A Further Experiment Results

In our experiments, we set the confidence scaling parameter α of LinUCB to 1, and the variance of noise in TS to 0.3.

The suboptimality comparisons of models in Low Non-stationary environments and High Non-stationary environments with $b = 0.018$ (Figure 3) are shown in Figure 4 and 5. We observe that the transformer not only outperforms the expert algorithms but also maintains a consistently low suboptimality rate.

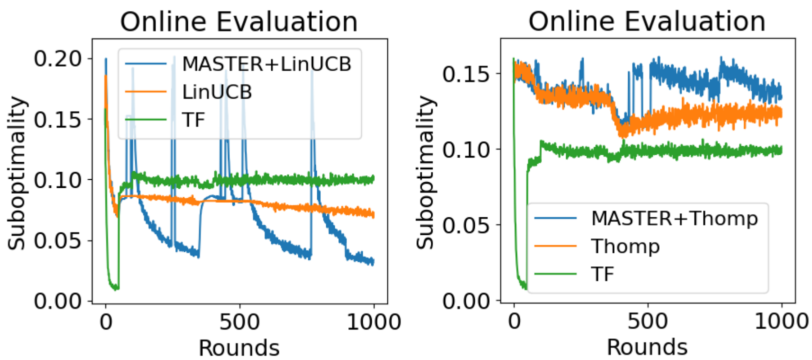

Figure 4: Suboptimality Comparisons of LinUCB, Thompson Sampling (TS), MASTER+LinUCB/TS, and transformer (TF) in Low Non-stationary environments. Linear bandit with $d = 32$, $A = 10$. Shading indicates the standard deviation of the regret estimates.

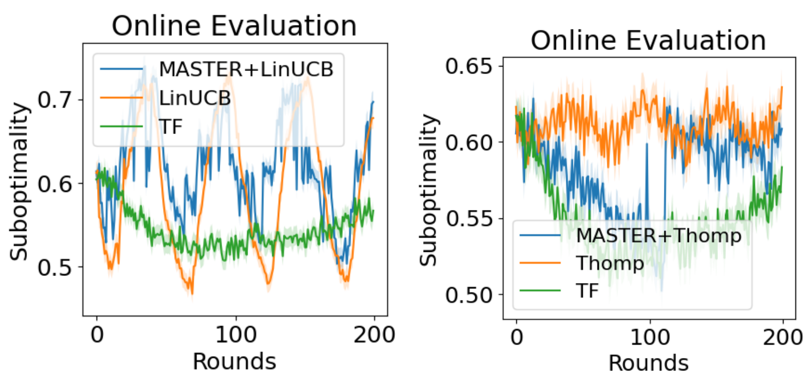

Figure 5: Suboptimality Comparisons of LinUCB, Thompson Sampling (TS), MASTER+LinUCB/TS, and transformer (TF) in High Non-stationary environments ($b = 0.018$). Linear bandit with $d = 32$, $A = 10$. Shading indicates the standard deviation of the regret estimates.

Figures 6 and 7 show the results for High Non-stationary environments with $b = 0.025$. We observe that when the level of non-stationarity is too high, the transformer loses its superiority compared to the other models. This suggests that the model generalizes well to $b = 0.018$, but does not sufficiently generalize to $b = 0.025$, possibly due to limited representation in the training data.

B MASTER Algorithm

We start by introducing MALG (Multi-scale ALGorithm) (Wei and Luo, 2021). MALG builds on a base reinforcement learning algorithm, ALG, by running multiple instances at different scales. Given an integer n and a non-increasing function $\rho : [T] \to \mathbb{R}$, MALG operates over a time horizon of length 2^n.

At each time step τ, given an integer m, if τ is a multiple of 2^m, MALG schedules a new instance of length 2^m with probability $\rho(2^n)/\rho(2^m)$. These are called order-m instances. The start and end

Figure 6: Suboptimality and Cumulative Regret Comparisons of LinUCB, MASTER+LinUCB, and transformer (TF) in High Non-stationary environments ($b = 0.025$). Linear bandit with $d = 32$, $A = 10$. Shading indicates the standard deviation of the regret estimates.

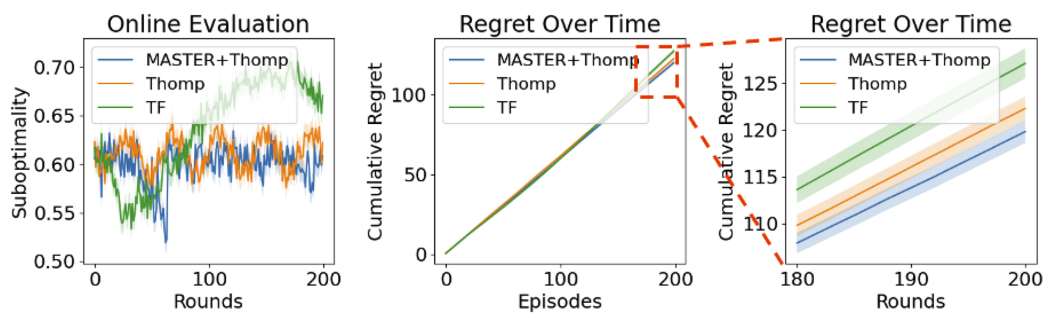

Figure 7: Suboptimality and Cumulative Regret Comparisons of Thompson sampling (TS), MASTER+TS, and transformer (TF) in High Non-stationary environments ($b = 0.025$). Linear bandit with $d = 32$, $A = 10$. Shading indicates the standard deviation of the regret estimates.

times of each instance are denoted by $alg.s$ and $alg.e$, respectively. Among all scheduled instances, only the one with the lowest order remains active at any given time, producing an auxiliary value $\widetilde{r}_t$.

After each step, MALG updates the active instance based on observed rewards R_t from the environment. All reinforcement learning algorithms in this framework produce a specific auxiliary value; in the LinUCB setting, $\widetilde{r}_t$ corresponds to a computed score that combines a reward estimate with an uncertainty term.

Algorithm 1: MALG (Multi-scale ALG)(Wei and Luo, 2021)

Input: n, $\rho(\cdot)$

for $\tau = 0, \ldots, 2^n - 1$ **do**
  for $m = n, n-1, \ldots, 0$ **do**
    if *τ is a multiple of 2^m* **then**
      With probability $\rho(2^n)/\rho(2^m)$, schedule a new instance *alg* of ALG at scales 2^m;
  Run the active instance *alg* to output $\widetilde{r}_\tau$, select an action, and update with feedback.

MASTER (MALG with Stationarity Tests and Restarts) further enhances this process. It repeatedly tests for stability and performance using two key checks. The first test ensures that the average reward within any completed instance does not significantly exceed a threshold tied to $\widetilde{r}_t$. The second test verifies that the difference between the auxiliary output $\widetilde{r}_t$ and observed rewards r_t remains bounded on average. If either test fails, MASTER resets the process, ensuring that the algorithm remains robust under changing conditions.

Algorithm 2: MALG with Stationarity TEsts and Restarts (MASTER)(Wei and Luo, 2021)

Input: $\widehat{\rho}(\cdot)$ where $\widehat{\rho}(t) = 6(\log_2 T + 1)\log(T)\rho(t)$ (T: block length)

1 **Initialize** $t \leftarrow 1$ **for** $n = 0, 1, \dots$ **do**

2   Set $t_n \leftarrow t$ and initialize an MALG (Algorithm 2) for the block $[t_n, t_n + 2^n - 1]$;

3   **while** $t < t_n + 2^n$ **do**

4   Run MALG to obtain prediction $\widetilde{r}_t$, select action a_t, and receive reward R_t;

5   Update MALG with feedback, and set $U_t = \min_{\tau \in [t_n, t]} \widetilde{r}_\tau$;

6   Perform Test 1 and Test 2 (see below);

7   Increment $t \leftarrow t + 1$;

8   **if** ***either test returns fail*** **then**

9   restart from Line 2;

10 **Test 1**: If $t = alg.e$ for some order-m alg and $\frac{1}{2^m}\sum_{\tau=alg.s}^{alg.e} R_\tau \geqslant U_t + 9\widehat{\rho}(2^m)$, return fail.

11 **Test 2**: If $\frac{1}{t-t_n+1}\sum_{t_n}^{t}(\widetilde{r}_\tau - r_\tau) \geqslant 3\widehat{\rho}(t - t_n + 1)$, return fail.

C Proofs in Section 3

C.1 Intermediate Statement

We provide the following intermediate statement to show Theorem 1.

Theorem 6. *Let Assumption 1 and 2 hold. Then for any small $\varepsilon > 0$, there exists an algorithm $\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}$ introduced by a transformer with*

$$\begin{gathered} D \leqslant D_0 + 10, \quad M = \max\{M_0, 2\}, \quad L = L_0, \\ D' = \mathcal{O}(\max\{D'_0, T\log_2 T/\varepsilon\}), \quad \|\!|\widehat{\boldsymbol{\theta}}|\!\| = \widetilde{\mathcal{O}}(C_0 + \sqrt{T\log_2 T/\varepsilon}) \end{gathered} \tag{12}$$

satisfying

$$|R_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}}(T) - R_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T)| = \widetilde{\mathcal{O}}\left(T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}, \overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_\Theta T/\delta)}{n}}\right) + \varepsilon AT\right). \tag{13}$$

C.2 Proof of Theorem 1

As established in Wei and Luo (2021), for any reinforcement learning algorithm alg_E, the combination of MASTER and alg results in a stabilized version: $\mathsf{alg}_{\mathcal{E}}$. The dynamic regret of $\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}$ can then be expressed as

$$\mathfrak{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}}(T) \leqslant \underbrace{\left|\mathfrak{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}}(T) - \mathfrak{R}_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T)\right|}_{=:\mathfrak{A}} + \underbrace{\left|\mathfrak{R}_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T)\right|}_{=:\mathfrak{B}}. \tag{14}$$

$\mathfrak{A}$ can be obtained from Theorem 6, and $\mathfrak{B}$ is bounded as follows:

$$\mathfrak{B} = \widetilde{\mathcal{O}}\left(\min\left\{\left(c_1 + \frac{c_2}{c_1}\right)\sqrt{JT},\ \left(c_1^{2/3} + c_2 c_1^{-4/3}\right)\Delta^{1/3}T^{2/3} + \left(c_1 + \frac{c_2}{c_1}\right)\sqrt{T}\right\}\right),$$

where $C(t) = c_1 t^{1/2} + c_2$ with $c_1, c_2 > 0$ (Wei and Luo, 2021). Setting $c_1 = 1$ and $c_2 \ll 1$, we obtain:

$$\mathfrak{B} = \widetilde{\mathcal{O}}\left(\min\left\{\sqrt{JT},\ \Delta^{1/3}T^{2/3} + \sqrt{T}\right\}\right).$$

□

C.3 Proof of Theorem 2

It suffices to prove that

$$T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}, \overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_\Theta T/\delta)}{n}} + \sqrt{\varepsilon}\right) + \varepsilon AT = \widetilde{\mathcal{O}}(\sqrt{T}).$$

According to Theorem 5, we have

$$\frac{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(a_t|D_{t-1},s_t)}{\overline{\mathsf{alg}}_E(a_t|D_{t-1},s_t)} \leqslant e^{\varepsilon}.$$

Thus, the cumulative distribution ratio over T rounds satisfies

$$\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E} = e^{\varepsilon T}.$$

Since we assume $\varepsilon \leqslant T^{-3}$, it follows that

$$\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E} = \mathcal{O}(1).$$

Consequently, we have

$$T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\cdot\sqrt{\varepsilon} + \varepsilon AT = \mathcal{O}(\sqrt{T}).$$

Next, following Vaart and Wellner (2023), there exist universal constants C_1 and C_2 such that

$$\mathcal{N}_\Theta = \mathcal{N}_\Theta((NT)^{-2}) \leqslant (2N^2T^2)^{C_1 V(C_2(NT)^{-2},\Theta)}$$

where $V(\varepsilon,\Theta)$ denotes the fat-shattering dimension of Θ. Let $V(\Theta)$ be the VC-dimension of Θ; then, for any $\varepsilon > 0$, it holds that $V(\varepsilon,\Theta) \leqslant V(\Theta)$ (Vaart and Wellner, 2023). Since Θ is a finite parameter class, there exists a constant C_3 such that $V(\varepsilon,\Theta) \leqslant V(\Theta) \leqslant C_3$ for all $\varepsilon > 0$. Therefore, under the assumption that $N \geqslant CT^3\log T$, we obtain:

$$\begin{aligned}
T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\sqrt{\frac{\log(\mathcal{N}_\Theta T/\delta)}{N}} &\leqslant T^2\sqrt{\frac{\log(\mathcal{N}_\Theta T/\delta)}{N}} \\
&\leqslant T^2\sqrt{\frac{(1+2C_1C_3)\log T + 2C_1C_3\log N + \log(2^{C_1C_3}/\delta)}{N}} \\
&\leqslant T^2\sqrt{\frac{\widetilde{\mathcal{O}}(\log T)}{\mathcal{O}(T^3\log T)}} \\
&= \widetilde{\mathcal{O}}(\sqrt{T})
\end{aligned}$$

which completes the proof. □

C.4 Proof of Theorem 5

Consider $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$. As shown in Section D, a noncontinuous transformer (Definition 7) with

$$D = ((d+5)(n+1)+5), \quad M = 2, \quad L > 0, \quad D' = T\log_2 T/\varepsilon, \quad |||\boldsymbol{\theta}||| = \widetilde{\mathcal{O}}\left(\sqrt{T\log_2 T/\varepsilon}\right),$$

can approximate the MASTER algorithm. Treating any reinforcement learning algorithm that uses MASTER as the expert algorithm $\mathsf{alg}_{\mathcal{E}}$ in Theorem 5, the result follows directly from Assumption 2. □

C.5 Proof of Theorem 6

By Theorem 5, we have

$$\begin{aligned}
&\log\frac{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(a_{t,k}|D_{t-1},s_t)}{\overline{\mathsf{alg}}_{\mathcal{E}}(a_{t,k}|D_{t-1},s_t)} < \varepsilon \\
&\Rightarrow \mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(a_{t,k}|D_{t-1},s_t) < e^{\varepsilon}\cdot\overline{\mathsf{alg}}_{\mathcal{E}}(a_{t,k}|D_{t-1},s_t) \\
&\Rightarrow \left|\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(a_{t,k}|D_{t-1},s_t) - \overline{\mathsf{alg}}_{\mathcal{E}}(a_{t,k}|D_{t-1},s_t)\right| < (e^{\varepsilon}-1),
\end{aligned}$$

where the last inequality uses the fact that $\mathsf{alg}(\cdot) \leqslant 1$. With a slight abuse of notation, let $r_{t,k}$ denote the reward of the k-th action at round t. Since $|r_{t,k}| \leqslant 1$ almost surely for all $t \in [T]$, we obtain the policy imitation error between transformers and MASTER as follows:

$$\text{Policy Imitation Error} = \left|\sum_{t=1}^{T}\sum_{k=1}^{A}\left(\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(a_{t,k}|D_{t-1},s_t) - \overline{\mathsf{alg}}_{\mathcal{E}}(a_{t,k}|D_{t-1},s_t)\right) r_{t,k}\right|$$

$$
\begin{aligned}
&\leqslant \sum_{t=1}^{T}\sum_{k=1}^{A}\left|\mathsf{alg}_{\hat{\boldsymbol{\theta}}}(a_{t,k}|D_{t-1},s_t) - \overline{\mathsf{alg}}_{\mathcal{E}}(a_{t,k}|D_{t-1},s_t)\right| \cdot |r_{t,k}| \\
&\leqslant \sum_{t=1}^{T}\sum_{k=1}^{A}(e^{\varepsilon}-1) \\
&= (e^{\varepsilon}-1)AT.
\end{aligned}
$$

Finally, since $\varepsilon > 0$ is small, we use the approximation

$$
e^{\varepsilon} = 1 + \varepsilon + \mathcal{O}(\varepsilon^2),
$$

and then the policy imitation error becomes εAT.

Next, we leverage the result from Lin et al. (2024):

$$
\left|\mathfrak{R}_{\mathsf{alg}_{\hat{\theta}}}(T) - \mathfrak{R}_{\overline{\mathsf{alg}}_E}(T)\right| = \mathcal{O}\left(T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}T/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right)\right) \tag{15}
$$

where

$$
\log \mathbb{E}_{\overline{D}_T \sim \mathbb{P}_{\mathsf{alg}_0,\mathsf{alg}_E}}\left[\frac{\overline{\mathsf{alg}}_E(\bar{a}_t|D_{t-1},s_t)}{\mathsf{alg}_{\hat{\boldsymbol{\theta}}}(\bar{a}_t|D_{t-1},s_t)}\right] \leqslant \varepsilon_{\text{real}}
$$

for all $t \in [T]$. We extend this result to non-stationary settings to derive a bound for $|\mathfrak{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}}}(T) - \mathfrak{R}_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T)|$.

Suppose there are $n_0 + 1$ orders of instances, and alg_E runs for T rounds. Let $k_i \geqslant 0$ represent the number of rounds for order-i instances, so that $k_0 + k_1 + \cdots + k_{n_0} = T$. Define T_i as:

$$
T_i := \{t_1^{(i)}, ..., t_{k_i}^{(i)}\}, \quad |T_i| = k_i, \quad i \in \{0, ..., n_0\}
$$

which is the set of rounds during which order-i instances are active. If $k_i = 0$, order-i instances are inactive. If $k_i > 0$, one or more order-i instances are active. Using Lemma 7, we know that running one instance during T_i yields a higher regret bound (as in (15)) than running zero or multiple instances in T_i. Furthermore, Theorem 6 guarantees that the regret for a reinforcement learning algorithm at T_i is bounded by $\varepsilon A k_i$. Thus, we have:

$$
\left|\mathfrak{R}_{\mathsf{alg}_{\hat{\theta}}}(T_i) - \mathfrak{R}_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T_i)\right| = \mathcal{O}\left(k_i^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}k_i/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right) + \varepsilon A k_i\right).
$$

Summing over all T_i, we get:

$$
\begin{aligned}
\left|\mathfrak{R}_{\mathsf{alg}_{\hat{\theta}}}(T) - \mathfrak{R}_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T)\right| &\leqslant \sum_{i=0}^{n_0}\left|\mathfrak{R}_{\overline{\mathsf{alg}}_{\hat{\theta}}}(T_i) - \mathfrak{R}_{\mathsf{alg}_{\mathcal{E}}}(T_i)\right| \\
&\leqslant \sum_{i=0}^{n_0}\left|R_{\mathsf{alg}_{\hat{\theta}}}(T_i) - R_{\overline{\mathsf{alg}}_{\mathcal{E}}}(T_i)\right| \\
&= \mathcal{O}\left(\sum_{i=0}^{n_0} k_i^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}k_i/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right) + \varepsilon A k_i\right) \\
&\leqslant \mathcal{O}\left(\sum_{i=0}^{n_0} k_i^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}T/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right) + \varepsilon A k_i\right) \\
&\leqslant \mathcal{O}\left(\left(\sum_{i=0}^{n_0} k_i\right)^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}T/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right) + \sum_{i=0}^{n_0}\varepsilon A k_i\right) \\
&= \mathcal{O}\left(T^2\sqrt{\mathcal{R}_{\mathsf{alg}_{\hat{\boldsymbol{\theta}}},\overline{\mathsf{alg}}_E}}\left(\sqrt{\frac{\log(\mathcal{N}_{\Theta}T/\delta)}{N}} + \sqrt{\varepsilon_{\text{real}}}\right) + \varepsilon AT\right),
\end{aligned}
$$

where the final inequality follows from the Cauchy–Schwarz inequality.

By Assumption 2, we have

$$\frac{\overline{\mathsf{alg}}_E(\overline{a}_t|D_{t-1},s_t)}{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(\overline{a}_t|D_{t-1},s_t)} < e^{\varepsilon}.$$

Therefore,

$$\log \mathbb{E}_{\overline{D}_T\sim\mathbb{P}_{\mathsf{alg}_0,\mathsf{alg}_E}}\left[\frac{\overline{\mathsf{alg}}_E(\overline{a}_t|D_{t-1},s_t)}{\mathsf{alg}_{\widehat{\boldsymbol{\theta}}}(\overline{a}_t|D_{t-1},s_t)}\right] \leqslant \varepsilon_{\text{real}} \leqslant \varepsilon.$$

Combining this with the result of Theorem 6 completes the proof. □

C.5.1 An auxiliary lemma

Lemma 7 (Regret bound for multiple instances)**.** *Let $a,b,c,N,C_1,C_2>0$ with $a+b=c$. The following bound holds:*

$$a^2\left(\sqrt{\frac{\log(C_1a)}{N}}+C_2\right)+b^2\left(\sqrt{\frac{\log(C_1b)}{N}}+C_2\right)\leqslant c^2\left(\sqrt{\frac{\log(C_1c)}{N}}+C_2\right). \tag{16}$$

Proof of Lemma 7. Let $0<u<1$ be such that $a=uc$ and $b=(1-u)c$. Substituting into (16), we obtain:

$$u^2\left(\sqrt{\frac{\log(C_1uc)}{N}}+C_2\right)+(1-u)^2\left(\sqrt{\frac{\log(C_1(1-u)c)}{N}}+C_2\right)\leqslant c^2\left(\sqrt{\frac{\log(C_1c)}{N}}+C_2\right).$$

Denoting $G:=\log(C_1c)/N$, we rewrite the inequality as:

$$u^2\left(\sqrt{\log u+G}+C_2\right)+(1-u)^2\left(\sqrt{\log(1-u)+G}+C_2\right)\leqslant\sqrt{G}+C_2. \tag{17}$$

Since $0<u<1$, we have $\log u<0$ and $\log(1-u)<0$. Substituting these into (17), we obtain:

$$u^2+(1-u)^2\leqslant 1,$$

which follows directly from $0<u<1$, as $u^2+(1-u)^2=1-2u(1-u)<1$. □

D Proofs in Section 4

D.1 Structural Approximation

To match the order-n instance length as described in Appendix B, we consider a single input matrix $\mathbf{H}=[\mathbf{h}_1,...,\mathbf{h}_{2^n}]\in\mathbb{R}^{d\times 2^n}$, where $\{\mathbf{h}_i\}_{i=1}^{2^n}\subset\mathbb{R}^d$. This matrix is then extended to $\mathcal{H}\in\mathbb{R}^{D\times 2^n}$ defined as $\mathcal{H}:=\left[\mathbf{H}^{(0)};\cdots;\mathbf{H}^{(n)};\mathbf{H}^*\right]\in\mathbb{R}^{D\times 2^n}$ (Figure 8) where $D:=((d+5)(n+1)+5)$, and $\mathbf{H}^*$ contains auxiliary entries for approximation purposes. Here, $\mathbf{H}^{(i)}:=[\mathbf{h}_1^{(i)}\cdots\mathbf{h}_{2^n}^{(i)}]$ and $\mathbf{h}_t^{(i)}=[\mathbf{h}_t;2^i,0,0,0,0]$where the four auxiliary entries capture the order information. Details of each entry are given in (21) and (22). Once $\mathcal{H}$ is defined, the transformer produces the output $\overline{\mathcal{H}}\in\mathbb{R}^{D\times 2^n}$. It's worth noting that even if the input length T is not exactly 2^n, the transformer can still perform operations as in MASTER and generate a length-T output by selecting the smallest 2^n greater than T and running the algorithm for T rounds.

We begin by defining functions σ_1 and σ_2 to replicate the MALG operation, where σ_1 stochastically schedules instances for each order, and σ_2 selects the instance with the lowest order to remain active in each round.

Definition 4 (σ_1)**.** *Given a non-increasing function $\rho:[2^n]\to\mathbb{R}$, we define $\sigma_1^{\rho}:\mathbb{R}^{D\times 2^n}\to\mathbb{R}^{D\times 2^n}$ as*

$$\sigma_1^{\rho}(\mathcal{H})=\left[\sigma_1(\mathbf{H}^{(0)},0,n,\rho);\cdots;\sigma_1(\mathbf{H}^{(n)},n,n,\rho)\right],$$

where for each $i\in\{0,...,n\}$

$$\sigma_1(\mathbf{H}^{(i)},i,n,\rho)=\left\{\mathbf{h}_j^{(i)}\cdot\mathrm{B}\left(\frac{\rho(2^n)}{\rho(2^i)}\right)\middle|\ j\in\{t,t+1,...,t+2^i-1\},\ t\ mod\ 2^i=1\right\}\in\mathbb{R}^{D\times 2^n}.$$

Here, $\mathrm{B}(k)$ *is a Bernoulli random variable with parameter k, meaning $B(k)=1$ with probability $k\leqslant 1$ and $B(k)=0$ otherwise.*

Using σ_1, multiple instances can be scheduled simultaneously (represented as the blueish blocks in Figure 8). Since only one instance can run in the environment at any given time, we define σ_2 to select the instance with the lowest order at each time t.

We denote $\mathfrak{h}_t := \left[\mathbf{h}_t^{(0)}; \cdots; \mathbf{h}_t^{(n)}\right] \in \mathbb{R}^D$ for each $t \in [2^n]$, so that $\mathcal{H} = [\mathfrak{h}_1, ..., \mathfrak{h}_{2^n}]$. Then, σ_2 is defined as follows.

Definition 5 (σ_2)**.** *We define* $\sigma_2 : \mathbb{R}^{D \times 2^n} \to \mathbb{R}^{D \times 2^n}$ *by*

$$\sigma_2(\mathcal{H}) = \left[\mathfrak{h}'_1, \ldots, \mathfrak{h}'_{2^n}\right],$$
$$\mathfrak{h}'_t = \left[\mathbf{0}; ...; \mathbf{0}; \mathbf{h}_t^{(k)}; \mathbf{0}; ...; \mathbf{0}; *; *; *; *; *\right] \ \left(\mathbf{h}_t^{(0)}, ..., \mathbf{h}_t^{(k-1)} = \mathbf{0},\ \mathbf{h}_t^{(k)} \neq \mathbf{0},\ 0 \leqslant k \leqslant n\right)$$

to reproduce the uniqueness of an instance scheduled at every moment in MALG. Here, $*$*s denote the last five entries of* $\mathfrak{h}_t$.

After passing through σ_2, only one instance remains active at each time t, and this active instance has the lowest order among all scheduled instances. See the reddish blocks in Figure 8 for illustration.

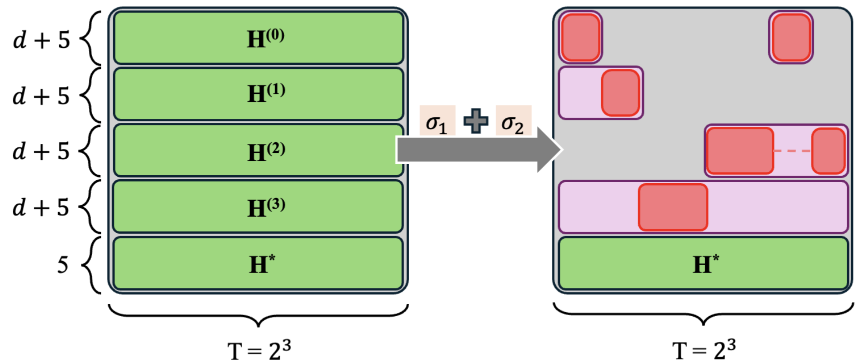

Figure 8: Illustration of $\mathcal{H}$ when $n = 3$. Purplish blocks represent instances scheduled by σ_1, while reddish blocks represent the active instances selected by σ_2. Reddish blocks connected by a dashed line are concatenated.

To approximate the stationary tests (**Test 1** and **Test 2**) in Algorithm 2, we define the following:

Definition 6 (`TEST`)**.** `TEST` *represents the stationary tests in MASTER, involving two functions* `test1` *and* `test2` *that map from* $\mathbb{R}^D \times \mathbb{R}^{D \times 2^n}$ *to* $\mathbb{R}$. *Given* $\widehat{\rho}$ *defined in Algorithm 2 and an active instance in* $\mathfrak{h}_t$ *of order* m*, where* $\mathfrak{h}_t$ *contains* $m, t, U_t, \widetilde{r}_t$*, and* r_t*, we define:*

$$\texttt{test1}_{\widehat{\rho}}(\mathfrak{h}_t, \mathcal{H}) = \begin{cases} 1, & \text{if some order}-m \ \mathsf{alg} \text{ ends and } \frac{1}{2^m} \sum\limits_{\tau = t - 2^m + 1}^{t} r_\tau < U_t + 9\widehat{\rho}(2^m) \\ 0, & \text{otherwise} \end{cases},$$

$$\texttt{test2}_{\widehat{\rho}}(\mathfrak{h}_t, \mathcal{H}) = \begin{cases} 1, & \text{if } \frac{1}{t - t_n + 1} \sum\limits_{\tau=1}^{t} (\widetilde{r}_\tau - r_\tau) < 3\widehat{\rho}(t - t_n + 1) \\ 0, & \text{otherwise} \end{cases}.$$

`TEST` *is then defined as:*

$$\texttt{TEST}_{\widehat{\rho}}(\mathfrak{h}_t, \mathcal{H}) := \texttt{test1}(\mathfrak{h}_t, \mathcal{H}) \cdot \texttt{test2}(\mathfrak{h}_t, \mathcal{H}) = \begin{cases} 1, & \text{if } \texttt{test1}(\mathfrak{h}_t) = \texttt{test2}(\mathfrak{h}_t) = 1 \\ 0, & \text{otherwise} \end{cases}.$$

Finally, we define the noncontinuous transformer. Since the stationary tests are included in the noncontinuous transformer and it stops processing the remaining sequence if any test fails, we define a transformer module that processes each element (not the whole sequence), performs the tests, and then concatenates the results.

Definition 7 (Noncontinuous Transformer)**.** *For $t \in [2^n]$, We first denote $\mathfrak{h}_t^*$ as*

$$\mathfrak{h}_t^* = \left(\mathrm{MLP}_{\theta_{mlp}^{(L)}} \circ \mathrm{Attn}_{\theta_{\mathrm{attn}}^{(L)}}\right) \circ \cdots \circ \left(\mathrm{MLP}_{\theta_{\mathrm{mlp}}^{(1)}} \circ \mathrm{Attn}_{\theta_{\mathrm{attn}}^{(1)}}\right)(\mathfrak{h}_t, \mathcal{H}) \in \mathbb{R}^D.$$

where parameters $\theta_{attn}^{(\ell)} = \{\mathbf{V}_m^{(\ell)}, \mathbf{Q}_m^{(\ell)}, \mathbf{K}_m^{(\ell)}\}_{m\in[2^n]} \subset \mathbb{R}^{D\times D}$ and $\theta_{\mathrm{mlp}}^{(\ell)} = \{(\mathbf{W}_1^{(\ell)}, \mathbf{W}_2^{(\ell)})\}_{m\in[2^n]} \subset \mathbb{R}^{D'\times D} \times \mathbb{R}^{D\times D'}$. After processing $\mathfrak{h}_t$, the model interacts with the environment, observes the reward, and inserts it into the sequence: $\mathfrak{h}_t^* \to \widetilde{\mathfrak{h}}_t$ ((21) and (22)). By concatenating $\widetilde{\mathfrak{h}}_{t\in[2^n]}$, we define the L-layer noncontinuous transformer $\mathcal{TF}_\theta : \mathbb{R}^{D\times 2^n} \to \mathbb{R}^{D\times 2^n}$ as

$$\mathcal{TF}_\theta = \mathcal{TF}_\theta^{(2^n)} \tag{18}$$

$$\mathcal{TF}_\theta^{(T)}(\mathcal{H}) = \widetilde{\mathcal{H}}^{(T)} \in \mathbb{R}^{D\times 2^n} \ \ (T \in [2^n]) \tag{19}$$

where $\widetilde{\mathcal{H}}^{(T)} := \left[\widetilde{\mathfrak{h}}_1, ..., \widetilde{\mathfrak{h}}_T, \mathbf{T}_T\,\mathfrak{h}_{T+1}, ..., \mathbf{T}_T\,\mathfrak{h}_{2^n}\right] \in \mathbb{R}^{D\times 2^n}$. Here, $\mathbf{T}_T \in \mathbb{R}^{D\times D}$ denotes the test matrix at time T:

$$\mathbf{T}_T := \mathrm{diag}(\underbrace{1, ..., 1}_{d+1}, \overbrace{\mathtt{TEST}, \mathtt{TEST}, \mathtt{TEST}, \mathtt{TEST}}^{\times(n+1)}, 1, 1, \mathtt{TEST}, \mathtt{TEST}, \mathtt{TEST}) \tag{20}$$

where $\mathtt{TEST} = \mathtt{TEST}(\overline{\mathfrak{h}}_T, \mathcal{H}^{(T)})$. Here, order information $\sum_{\tau=t-2^i+1}^{t-1} r_\tau, \sum_{k=0}^{i-1} \widetilde{r}_t^{(k)}, \widetilde{r}_t^{(i)}$, historical data $\sum_{\tau=1}^{t-1} r_\tau, \sum_{\tau=1}^{t-1} \widetilde{r}_\tau, U_{t-1}$, and `rand` in (21) and 22 are reset when $\mathtt{TEST}= 0$. $\widetilde{r}_t^{(i)}$ denotes the auxiliary value generated at order i.

Interaction with the environment To illustrate the transformer's interaction with the environment, we consider the following:

$$\mathbf{h}_t^{(i)} = \begin{bmatrix} \mathbf{x}_t^{(i)} \\ 2^i \\ \sum_{\tau=t-2^i+1}^{t-1} r_\tau \\ 0 \\ \sum_{k=0}^{i-1} \widetilde{r}_t^{(k)} \\ \widetilde{r}_t^{(i)} \end{bmatrix} \xrightarrow[\text{Interact with the environment}]{\text{Processed by the module}} \begin{bmatrix} \overline{\mathbf{x}}_t^{(i)} \\ 2^i \\ \sum_{\tau=t-2^i+1}^{t} r_\tau \\ \mathtt{rand} \\ \sum_{k=0}^{i-1} \widetilde{r}_t^{(k)} \\ \widetilde{r}_t^{(i)} \end{bmatrix} = \overline{\mathbf{h}}_t^{(i)}, \tag{21}$$

$$\mathfrak{h}_t = \begin{bmatrix} \mathbf{h}_t^{(0)} \\ \vdots \\ \mathbf{h}_t^{(n)} \\ 2^n \\ t \\ \sum_{\tau=1}^{t-1} r_\tau \\ \sum_{\tau=1}^{t-1} \widetilde{r}_\tau \\ U_{t-1} \end{bmatrix} \xrightarrow[\text{Interact with the environment}]{\text{Processed by the module}} \begin{bmatrix} \overline{\mathbf{h}}_t^{(0)} \\ \vdots \\ \overline{\mathbf{h}}_t^{(n)} \\ 2^n \\ t \\ \sum_{\tau=1}^{t} r_\tau \\ \sum_{\tau=1}^{t} \widetilde{r}_\tau \\ U_t \end{bmatrix} = \overline{\mathfrak{h}}_t. \tag{22}$$

Here, $\sum_{\tau=1}^{0} r_\tau = 0$ and $r_\tau = 0$ for $\tau \leqslant 0$. The `rand` entry is a random number generated by the transformer. The vectors $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$ are used for in-context learning, and $\overline{\mathbf{x}}_t^{(i)}$ represents their processed forms. For instance, these vectors can encode action choices in bandit problems (Lin et al., 2024).

Each vector entry captures cumulative information or constants relevant to the model's operations up to time t. After passing the sequence through the MLP+Attention module, the module schedules certain orders at each time t, setting all other instances to $\mathbf{0}$. The σ_2 operation ensures that only one instance is active at a time by selecting the instance with the lowest order and setting all others to $\mathbf{0}$. If the active instance is order $q(\leqslant n)$, then $\widetilde{r}_t^{(q)}$ is assigned as $\widetilde{r}_t$ at time t, and $U_t = \min\{U_{t-1}, \widetilde{r}_t^{(q)}\}$. Finally, the module interacts with the environment, observes reward R_t, and updates the tensor accordingly.

Notation with regard to the input sequence To avoid potential confusion, the notations defined above are summarized below:

- $\mathbf{h}_t \in \mathbb{R}^d$: The t-th element of the input sequence.
- $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_{2^n}] \in \mathbb{R}^{D \times 2^n}$: The input sequence as a matrix.
- $\mathcal{H} = [\mathbf{H}^{(0)}; \cdots; \mathbf{H}^{(n)}; \mathbf{H}^*] = [\mathfrak{h}_1 \cdots \mathfrak{h}_{2^n}] \in \mathbb{R}^{D \times 2^n}$: The extended input sequence, where $\mathbf{H}^{(i)}$ denotes the order-i sequence.
- $\mathfrak{h}_t = \left[\mathbf{h}_t^{(0)}; \cdots; \mathbf{h}_t^{(n)}; \mathbf{h}_t^*\right] \in \mathbb{R}^D$: The t-th vector of the extended sequence.
- $\mathfrak{h}'_t = \left[\mathbf{0}; \cdots; \mathbf{0}; \mathbf{h}_t^{(k)}; \mathbf{0}; \cdots; \mathbf{0}; \mathbf{h}_t^*\right] \in \mathbb{R}^D$: The t-th vector of the extended input after being processed by σ_2. Here, k represents the lowest nonzero order.
- $\mathfrak{h}_t^*$: The result of processing $\mathfrak{h}_t$ by the noncontinuous transformer module.
- $\overline{\mathfrak{h}}_t$: The transformer output after interacting with the environment.

Restart The restart mechanism (line 9 of Algorithm 2) is built into the revised transformer. If no tests fail, the matrices $\mathbf{T}_T$ ($T \in [2^n]$) remain identity matrices, and the sequence is processed as usual. If a test fails at time T, the matrices $\mathbf{T}_T$ ($T \in [2^n]$) modify the states $\overline{\mathfrak{h}}_{T+1}, \ldots, \overline{\mathfrak{h}}_{2^n}$. They keep the first $D-7$ elements—containing $\mathbf{x}_T$, T, 2^n, and 2^i—and set the remaining seven elements to zero. This ensures that when a test fails, all earlier information is erased while the key components are retained.

Rollout The noncontinuous transformer's architecture is more complex than the classic version, so we outline its input-output flow.

Starting with an input sequence $\mathbf{H} \in \mathbb{R}^{D \times 2^n}$, we extend it to $\mathcal{H} = [\mathbf{H}^{(0)}; \cdots; \mathbf{H}^{(n)}; \mathbf{H}^*] \in \mathbb{R}^{D \times 2^n}$, where submatrices $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(j)}$ are only different in several entries to record order information.

$\mathcal{H}$ is passed through σ_1 and σ_2: σ_1 schedules instances for each order, and σ_2 activates only the instance with the lowest order at any given time t. As a result, at each time t, only one instance is active, and the active instance may come from a different order at each time step.

Next, each element in the sequence is processed by the traditional MLP+Attention module. The module uses this information to interact with the environment, collects the reward, and inserts it into the sequence: $\mathfrak{h}_t^* \to \overline{\mathfrak{h}}_t$ ((21) and (22)). The updated sequence $\overline{\mathfrak{h}}_t$ is evaluated by `TEST`, which performs stationary tests. If the tests pass, the block remains unchanged; otherwise, the remaining block entries are set to zero, keeping only essential information. This effectively ends the current 2^n-length block and starts a new 2^n-length block (line 2, Algorithm 2). On restart, variables like R_t and f_t are reinserted at their appropriate positions, following the same procedure. After processing the entire sequence, the final tensor $\overline{\mathcal{H}}$ is produced. By gathering the rewards from all orders at each t (only one reward is nonzero at a given time), the resulting sequence $\overline{\mathbf{H}}$ serves as the output of the noncontinuous transformer, containing cumulative rewards up to time 2^n.

D.2 Approximation of `WS` (Proposition 3)

We divide the proof into three parts: approximating random number generation, σ_1, and σ_2. Throughout this section, we denote the ReLU activation function by $\sigma(\cdot)$.

D.2.1 Approximating random number generation

We start by approximating $\mathbb{1}_{>0}[\cdot]$. According to the definition of the ReLU function, we have

$$\sigma(kx) - \sigma(kx-1) = \begin{cases} 0, & x \leqslant 0 \\ kx, & 0 < x \leqslant \frac{1}{k} \\ 1, & x > \frac{1}{k} \end{cases} .$$

When $k \to \infty$, it approximates $\mathbb{1}[x] = \begin{cases} 0, & x \leqslant 0 \\ 1, & x > 0 \end{cases}$.

We denote $z_k := \mathbf{x}_{k,0}$ and $P_z(z)$ being the cumulative distribution function (CDF) of $\{z_1, ..., z_{2^n}\}$. Following the proof in Hataya and Imaizumi (2024), we construct a 2-head attention layer to approximate the CDF.

From the approximation of $\mathbb{1}[\cdot]$, $P_z(t) = 1/2^n \sum_{k=1}^{2^n} \mathbb{1}[\mathbf{x}_{k,0} \leqslant t]$ can be approximated by sum of ReLU functions as

$$\widehat{P}_z(t) \approx \frac{1}{2^n} \sum_{k=1}^{2^n} \{\sigma(k(t-x)) - \sigma(k(t-x)-1)\}$$

where k is sufficiently large. Suppose the vector in formula 21 and 22 (left) is one of the input $\mathbf{h}_t$. By selecting $\{\mathbf{Q}_{1,2}, \mathbf{K}_{1,2}, \mathbf{V}_{1,2}\}$ such that

$$\mathbf{Q}_1\mathbf{h}_i = \begin{bmatrix} k\mathbf{x}_{i,0} \\ -1 \\ 0 \\ \mathbf{0} \end{bmatrix}, \ \mathbf{Q}_2\mathbf{h}_i = \begin{bmatrix} k\mathbf{x}_{i,0} \\ -1 \\ -1 \\ \mathbf{0} \end{bmatrix}, \ \mathbf{K}_1\mathbf{h}_i = \mathbf{K}_2\mathbf{h}_i = \begin{bmatrix} 1 \\ k\mathbf{x}_{i,0} \\ 1 \\ \mathbf{0} \end{bmatrix},$$

$$\mathbf{V}_1\mathbf{h}_i = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \ \mathbf{V}_2\mathbf{h}_i = \begin{bmatrix} -1 \\ \mathbf{0} \end{bmatrix},$$

we have

$$\sum_{m=1}^{2} \sigma(\langle \mathbf{Q}_m\mathbf{h}_j, \mathbf{K}_m\mathbf{h}_i \rangle) \mathbf{V}_m\mathbf{h}_i = \sigma(k(\mathbf{x}_{j,0} - \mathbf{x}_{i,0})) - \sigma(k(\mathbf{x}_{j,0} - \mathbf{x}_{i,0}) - 1).$$

Therefore, the output is

$$\widetilde{\mathbf{h}}_i = \mathbf{h}_i + \frac{1}{2^n} \sum_{j=1}^{2^n} \sum_{m=1}^{2} \sigma(\langle \mathbf{Q}_m\mathbf{h}_i, \mathbf{K}_m\mathbf{h}_i \rangle) \mathbf{V}_m\mathbf{h}_i = \begin{bmatrix} \texttt{rand} \\ \vdots \end{bmatrix}$$

where $\texttt{rand} = \widehat{P}_z(\mathbf{x}_{i,0})$ can be regarded as a random variable sampled from $\mathcal{U}(0,1)$. □

D.2.2 Approximating σ_1

Step 1. Generating Bernoulli masks. Define Bernoulli Masks as $\mathtt{B}_{i,t} = \mathbb{1}[\texttt{rand}_{i,t} \leqslant \rho(2^n)/\rho(2^i)]$. We consider a random number $\texttt{rand}_{i,t} \in \mathbb{R}$ at order i and time t, and $\texttt{rand}_{i,t}, \rho(\cdot) \in [0,1]$. Following the approach in Step 1, we begin by considering the input $\mathbf{h}_t^{(i)}$. This input includes $\texttt{rand}_{i,t}, 2^i, 2^n$, and we also assume that $\rho(2^i)$ and $\rho(2^n)$ are added to $\mathbf{h}_t^{(i)}$ through $\rho(\cdot)$. In this case, a two-layer MLP can implement Bernoulli Masks.

First, consider the function $f(x,y) = x \cdot y \in \mathbb{R}$. For $x, y \in [0,1]$, dividing the domain $[0,1] \times [0,1]$ into $\lfloor 1/\epsilon \rfloor$ segments allows us to express it as:

$$\mathbf{W}\begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{b} = \sum_{k,l=1}^{n} \alpha_{k,l}x + \beta_{k,l}y + \gamma_{k,l}.$$

This form can approximate f using an MLP. Based on this strategy, the two-layer MLP is constructed as follows:

$$\begin{aligned} z_1 &= \sigma(\mathbf{W}_1\mathbf{h}_t^{(i)} + \mathbf{b}_1) \\ &= \sigma\left(\begin{bmatrix} k_2\left(\rho(2^n) - (\alpha_{1,1}\texttt{rand}_{i,t} + \beta_{1,1}\rho(2^i) + \gamma_{1,1})\right) \\ \vdots \\ k_2\left(\rho(2^n) - (\alpha_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor}\texttt{rand}_{i,t} + \beta_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor}\rho(2^i) + \gamma_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor})\right) \\ k_2\left(\rho(2^n) - (\alpha_{1,1}\texttt{rand}_{i,t} + \beta_{1,1}\rho(2^i) + \gamma_{1,1})\right) - 1 \\ \vdots \\ k_2\left(\rho(2^n) - (\alpha_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor}\texttt{rand}_{i,t} + \beta_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor}\rho(2^i) + \gamma_{\lfloor \frac{1}{\sqrt{\epsilon}} \rfloor, \lfloor \frac{1}{\sqrt{\epsilon}} \rfloor})\right) - 1 \end{bmatrix}\right), \end{aligned}$$

where $\mathbf{W}_1 \in \mathbb{R}^{2\lfloor 1/\varepsilon \rfloor \times D}$.The output is then processed by another layer:

$$\begin{aligned}
\mathbf{W}_2 z_1 + \mathbf{b}_2 &= \sigma\Big(k_2\big(\rho(2^n) - \sum_{k,l=1}^{n} (\alpha_{k,l}\mathtt{rand}_{i,t} + \beta_{k,l}\rho(2^i) + \gamma_{k,l})\big)\Big) \\
&\quad - \sigma\Big(k_2\big(\rho(2^n) - \sum_{k,l=1}^{n} (\alpha_{k,l}\mathtt{rand}_{i,t} + \beta_{k,l}\rho(2^i) + \gamma_{k,l})\big) - 1\Big) \\
&\approx \sigma\Big(k_2(\rho(2^n) - \mathtt{rand} \cdot \rho(2^i))\Big) - \sigma\Big(k_2(\rho(2^n) - \mathtt{rand} \cdot \rho(2^i)) - 1\Big) \\
&\longrightarrow \begin{cases} 1, & \mathtt{rand}_{i,t} \leqslant \frac{\rho(2^n)}{\rho(2^i)}, \\ 0, & \mathtt{rand}_{i,t} > \frac{\rho(2^n)}{\rho(2^i)}. \end{cases} \quad (k_2 \to \infty), \quad \mathbf{W}_2 \in \mathbb{R}^{1 \times 2\lfloor 1/\varepsilon \rfloor}.
\end{aligned}$$

The hidden dimensions of $\mathbf{W}_1, \mathbf{W}_1$ are $\mathcal{O}(1/\varepsilon)$. If the input is expanded from $\mathbf{h}_t^{(i)}$ to $\mathbf{H}^{(i)}$, the hidden dimensions increase to $\mathcal{O}(2^n/\varepsilon)$. Further expand further to $\mathcal{H}$, the hidden dimensions of $\mathbf{W}_1, \mathbf{W}_1$ become $\mathcal{O}(n2^n/\varepsilon) = \mathcal{O}(T \log_2 T/\varepsilon)$. The operator norms $\|\mathbf{W}_1\|_{\mathrm{op}}, \|\mathbf{W}_1\|_{\mathrm{op}}$ scale as $\mathcal{O}(\sqrt{T \log_2 T/\varepsilon})$. Finally, by taking $k_2 \to \infty$, the error approaches zero.

Step 2: Generating block masks. By setting $\mathtt{rand}_{i,t} = \cdots = \mathtt{rand}_{i,t+2^i-1}$ $(t \equiv 1\,(mod\ 2))$, we have $\mathtt{B}_{i,t} = \cdots = \mathtt{B}_{i,t+2^i-1}$ $(t \equiv 1\,(mod\ 2))$. Since the output of Bernoulli masks can be expressed as $\mathcal{H} := [\mathfrak{h}_1 \cdots \mathfrak{h}_{2^n}] \in \mathbb{R}^{D \times 2^n}$, where for $t \in [2^n]$:

$$\mathfrak{h}_t = \begin{bmatrix} \mathbf{h}_t^{(0)} \\ \vdots \\ \mathbf{h}_t^{(n)} \\ \star \end{bmatrix}, \quad \mathbf{h}_t^{(0)} = \begin{bmatrix} \mathbf{x}_t^{(0)} \\ 0 \\ 0 \\ \mathtt{B}_{0,t} \\ 0 \\ f_t^{(0)} \end{bmatrix} \quad \mathbf{h}_t^{(1)} = \begin{bmatrix} \mathbf{x}_t^{(1)} \\ 0 \\ 0 \\ \mathtt{B}_{1,t} \\ f_t^{(0)} \\ f_t^{(1)} \end{bmatrix}, \quad \ldots, \quad \mathbf{h}_t^{(n)} = \begin{bmatrix} \mathbf{x}_t^{(n)} \\ 0 \\ 0 \\ \mathtt{B}_{n,t} \\ \sum_{p=0}^{n-1} f_t^{(p)} \\ f_t^{(n)} \end{bmatrix},$$

we choose $\{\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m\}_{m \in \{0,\ldots,n\}}$ such that for tokens $\mathfrak{h}_t$,

$$\mathbf{Q}_m \mathfrak{h}_t = \begin{bmatrix} t \\ 1 \\ \mathtt{B}_{m,t} \\ -1 \end{bmatrix}, \quad \mathbf{K}_m \mathfrak{h}_k = \begin{bmatrix} -1 \\ k \\ 1 \\ 0.5 \end{bmatrix}, \quad \mathbf{V}_m \mathfrak{h}_k = \Big[\mathbf{0}; \ldots, \mathbf{0}; \mathbf{h}_k^{(m)}; \mathbf{0}; \ldots, \mathbf{0}\Big]$$

In this way, with $k < t$ we have

$$\sum_{m=0}^{n} \mathbb{1}_{>0}\left[\langle \mathbf{Q}_m \mathfrak{h}_t, \mathbf{K}_m \mathfrak{h}_k \rangle\right] \mathbf{V}_m \mathfrak{h}_k = \mathbf{0},$$

and with $k = t$ we have

$$\sum_{m=0}^{n} r\left(\langle \mathbf{Q}_m \mathfrak{h}_t, \mathbf{K}_m \mathfrak{h}_t \rangle\right) \mathbf{V} \mathfrak{h}_t = \sum_{m=0}^{n} \mathbb{1}_{>0}\left[\mathtt{B}_{m,t} - 0.5\right] \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{h}_k^{(m)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathtt{B}_{0,t}\mathbf{h}_t^{(0)} \\ \vdots \\ \mathtt{B}_{n,t}\mathbf{h}_t^{(n)} \end{bmatrix}.$$

Finally, by referring to the approximation of $\mathbb{1}_{>0}[\cdot]$ (D.2.1), we note that this indicator function can be approximated using two ReLU functions. Consequently, the block mask can be approximated by a $2(n+1)$-head masked attention layer. By setting $1/\varepsilon = 4(n+1)^2$, we have $M = \mathcal{O}(1/\sqrt{\varepsilon})$

Combining this block mask with the random number generation and the Bernoulli mask completes the proof. □

D.2.3 Approximating σ_2

Based on the output of σ_1, we can express the input of σ_2 as $\mathcal{H} := [\mathfrak{h}_1 \cdots \mathfrak{h}_{2^n}] \in \mathbb{R}^{D\times 2^n}$, where for $t \in [2^n]$:

$$\mathfrak{h}_t = \begin{bmatrix} \mathbf{h}_t^{(0)} \\ \vdots \\ \mathbf{h}_t^{(n)} \\ \star \end{bmatrix}, \quad \mathbf{h}_t^{(0)} = \begin{bmatrix} \mathbf{x}_t^{(0)} \\ 0 \\ 0 \\ 0 \\ f_t^{(0)} \end{bmatrix} \quad \mathbf{h}_t^{(1)} = \begin{bmatrix} \mathbf{x}_t^{(1)} \\ 0 \\ 0 \\ f_t^{(0)} \\ f_t^{(1)} \end{bmatrix}, \quad \ldots, \quad \mathbf{h}_t^{(n)} = \begin{bmatrix} \mathbf{x}_t^{(n)} \\ 0 \\ 0 \\ \sum_{p=0}^{n-1} f_t^{(p)} \\ f_t^{(n)} \end{bmatrix}.$$

From the definition of σ_1, we know that some $\mathbf{h}_t^{(i)}$ values are nonzero, while others are zero. If we assume $i_1 < ... < i_k$ $(0 < k \leqslant n)$ such that $\mathbf{h}_t^{(i_1)}, ..., \mathbf{h}_t^{(i_k)} \neq \mathbf{0}$ while all other $\mathbf{h}_t^{(n)}$ values are $\mathbf{0}$, the goal of σ_2 is to ensure that $\mathbf{h}_t^{(i_1)} \neq \mathbf{0}$ and the others being $\mathbf{0}$.

We choose $\{\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m\}_{m\in\{0,\ldots,n\}}$ such that for tokens $\mathfrak{h}_t$,

$$\mathbf{Q}_m\mathfrak{h}_t = \begin{bmatrix} t \\ \sum_{p=0}^{m-1} f_t^{(p)} \\ 1 \\ \epsilon_0 \end{bmatrix}, \quad \mathbf{K}_m\mathfrak{h}_k = \begin{bmatrix} -1 \\ -c_0 \\ k \\ 1 \end{bmatrix}, \quad \mathbf{V}_m\mathfrak{h}_k = \left[\mathbf{0}; ..., \mathbf{0}; \mathbf{h}_k^{(m)}; \mathbf{0}; ..., \mathbf{0}\right]$$

where $c_0 > 0$ is sufficiently large and $\epsilon_0 > 0$ is sufficiently small such that $\epsilon_0 - c_0 f_j < 0$ for any $f_j > 0$. In this way, with $k < t$ we have

$$\sum_{m=0}^{n} \mathbb{1}_{>0}\left(\langle \mathbf{Q}_m\mathfrak{h}_t, \mathbf{K}_m\mathfrak{h}_k\rangle\right)\mathbf{V}_m\mathfrak{h}_k = \mathbf{0},$$

and with $k = t$ we have

$$\begin{aligned} \sum_{m=0}^{n} r\left(\langle \mathbf{Q}_m\mathfrak{h}_t, \mathbf{K}_m\mathfrak{h}_t\rangle\right)\mathbf{V}_m\mathfrak{h}_t &= \sum_{m=0}^{n} \mathbb{1}_{>0}\left[\epsilon_0 - c_0 \sum_{p=0}^{m-1} f_t^p\right] \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{h}_k^{(m)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \\ &= \begin{cases} \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{h}_k^{(m)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, & \text{if } \sum_{p=0}^{m-1} f_t^p = 0 \\ \mathbf{0}, & \text{otherwise} \end{cases}. \end{aligned}$$

Since $\sum_{p=0}^{m-1} f_t^p = 0$ is equivalent to $\mathbf{h}_t^{(0)} = \cdots = \mathbf{h}_t^{(p-1)} = \mathbf{0}$, we have for each $t \in [T]$ that

$$\sum_{m=0}^{n} \sum_{k=1}^{t} \mathbb{1}_{>0} \left[\langle \mathbf{Q}_m \mathfrak{h}_t, \mathbf{K}_m \mathfrak{h}_k \rangle \right] \mathbf{V}_m \mathfrak{h}_k = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{h}_t^{(q)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{h}_t^{(0)} = \cdots = \mathbf{h}_t^{(q-1)} = \mathbf{0}$ and $\mathbf{h}_t^{(m)} \neq 0$. Finally, referring the approximation of $\mathbb{1}_{>0}[\cdot]$ (D.2.1), we can approximate $\mathbb{1}_{>0}[\cdot]$ using 2 ReLU functions. Therefore, σ_2 can be approximated by a $2(n+1)$-head masked attention layer. ☐

D.3 Equivalence of `WS` and `RM` with previous works

D.3.1 `WS`

Sliding windows used in previous works (e.g., (Cheung et al., 2022; Fiandri et al., 2024)) can be regarded as special cases of `WS`. Specifically, they are equivalent to `WS` when `WS` contains only one instance, all windows are of equal size, and they are connected without overlap. Figure 9 shows the equivalence between the stochastic instance scheduler in MASTER and the sliding window schedulers from previous works.

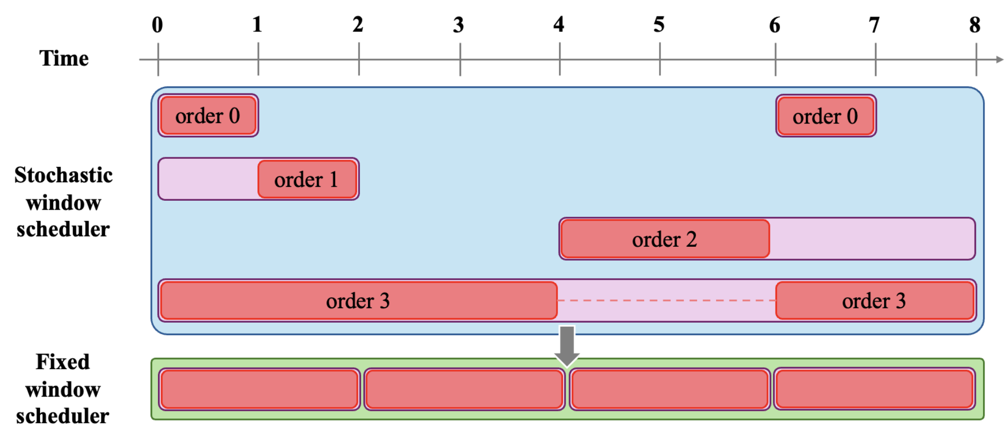

Figure 9: Equivalence between the stochastic instance scheduler in MASTER and the sliding window schedulers from previous works. Purplish blocks represent scheduled blocks, while reddish blocks represent active ones. Reddish blocks connected by a dashed line are concatenated. Here, $T = 8$ and the sliding window size is 2.

D.3.2 `RM`

The restart strategies in previous works can be classified into two types: stochastic and deterministic (Gomes et al., 1998; Streeter and Golovin, 2008). Stochastic restarts typically involve randomness or a test, whereas deterministic restarts are generally performed after a set number of rounds. Therefore, deterministic restarts can be considered a special case of sliding window strategies. Regarding stochastic restarts, since we have demonstrated that transformers can generate random numbers and that MLPs can perform stationary tests, we can conclude, based on the universal approximation theorem (Definition 8), that stochastic restarts can also be approximated by transformers.

Definition 8 (Universal Approximation Theorem for MLPs)**.** *Let f be a continuous function from a compact subset $K \subseteq \mathbb{R}^n$ to $\mathbb{R}^m$. Suppose $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-polynomial, continuous activation function applied component-wise.*

Then, for any $\epsilon > 0$, there exists a single hidden-layer MLP that approximates f to within ϵ. Specifically, there exist weights $A \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$, $C \in \mathbb{R}^{m \times k}$, and a sufficiently large number of hidden units k such that the MLP

$$g(x) = C \cdot \sigma(A \cdot x + b)$$

satisfies

$$\sup_{x \in K} \|f(x) - g(x)\| < \epsilon.$$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly stated the main claim in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We stated the limitation in the last section for discussion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have described the full assumption, statements, and proof in the main body and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have described the detailed experimental details in the corresponding section in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We clearly stated the source of data. Also, we will open the source code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have clearly described the experimental details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We have repeated the experimental multiple times and report the standard deviation comes from the replication.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Our experiments are small-scale and implementable by a small laptop. Also, we do not pursue the computational cost in this study, so the computational resource is out of our focus.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have checked the code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: A main focus of this study is fundamental, so there is almost no effect on social impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [No]

Justification: Since this paper is theoretical, the outcome does not have a righ risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited the source of the data which we use.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We have not created a new asset throughout this study.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: We have not performed the crowdsourcing experiments and others.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: Since this study is fundamental, there is no potential risk on this point.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: We have used LLM only for the formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.