}_0 = \emptyset, I_0 = 1, \mathbf{x}_{I_0,0} = \emptyset$.
**while** $\mathrm{EOS} \notin \mathbf{x}_{I_t,t}$ **do**
  $t = t + 1$.
  Select a hyperparameter index $I_t = \mathtt{ALG}(\mathcal{H}_{t-1})$.
  Receive the accepted tokens $\mathbf{x}_{I_t,t} = \textsc{SpecDecSub}(\mathrm{pt}_{t-1}, P, S_{I_t}, L)$.
  Update $\mathrm{pt}_t = \mathrm{concat}(\mathrm{pt}_{t-1}, \mathbf{x}_{I_t,t})$, $\mathcal{H}_t = \mathrm{concat}(\mathcal{H}_{t-1}, (I_t, \mathbf{x}_{I_t,t}))$.
**end while**
**return** $\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) = t,\ \mathrm{pt}_{\mathrm{ST}(\mathtt{ALG},\mathrm{pt},\nu)} = \mathrm{pt}_t$.

---

$\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu)$: Number of calls to SpecDecSub.

# Bandit Framework for Speculative Decoding: BanditSpec

**Proposition (Property of BanditSpec)**

*For any arm selection algorithm* ALG *that selects an arm according to the history,*

$$\mathrm{pt}_{\mathrm{ST}(\texttt{ALG})} \stackrel{d}{=} \mathrm{pt}_{\tau_c}.$$

*Furthermore, the stopping time* $\mathrm{ST}(\texttt{ALG})$ *can be bounded as*

$$\frac{\mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\texttt{ALG})})}{L+1} \le \mathrm{ST}(\texttt{ALG}) \le \mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\texttt{ALG})}),\ a.s.$$

# Bandit Framework for Speculative Decoding: BANDITSPEC

**Proposition (Property of BANDITSPEC)**

*For any arm selection algorithm* ALG *that selects an arm according to the history,*

$$\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})} \overset{d}{=} \mathrm{pt}_{\tau_c}.$$

*Furthermore, the stopping time* $\mathrm{ST}(\mathtt{ALG})$ *can be bounded as*

$$\frac{\mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})})}{L+1} \leq \mathrm{ST}(\mathtt{ALG}) \leq \mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})}),\ a.s.$$

- Quality of the token sequence is not compromised.

# Bandit Framework for Speculative Decoding: BanditSpec

**Proposition (Property of BanditSpec)**

*For any arm selection algorithm* ALG *that selects an arm according to the history,*

$$\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})} \overset{d}{=} \mathrm{pt}_{\tau_c}.$$

*Furthermore, the stopping time* $\mathrm{ST}(\mathtt{ALG})$ *can be bounded as*

$$\frac{\mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})})}{L+1} \leq \mathrm{ST}(\mathtt{ALG}) \leq \mathrm{len}(\mathrm{pt}_{\mathrm{ST}(\mathtt{ALG})}),\ a.s.$$

- Quality of the token sequence is not compromised.
- The stopping time and the length of the generated tokens sequence are equivalent up to a constant factor.

# Bandit Framework for Speculative Decoding: BanditSpec

- **Objective:** Devise an arm selection rule ALG to minimize the *stopping time regret*

$$\begin{aligned}\mathrm{Reg}(\texttt{ALG}, \mathrm{pt}, \nu) := \mathbb{E}\big[\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big] \\ - \mathbb{E}\big[\mathrm{ST}(\texttt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big].\end{aligned}$$

# Bandit Framework for Speculative Decoding: BanditSpec

- **Objective:** Devise an arm selection rule ALG to minimize the *stopping time regret*

$$\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu) := \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big] - \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big].$$

  - $\mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of speculative decoding rounds for ALG to stop.

# Bandit Framework for Speculative Decoding: BanditSpec

- **Objective:** Devise an arm selection rule ALG to minimize the *stopping time regret*

$$\mathrm{Reg}(\texttt{ALG}, \mathrm{pt}, \nu) := \mathbb{E}\big[\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big] - \mathbb{E}\big[\mathrm{ST}(\texttt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big].$$

  - $\mathbb{E}\big[\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of speculative decoding rounds for ALG to stop.
  - $\mathbb{E}\big[\mathrm{ST}(\texttt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of rounds to stop if the oracle best hyperparameter $S_{i^*(\mathrm{pt},\nu)}$ is adopted.

# Bandit Framework for Speculative Decoding: BanditSpec

- **Objective:** Devise an arm selection rule ALG to minimize the *stopping time regret*

$$\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu) := \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big] - \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big].$$

  - $\mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of speculative decoding rounds for ALG to stop.
  - $\mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of rounds to stop if the oracle best hyperparameter $S_{i^*(\mathrm{pt},\nu)}$ is adopted.
  - $\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu)$ measures the wasted/additional rounds.

# Bandit Framework for Speculative Decoding: BANDITSPEC

- **Objective:** Devise an arm selection rule ALG to minimize the *stopping time regret*

$$\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu) := \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big] - \mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big].$$

  - $\mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of speculative decoding rounds for ALG to stop.
  - $\mathbb{E}\big[\mathrm{ST}(\mathtt{ALG}_{i^*(\mathrm{pt},\nu)}, \mathrm{pt}, \nu) \mid \mathrm{pt}, \nu\big]$: the expected number of rounds to stop if the oracle best hyperparameter $S_{i^*(\mathrm{pt},\nu)}$ is adopted.
  - $\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu)$ measures the wasted/additional rounds.
- **Desired result:**

$$\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu) = o\Big(\mathbb{E}\big[\mathrm{len}(\mathrm{pt}_{\tau_c})\big]\Big) \text{ or } o\Big(\mathbb{E}\big[\tau_c\big]\Big).$$

  - The additional rounds are negligible for ALG, compared to the oracle best hyperparameter $S_{i^*(\mathrm{pt},\nu)}$.

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits
  - For instance, even if we know there will be $1$ accepted token at each round, the stopping time is

$$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = \mathrm{len}(\mathrm{pt}_{\tau_c})$$

which depends on the target model $P$.

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits
  - For instance, even if we know there will be $1$ accepted token at each round, the stopping time is

$$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = \mathrm{len}(\mathrm{pt}_{\tau_{\mathrm{c}}})$$

    which depends on the target model $P$.
- The stopping time depends on the generated tokens. Mathematically,

$$\psi_t : \mathcal{X}^* \to \{\mathrm{stop}, \mathrm{continue}\}, \quad \psi_t(\mathrm{pt}_t) = \mathrm{stop} \text{ or } \mathrm{continue}$$

  is a function that maps the generated tokens to the stopping decision, which the agent does not know in advance.

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits
  - For instance, even if we know there will be $1$ accepted token at each round, the stopping time is

    $$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = \mathrm{len}(\mathrm{pt}_{\tau_c})$$

    which depends on the target model $P$.
- The stopping time depends on the generated tokens. Mathematically,

  $$\psi_t : \mathcal{X}^* \to \{\mathrm{stop}, \mathrm{continue}\}, \quad \psi_t(\mathrm{pt}_t) = \mathrm{stop} \text{ or } \mathrm{continue}$$

  is a function that maps the generated tokens to the stopping decision, which the agent does not know in advance.
- The tokens are generated autoregressively, thus, the accepted tokens are not i.i.d.

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits
  - For instance, even if we know there will be $1$ accepted token at each round, the stopping time is

$$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = \mathrm{len}(\mathrm{pt}_{\tau_c})$$

which depends on the target model $P$.

- The stopping time depends on the generated tokens. Mathematically,

$$\psi_t : \mathcal{X}^* \to \{\mathrm{stop}, \mathrm{continue}\}, \quad \psi_t(\mathrm{pt}_t) = \mathrm{stop} \text{ or } \mathrm{continue}$$

is a function that maps the generated tokens to the stopping decision, which the agent does not know in advance.

- The tokens are generated autoregressively, thus, the accepted tokens are not i.i.d.

Some simplifications are required to make the problem tractable.

# Challenges

- The there is another source of uncertainty, the stopping time, in addition to the arm rewards in standard multi-armed bandits
  - For instance, even if we know there will be $1$ accepted token at each round, the stopping time is

$$\mathrm{ST}(\texttt{ALG}, \mathrm{pt}, \nu) = \mathrm{len}(\mathrm{pt}_{\tau_c})$$

which depends on the target model $P$. $\Rightarrow$ Anytime-version algorithms

- The stopping time depends on the generated tokens. Mathematically,

$$\psi_t : \mathcal{X}^* \to \{\mathrm{stop}, \mathrm{continue}\}, \quad \psi_t(\mathrm{pt}_t) = \mathrm{stop} \text{ or } \mathrm{continue}$$

is a function that maps the generated tokens to the stopping decision, which the agent does not know in advance. $\Rightarrow$ Martingale arguments

- The tokens are generated autoregressively, thus, the accepted tokens are not i.i.d. $\Rightarrow$ Stochastic/Adversarial models

Some simplifications are required to make the problem tractable.

# Arm Selection: Modeling Tokens Stochastically

**Assumption (Stationary Mean Values)**

*There exist $K$ values $\{\mu_i\}_{i\in[K]}\subset[1,L+1]$, such that the expected number of the accepted tokens satisfies*

$$\mathbb{E}[Y_{I_t,t}\,|\,\mathcal{H}_{t-1},I_t]=\mu_{I_t}.$$

# Arm Selection: Modeling Tokens Stochastically

**Assumption (Stationary Mean Values)**

*There exist $K$ values $\{\mu_i\}_{i\in[K]} \subset [1, L+1]$, such that the expected number of the accepted tokens satisfies*

$$\mathbb{E}[Y_{I_t,t} \,|\, \mathcal{H}_{t-1}, I_t] = \mu_{I_t}.$$

- UCBSpec: A UCB-type algorithm, whose confidence radius is derived from the self-normalized bound, which holds for all $t \in \mathbb{N}$ (Abbasi-yadkori et al., 2011).
- We design the algorithm as an anytime version, meaning it does not require the time horrizon as an input.

# Arm Selection: Modeling Tokens Stochastically

**Algorithm 7** UCBSPEC

**Inputs:** number of hyperparameter specifications $K$, history $\mathcal{H}_t = \big((I_s, X_{I_s,s})\big)_{s=1}^{t}$, confidence parameter $\delta$.

**Procedures:**

1: **if** $t \leq K - 1$ **then return** $I_{t+1} = t + 1$.

2: Compute the lengths $Y_{I_s,s} = \mathrm{len}(X_{I_s,s})$ for all $s \in [t]$.

3: Set the statistics $\{\hat{\mu}_{i,t}\}_{i\in[K]}, \{\mathrm{UCB}_{i,t} = \hat{\mu}_{i,t} + \mathrm{cr}_{i,t}\}_{i\in[K]}$, where

$$n_{i,t} = \sum_{s=1}^{t} \mathbb{1}\{I_s = i\},\ \hat{\mu}_{i,t} = \frac{\sum_{s=1}^{t} Y_{i,s}\mathbb{1}\{I_s = i\}}{n_{i,t}},$$

$$\mathrm{cr}_{i,t} = \frac{L}{2}\sqrt{\frac{1+n_{i,t}}{n_{i,t}^2}\left(1 + 2\log\frac{Kt^2(1+n_{i,t})^{\frac{1}{2}}}{\delta}\right)},$$

4: **return** index $I_{t+1} = \mathrm{argmax}_{i\in[K]}\, \mathrm{UCB}_{i,t}$.

# Arm Selection: Modeling Tokens Stochastically

**Theorem (Upper Bound for Stationary Case)**

*Under Assumptions 2 and finite length assumption, given any prompt* $\mathrm{pt} \in \mathcal{X}^*$ *and bandit configuration* $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$, *the expected stopping time regret of* $\texttt{ALG} = \textsc{UCBSpec}$ *is upper bounded as*

$$\mathrm{Reg}(\texttt{ALG}, \mathrm{pt}, \nu) = O\Big(\mathrm{H}(\mathrm{pt}, \nu) \cdot L^2 \cdot \log \mathbb{E}[\mathrm{len}(\mathrm{pt}_{\tau_c})]\Big).$$

Here, $\Delta_i := \mu_{i^*} - \mu_i$ and the *hardness parameter*

$$\mathrm{H}(\mathrm{pt}, \nu) := \sum_{i \neq i^*} \frac{1}{\mu_{i^*} \Delta_i}.$$

# Arm Selection: Modeling Tokens Stochastically

**Theorem (Upper Bound for Stationary Case)**

*Under Assumptions 2 and finite length assumption, given any prompt* $\mathrm{pt} \in \mathcal{X}^*$ *and bandit configuration* $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$, *the expected stopping time regret of* $\texttt{ALG} = \mathrm{UCBSPEC}$ *is upper bounded as*

$$\mathrm{Reg}(\texttt{ALG}, \mathrm{pt}, \nu) = O\Big(\mathrm{H}(\mathrm{pt}, \nu) \cdot L^2 \cdot \log \mathbb{E}[\mathrm{len}(\mathrm{pt}_{\tau_c})]\Big).$$

Here, $\Delta_i := \mu_{i^*} - \mu_i$ and the *hardness parameter*

$$\mathrm{H}(\mathrm{pt}, \nu) := \sum_{i \neq i^*} \frac{1}{\mu_{i^*} \Delta_i}.$$

# Theoretical Results: Lower Bound

*Truncated geometric distribution* (TGD) on $[1, L+1]$:

$$P_S(x) = \begin{cases} p^{x-1}(1-p), & x = 1, 2, \ldots, L, \\ p^L, & x = L+1. \end{cases} \quad (1)$$

which was considered in the seminal work on speculative decoding (Leviathan et al., 2023).

**Proposition (Tightness Result (Informal))**

*Let* $\mathcal{S}_{\mathrm{TGD}} = \{S : P_S \text{ satisfies } (1)\}$. *Let* $\{S_i\}_{i=1}^{K} \subset \mathcal{S}_{\mathrm{TGD}}$ *and* $S_i$ *satisfies* (1) *with* $p_i$*, then under the greedy decoding strategy,*

$$\liminf_{m\to\infty} \frac{\mathrm{Reg}(\texttt{ALG}, \mathrm{pt}^m, \nu)}{\log(\mathrm{len}(\mathrm{pt}^m_{\tau_c}))} \geq \mathrm{H}(\mathrm{pt}^m, \nu) \cdot \frac{p_{i^*}(1-p_{i^*}^L)}{(1-p_{i^*})}.$$

*If* $p_{i^*} \in \left(2^{-1/L}, 1\right)$*, the bounds match up absolute constants and* $L$*.*

# Arm Selection: Modeling Tokens Adversarially

**Assumption (Adversarial Mean Values)**

*Let the number of accepted tokens generated by hyperparameter $S_i$ at time step $t$ be $y_{i,t} = \mathrm{len}(X_{i,t})$. We assume $\{y_{i,t}\}_{i \in [K], t \in \mathbb{N}}$ is fixed by the environment before the algorithm starts.*

- This is analogous to the oblivious adversarial case in bandits.
- EXP3SPEC: An EXP3-type algorithm (anytime version).

# Arm Selection: Modeling Tokens Adversarially

**Assumption (Adversarial Mean Values)**

*Let the number of accepted tokens generated by hyperparameter $S_i$ at time step $t$ be $y_{i,t} = \mathrm{len}(X_{i,t})$. We assume $\{y_{i,t}\}_{i \in [K], t \in \mathbb{N}}$ is fixed by the environment before the algorithm starts.*

- This is analogous to the oblivious adversarial case in bandits.
- $\textsc{EXP3Spec}$: An EXP3-type algorithm (anytime version).

**Theorem (Upper Bound for Adversarial Case, Informal)**

*Under some assumptions and using the greedy decoding strategy, given any prompt $\mathrm{pt} \in \mathcal{X}^*$ and bandit configuration $\nu = (P, \mathcal{S} = \{S_i\}_{i \in [K]}, L)$, the expected stopping time regret of $\mathtt{ALG} = \textsc{EXP3Spec}$ is*

$$\mathrm{Reg}(\mathtt{ALG}, \mathrm{pt}, \nu) = O\left(L \sqrt{\min_{i \in [K]} \mathrm{ST}(\mathtt{ALG}_i) K \log K}\right).$$

# Arm Selection: Modeling Tokens Adversarially

**Algorithm 8** EXP3SPEC

**Inputs:** Num. of hyperparams. $K$, history $\mathcal{H}_t = \big((I_s, X_{I_s,s})\big)_{s=1}^{t}$.

**Procedures:**

1: Compute the lengths $Y_{I_s,s} = \mathrm{len}(X_{I_s,s})$ for all $s \in [t]$.

2: Set the statistics $\widehat{Z}_{i,t} = \mathbb{1}\{i = I_t\} \cdot \frac{L+1-Y_{i,t}}{L \cdot p_{t,i}}$ for all $i \in [K]$.

3: Set learning rate $\eta_t = \sqrt{\log K/(t \cdot K)}$.

4: Set probability vector $p_t \in \mathbf{\Delta}_{[K]}$ with for all $i \in [K]$

$$p_{t,i} = \frac{\exp\Big(-\eta_t \sum_{s=1}^{t-1} \widehat{Z}_{i,s}\Big)}{\sum_{j=1}^{K} \exp\Big(-\eta_t \sum_{s=1}^{t-1} \widehat{Z}_{j,s}\Big)}.$$

5: **return** hyperparameter index $I_{t+1} \sim p_t$.

# Experimental Setup

- **Two Target Models:** LLaMA3-8B-Instruct and Qwen2-7B-Instruct.
- **Four Benchmarks**: Spec Bench (Xia et al., 2024), Alpaca (Taori et al., 2023), Code Editor (Guo et al., 2024) and Debug Bench (Tian et al., 2024).
- **Two Metrics:**
  - *Mean Accepted Tokens (MAT)*: important measurement, mainly focus on the design of the speculative decoding algorithm.
  - *Throughput (Tokens/s)*: closely related to user experience, and it takes all relevant factors into account, e.g., the hardware, the coding, the algorithm, etc.

# Experimental Setup

- **Two Target Models:** LLaMA3-8B-Instruct and Qwen2-7B-Instruct.
- **Four Benchmarks**: Spec Bench (Xia et al., 2024), Alpaca (Taori et al., 2023), Code Editor (Guo et al., 2024) and Debug Bench (Tian et al., 2024).
- **Two Metrics:**
  - *Mean Accepted Tokens (MAT)*: important measurement, mainly focus on the design of the speculative decoding algorithm.
  - *Throughput (Tokens/s)*: closely related to user experience, and it takes all relevant factors into account, e.g., the hardware, the coding, the algorithm, etc.
- **Two Settings:**
  - Batchsize $= 1$ and four drafting methods: PLD (Saxena, 2023), Rest (He et al., 2024), Suffix Tree (Oliaro et al., 2024; Hu et al., 2024) and Eagle-2 (Li et al., 2024a).
  - Simulated real-world scenario: heterogeneous requests at one time, with different hyperparameters (i.e., speculation length $\gamma \in [4]$) as arms.

# Experimental Results: Setting One

| Methods | Spec Bench | | Alpaca | | Code Editor | | Debug Bench | |
|---|---|---|---|---|---|---|---|---|
| | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) |
| ***LLaMA3-8B-Instruct*** | | | | | | | | |
| Vanilla | 1.00 | 35.73 | 1.00 | 35.92 | 1.00 | 36.32 | 1.00 | 36.89 |
| PLD | 1.46 | 43.96 | 1.53 | 53.06 | 2.13 | 82.61 | 1.67 | 82.76 |
| Rest | 1.29 | 40.67 | 1.48 | 52.40 | 1.33 | 51.32 | 1.29 | 48.49 |
| Suffix Tree | 1.83 | 55.10 | 1.71 | 64.02 | 2.30 | 90.21 | 2.13 | 77.56 |
| Eagle-2 | 3.94 | 98.15 | 4.04 | 110.00 | 4.79 | 128.76 | **4.78** | 119.12 |
| EXP3SPEC | 3.65 | 102.10 | 4.23 | 120.38 | 4.36 | 137.29 | 4.50 | 132.25 |
| UCBSPEC | **3.98** | **105.72** | **4.35** | **125.78** | **4.83** | **138.27** | 4.60 | **135.34** |
| ***Qwen2-7B-Instruct*** | | | | | | | | |
| Vanilla | 1.00 | 38.71 | 1.00 | 39.32 | 1.00 | 39.30 | 1.00 | 39.57 |
| PLD | 1.55 | 52.44 | 1.42 | 58.41 | 1.89 | 64.56 | 2.15 | 70.49 |
| Rest | 1.31 | 46.42 | 1.47 | 59.01 | 1.31 | 53.79 | 1.22 | 50.51 |
| Suffix Tree | 1.96 | 68.42 | 1.46 | 62.60 | 2.18 | 85.75 | 2.49 | 101.47 |
| Eagle-2 | 3.64 | 97.82 | 3.61 | 104.43 | 4.88 | 138.58 | 4.79 | 126.01 |
| EXP3SPEC | 3.76 | 107.36 | 3.83 | 113.90 | 4.90 | 160.41 | 4.86 | **151.73** |
| UCBSPEC | **4.13** | **112.33** | **3.93** | **114.20** | **4.92** | **161.35** | **5.10** | 151.37 |

# Experimental Results: Setting One

| Methods | Spec Bench | | Alpaca | | Code Editor | | Debug Bench | |
|---|---|---|---|---|---|---|---|---|
| | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) |
| ***LLaMA3-8B-Instruct*** | | | | | | | | |
| Vanilla | 1.00 | 35.73 | 1.00 | 35.92 | 1.00 | 36.32 | 1.00 | 36.89 |
| PLD | 1.46 | 43.96 | 1.53 | 53.06 | 2.13 | 82.61 | 1.67 | 82.76 |
| Rest | 1.29 | 40.67 | 1.48 | 52.40 | 1.33 | 51.32 | 1.29 | 48.49 |
| Suffix Tree | 1.83 | 55.10 | 1.71 | 64.02 | 2.30 | 90.21 | 2.13 | 77.56 |
| Eagle-2 | <u>3.94</u> | 98.15 | 4.04 | 110.00 | <u>4.79</u> | 128.76 | **4.78** | 119.12 |
| EXP3SPEC | 3.65 | <u>102.10</u> | <u>4.23</u> | <u>120.38</u> | 4.36 | <u>137.29</u> | 4.50 | <u>132.25</u> |
| UCBSPEC | **3.98** | **105.72** | **4.35** | **125.78** | **4.83** | **138.27** | <u>4.60</u> | **135.34** |

- The best throuput performance is always achieved by the proposed algorithms.

# Experimental Results: Setting One

| Methods | Spec Bench | | Alpaca | | Code Editor | | Debug Bench | |
|---|---|---|---|---|---|---|---|---|
| | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) |
| ***LLaMA3-8B-Instruct*** | | | | | | | | |
| Vanilla | 1.00 | 35.73 | 1.00 | 35.92 | 1.00 | 36.32 | 1.00 | 36.89 |
| PLD | 1.46 | 43.96 | 1.53 | 53.06 | 2.13 | 82.61 | 1.67 | 82.76 |
| Rest | 1.29 | 40.67 | 1.48 | 52.40 | 1.33 | 51.32 | 1.29 | 48.49 |
| Suffix Tree | 1.83 | 55.10 | 1.71 | 64.02 | 2.30 | 90.21 | 2.13 | 77.56 |
| Eagle-2 | 3.94 | 98.15 | 4.04 | 110.00 | 4.79 | 128.76 | **4.78** | 119.12 |
| EXP3SPEC | 3.65 | 102.10 | 4.23 | 120.38 | 4.36 | 137.29 | 4.50 | 132.25 |
| UCBSPEC | **3.98** | **105.72** | **4.35** | **125.78** | **4.83** | **138.27** | 4.60 | **135.34** |

- The best throuput performance is always achieved by the proposed algorithms.
- The proposed methods automatically adapt to different prompts, as they are better than the best fixed method.

# Experimental Results: Setting One

| Methods | Spec Bench | | Alpaca | | Code Editor | | Debug Bench | |
|---|---|---|---|---|---|---|---|---|
| | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) | MAT(↑) | Tokens/s(↑) |
| ***LLaMA3-8B-Instruct*** | | | | | | | | |
| Vanilla | 1.00 | 35.73 | 1.00 | 35.92 | 1.00 | 36.32 | 1.00 | 36.89 |
| PLD | 1.46 | 43.96 | 1.53 | 53.06 | 2.13 | 82.61 | 1.67 | 82.76 |
| Rest | 1.29 | 40.67 | 1.48 | 52.40 | 1.33 | 51.32 | 1.29 | 48.49 |
| Suffix Tree | 1.83 | 55.10 | 1.71 | 64.02 | 2.30 | 90.21 | 2.13 | 77.56 |
| Eagle-2 | <u>3.94</u> | 98.15 | 4.04 | 110.00 | <u>4.79</u> | 128.76 | **4.78** | 119.12 |
| EXP3SPEC | 3.65 | <u>102.10</u> | <u>4.23</u> | <u>120.38</u> | 4.36 | <u>137.29</u> | 4.50 | <u>132.25</u> |
| UCBSPEC | **3.98** | **105.72** | **4.35** | **125.78** | **4.83** | **138.27** | <u>4.60</u> | **135.34** |

- The best throuput performance is always achieved by the proposed algorithms.
- The proposed methods automatically adapt to different prompts, as they are better than the best fixed method.
- As the empirical performance of UCBSPEC is better than EXP3SPEC, it implies that real-life scenario tends to be benign and may be more aligned with the stationary mean assumption.

# Experimental Results: Setting Two

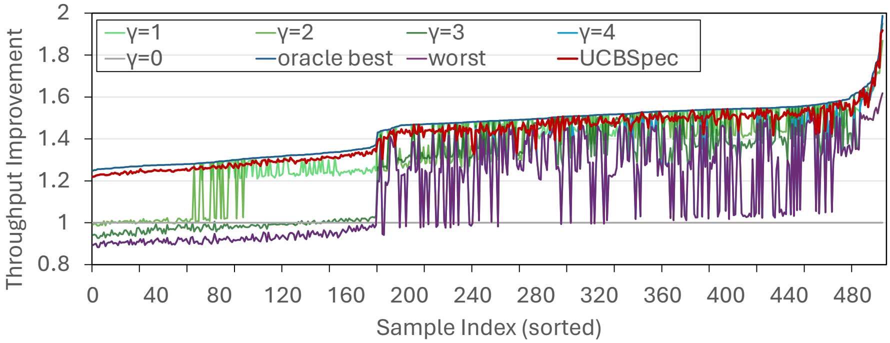



Figure: Target model: LLaMA3-8B-Instruct, Draft Model: Eagle-1

# Experimental Results: Setting Two

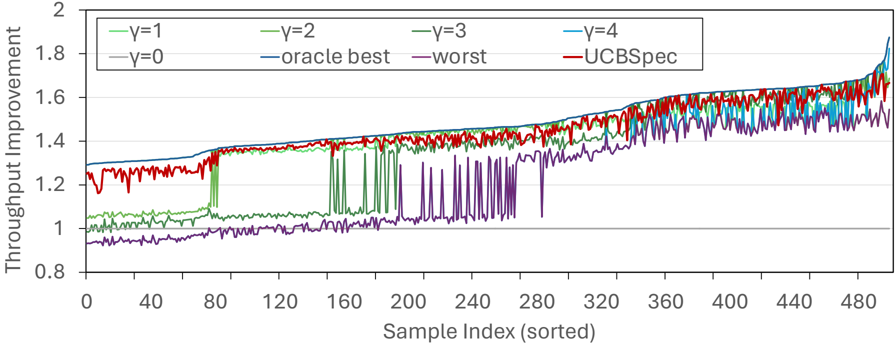



Figure: Target model: Qwen2-7B-Instruct, Draft Model: Eagle-1

# Discussions

- The stationary mean assumption (Assumption 2)

$$\mathbb{E}[Y_{I_t,t} \,|\, \mathcal{H}_{t-1}, I_t] = \mu_{I_t}.$$

  is strictly weaker than the usual i.i.d. assumption in bandits.

- Many variants of UCB (e.g., KL-UCB) algorithms that one can try.
- However, the implementation time of these arm selection algorithm should also be considered, as we aim at reduce the wall-time of the decoding process.

# Conclusions and Future Directions

**Summary:**

- Introduced BANDITSPEC, a bandit-based *training-free* framework for *adaptive* speculative decoding.
- Developed two arm algorithms (UCBSPEC and EXP3SPEC) with theoretical regret guarantees.
- Demonstrated significant empirical improvements in decoding throughput.

**Future work:**

- Robust bandits and Non-stationary bandits.
- Contextual Bandits.

# Reference I

Abbasi-yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, pages 2312–2320.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. (2023). Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Guo, J., Li, Z., Liu, X., Ma, K., Zheng, T., Yu, Z., Pan, D., LI, Y., Liu, R., Wang, Y., Guo, S., Qu, X., Yue, X., Zhang, G., Chen, W., and Fu, J. (2024). Codeeditorbench: Evaluating code editing capability of large language models.

He, Z., Zhong, Z., Cai, T., Lee, J., and He, D. (2024). REST: Retrieval-based speculative decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1582–1595. Association for Computational Linguistics.

Hu, Y., Wang, K., Zhang, J., Li, C., and Chen, H. (2024). Sam decoding: Speculative decoding via suffix automaton. *arXiv preprint arXiv:2411.10666*.

Leviathan, Y., Kalman, M., and Matias, Y. (2023). Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Li, Y., Wei, F., Zhang, C., and Zhang, H. (2024a). EAGLE-2: Faster inference of language models with dynamic draft trees. In *Empirical Methods in Natural Language Processing*.

Li, Y., Wei, F., Zhang, C., and Zhang, H. (2024b). Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Oliaro, G., Jia, Z., Campos, D., and Qiao, A. (2024). Suffixdecoding: A model-free approach to speeding up large language model inference. *arXiv preprint arXiv:2411.04975*.

Saxena, A. (2023). Prompt lookup decoding.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Tian, R., Ye, Y., Qin, Y., Cong, X., Lin, Y., Liu, Z., and Sun, M. (2024). Debugbench: Evaluating debugging capability of large language models.

# Reference II

Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W., and Sui, Z. (2024). Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 7655–7671, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yin, M., Chen, M., Huang, K., and Wang, M. (2024). A theoretical perspective for speculative decoding algorithm. *arXiv preprint arXiv:2411.00841*.

# Thanks for Listening!