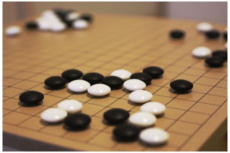
Policy Optimization for Stochastic Shortest Path

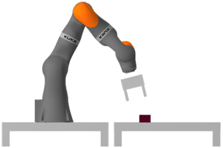
Liyu Chen, Haipeng Luo, Aviv Rosenberg
University of Southern California

August 31, 2022

Motivation

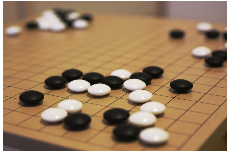
Many real-world applications can be modelled by goal-oriented reinforcement learning.

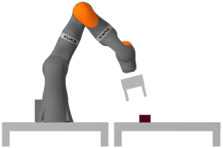




Motivation

Many real-world applications can be modelled by goal-oriented reinforcement learning.





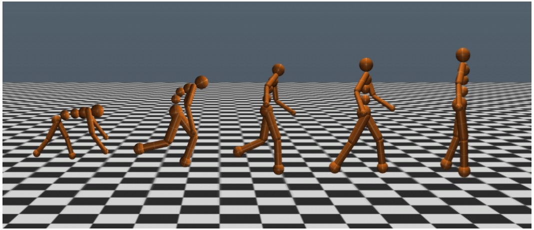
Goal-oriented reinforcement learning can be formulated as Stochastic Shortest Path (SSP) problem.

- Episodic MDP with a goal state.
- The objective is to reach the goal state with minimum cost.

Motivation

Policy Optimization (PO) is among the most popular methods in reinforcement learning

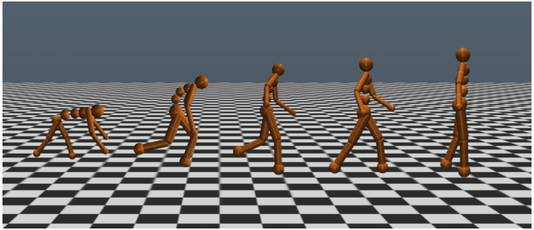
- Widely applied in practice: REINFORCE [W1992], TRPO [SLMJA2017], PPO [SWDRK2017], etc.
- Easy to implement, computationally efficient.



Motivation

Policy Optimization (PO) is among the most popular methods in reinforcement learning

- Widely applied in practice: REINFORCE [W1992], TRPO [SLMJA2017], PPO [SWDRK2017], etc.
- Easy to implement, computationally efficient.
- Easily handle different types of environments: stochastic or adversarial costs [SERM2020], function approximation [CYJW2020], non-stationary environments [FYWX2020], etc.



Our Contributions

We propose the first set of PO algorithms for SSP.

1. Stacked Discounted Approximation (SDA): approximate SSP by a simpler MDP with some best-of-both-worlds property.
2. Near optimal regret bounds in various settings including both stochastic costs and adversarial costs.

Problem Formulation

for *episode* $k = 1, \ldots, K$ **do**

  learner starts in state $s_1^k = s_0 \in \mathcal{S}$, $i \leftarrow 1$

Problem Formulation

for *episode* $k = 1, \ldots, K$ **do**

- learner starts in state $s_1^k = s_0 \in \mathcal{S}, i \leftarrow 1$
- **while** $s_k^i \neq g$ **do**
  - learner chooses action $a_i^k \in \mathcal{A}$, suffer cost c_i^k (may not observe immediately), and observes state $s_{i+1}^k \sim P(\cdot|s_i^k, a_i^k)$
  - $i \leftarrow i + 1$

Problem Formulation

for *episode* $k = 1, \ldots, K$ **do**

  learner starts in state $s_1^k = s_0 \in \mathcal{S}, i \leftarrow 1$

  while $s_k^i \neq g$ **do**

    learner chooses action $a_i^k \in \mathcal{A}$, suffer cost c_i^k (may not observe immediately), and observes state $s_{i+1}^k \sim P(\cdot | s_i^k, a_i^k)$

    $i \leftarrow i + 1$

$$\text{Regret:} \quad R_K = \sum_{k=1}^{K} \sum_{i=1}^{I_k} c_i^k - \sum_{k=1}^{K} V_k^{\pi^\star}(s_0)$$

where $V_k^\pi(s)$ is the expected cost of policy π starting from s in episode k, $\pi^\star = \operatorname{argmin}_{\pi \in \Pi} \sum_{k=1}^{K} V_k^\pi(s_0)$, and Π is the set of proper policies which reaches g with probability 1.

Feedback Type

- **Stochastic Environments:** there exists a fixed unknown mean cost function $c \in [0, 1]^{\mathcal{S} \times \mathcal{A}}$.

Feedback Type

- **Stochastic Environments:** there exists a fixed unknown mean cost function $c \in [0,1]^{\mathcal{S}\times\mathcal{A}}$.
  - *Stochastic Costs (SC):* whenever learner visits (s,a), it immediately observes an i.i.d cost sample with mean $c(s,a)$.

Feedback Type

- **Stochastic Environments:** there exists a fixed unknown mean cost function $c \in [0,1]^{\mathcal{S}\times\mathcal{A}}$.
  - *Stochastic Costs (SC):* whenever learner visits (s, a), it immediately observes an i.i.d cost sample with mean $c(s, a)$.
  - *Stochastic Adversary, Full information (SAF):* before learning starts, adversary samples K i.i.d cost functions $\{c_k\}_{k=1}^{K}$ with mean c; learner suffers $c_k(s, a)$ when it visits (s, a), and the learner observes the entire cost function c_k at the end of episode k.

Feedback Type

- **Stochastic Environments:** there exists a fixed unknown mean cost function $c \in [0,1]^{\mathcal{S}\times\mathcal{A}}$.
  - *Stochastic Costs (SC):* whenever learner visits (s, a), it immediately observes an i.i.d cost sample with mean $c(s, a)$.
  - *Stochastic Adversary, Full information (SAF):* before learning starts, adversary samples K i.i.d cost functions $\{c_k\}_{k=1}^K$ with mean c; learner suffers $c_k(s, a)$ when it visits (s, a), and the learner observes the entire cost function c_k at the end of episode k.
  - *Stochastic Adversary, Bandit feedback (SAB):* same as above except that the learner observes the costs of visited state-action pairs $\{c_k(s_i^k, a_i^k)\}_{i=1}^{I_k}$ at the end of episode k.

Feedback Type

- **Stochastic Environments:** there exists a fixed unknown mean cost function $c \in [0,1]^{\mathcal{S}\times\mathcal{A}}$.
  - *Stochastic Costs (SC):* whenever learner visits (s, a), it immediately observes an i.i.d cost sample with mean $c(s, a)$.
  - *Stochastic Adversary, Full information (SAF):* before learning starts, adversary samples K i.i.d cost functions $\{c_k\}_{k=1}^{K}$ with mean c; learner suffers $c_k(s, a)$ when it visits (s, a), and the learner observes the entire cost function c_k at the end of episode k.
  - *Stochastic Adversary, Bandit feedback (SAB):* same as above except that the learner observes the costs of visited state-action pairs $\{c_k(s_i^k, a_i^k)\}_{i=1}^{I_k}$ at the end of episode k.

- **Adversarial Environments:** in episode k, the environment picks a cost function c_k possibly depending on the interaction history.
  - *Adversarial costs, Full information (AF)*
  - *Adversarial costs, Bandit feedback (AB)*

Our Results

We obtain near optimal regret bounds in various settings.

	Regret	Time	Space	Feedback
Cohen et al., 2021	$B_\star\sqrt{SAK}$	$S^3A^2T_{\max}$	$S^2AT_{\max}$	SC
Our work	$B_\star S\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
Chen and Luo, 2021	$\sqrt{DT_\star K}+DS\sqrt{AK}$	$\mathrm{poly}(S,A,T_{\max})\cdot K$	$S^2AT_{\max}$	SAF
Our work	$\sqrt{DT_\star K}+DS\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
Chen and Luo, 2021	$\sqrt{SADT_\star K}+DS\sqrt{AK}$	$\mathrm{poly}(S,A,T_{\max})\cdot K$	$S^2AT_{\max}$	SAB
Our work	$\sqrt{SADT_\star K}+DS\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
Chen and Luo, 2021	$\sqrt{S^2ADT_\star K}$	$\mathrm{poly}(S,A,T_{\max})\cdot K$	$S^2AT_{\max}$	AF
Our work	$\sqrt{(S^2A+T_\star)DT_\star K}$	$S^2AT_{\max}K$	S^2A	
Chen and Luo, 2021	$\sqrt{S^3A^2DT_\star K}$	$\mathrm{poly}(S,A,T_{\max})\cdot K$	$S^2AT_{\max}$	AB
Our work	$\sqrt{S^2AT_{\max}^5K}$	$\mathrm{poly}(S,A,T_{\max})\cdot K$	S^2A	

$B_\star=\max_s V^{\pi^\star}(s)$, $T_\star=T^{\pi^\star}(s_{\text{init}})$, $T_{\max}=\max_s T^{\pi^\star}(s)$, and $D=\max_s\min_\pi T^\pi(s)$, where $T^\pi(s)$ is the hitting time of policy π starting from state s.

Stacked Discounted Approximation

Issue: PO requires policy evaluation, which does not make sense for policies that may not reach the goal.

Stacked Discounted Approximation

Issue: PO requires policy evaluation, which does not make sense for policies that may not reach the goal.
Solution: Approximate SSP by a simpler MDP model, where all policies are proper.

	Optimal regret?	Stationary policy?
Finite-Horizon	Yes	No ($\times$ horizon)
Discounted	No	Yes

Stacked Discounted Approximation

Issue: PO requires policy evaluation, which does not make sense for policies that may not reach the goal.
Solution: Approximate SSP by a simpler MDP model, where all policies are proper.

	Optimal regret?	Stationary policy?
Finite-Horizon	Yes	No ($\times$ horizon)
Discounted	No	Yes

$$\textit{Finite-Horizon} + \textit{Discounted} = \textit{?}$$

Question: Can we obtain a best-of-both-world approximation?

Stacked Discounted Approximation

Finite-Horizon + *Discounted* = **Stacked Discounted**

$\mathcal{M} \to \widetilde{\mathcal{M}}$: stack H γ-discounted MDPs

- State space: $\mathcal{S} \times [H+1]$.
- In each step the learner transits to the next layer w.p. $1-\gamma$: $P((s',h)|(s,h),a) = \gamma P(s'|s,a)$, $P((s',h+1)|(s,h),a) = (1-\gamma)P(s'|s,a)$, and $P(g|(s,h),a) = P(g|s,a)$.

Stacked Discounted Approximation

Finite-Horizon + *Discounted* = **Stacked Discounted**

$\mathcal{M} \to \widetilde{\mathcal{M}}$: stack H γ-discounted MDPs

- State space: $\mathcal{S} \times [H+1]$.
- In each step the learner transits to the next layer w.p. $1-\gamma$: $P((s',h)|(s,h),a) = \gamma P(s'|s,a)$, $P((s',h+1)|(s,h),a) = (1-\gamma)P(s'|s,a)$, and $P(g|(s,h),a) = P(g|s,a)$.
- When $h = H+1$, the learner suffers a terminal cost of $\tilde{\mathcal{O}}(D)$ (high probability upper bound on the cost of executing fast policy).

Stacked Discounted Approximation

Finite-Horizon + *Discounted* = **Stacked Discounted**

$\mathcal{M} \rightarrow \widetilde{\mathcal{M}}$: stack H γ-discounted MDPs

- State space: $\mathcal{S} \times [H+1]$.
- In each step the learner transits to the next layer w.p. $1-\gamma$: $P((s',h)|(s,h),a) = \gamma P(s'|s,a)$, $P((s',h+1)|(s,h),a) = (1-\gamma)P(s'|s,a)$, and $P(g|(s,h),a) = P(g|s,a)$.
- When $h = H+1$, the learner suffers a terminal cost of $\tilde{\mathcal{O}}(D)$ (high probability upper bound on the cost of executing fast policy).

$\widetilde{\pi} \rightarrow \pi = \sigma(\widetilde{\pi})$: maintain a counter h, increase h by 1 with probability $1-\gamma$ at every time step; follow $\widetilde{\pi}(\cdot|s,h)$ for $h \leq H$, and switch to fast policy when $h = H+1$.

Stacked Discounted Approximation

Observation 1: With finite-horizon approximation, we need horizon of $\mathcal{O}(T_{\max} \ln \frac{1}{\epsilon})$ to achieve ϵ approximation error.

$T_{\max}$: expected hitting time of $\pi^\star$ over all states

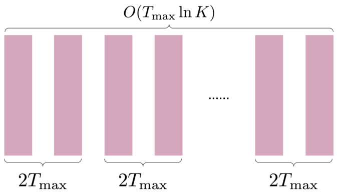
Stacked Discounted Approximation

Observation 1: With finite-horizon approximation, we need horizon of $\mathcal{O}(T_{\max} \ln \frac{1}{\epsilon})$ to achieve ϵ approximation error.
Observation 2: a discounted MDP with discount factor $\gamma \approx$ a finite-horizon MDP with horizon $\frac{1}{1-\gamma}$ (compressed representation).

$T_{\max}$: expected hitting time of $\pi^\star$ over all states

Stacked Discounted Approximation

Observation 1: With finite-horizon approximation, we need horizon of $\mathcal{O}(T_{\max} \ln \frac{1}{\epsilon})$ to achieve ϵ approximation error.
Observation 2: a discounted MDP with discount factor $\gamma \approx$ a finite-horizon MDP with horizon $\frac{1}{1-\gamma}$ (compressed representation).
Implication: Setting $\gamma = 1 - \frac{1}{T_{\max}}$, we only need $H = \mathcal{O}(\ln K)$ layers to achieve $1/K$ approximation error. This gives a nearly stationary policy that only changes $\mathcal{O}(\ln K)$ times.



$T_{\max}$: expected hitting time of $\pi^\star$ over all states

Template of PO Algorithm for SSP

Initialize: $\mathcal{P}_1$ as the set of all possible transitions in $\widetilde{\mathcal{M}}$, $\eta > 0$ some learning rate.

for $k = 1, \ldots, K$ **do**

Compute $\pi_k(a|s,h) \propto \exp\left(-\eta \sum_{j=1}^{k-1}(\widetilde{Q}_j(s,a,h) - B_j(s,a,h))\right)$, where $\widetilde{Q}_j$ is some optimistic action-value estimator and B_j is some exploration bonus.

Template of PO Algorithm for SSP

Initialize: $\mathcal{P}_1$ as the set of all possible transitions in $\widetilde{\mathcal{M}}$, $\eta > 0$ some learning rate.
for $k = 1, \ldots, K$ **do**

Compute $\pi_k(a|s,h) \propto \exp\left(-\eta \sum_{j=1}^{k-1}(\widetilde{Q}_j(s,a,h) - B_j(s,a,h))\right)$, where $\widetilde{Q}_j$ is some optimistic action-value estimator and B_j is some exploration bonus.
Execute $\sigma(\pi_k)$ for one episode.
Compute Bernstein-style transition confidence set $\mathcal{P}_{k+1}$.

Template of PO Algorithm for SSP

Initialize: $\mathcal{P}_1$ as the set of all possible transitions in $\widetilde{\mathcal{M}}$, $\eta > 0$ some learning rate.
for $k = 1, \ldots, K$ **do**

Compute $\pi_k(a|s,h) \propto \exp\left(-\eta \sum_{j=1}^{k-1}(\widetilde{Q}_j(s,a,h) - B_j(s,a,h))\right)$, where $\widetilde{Q}_j$ is some optimistic action-value estimator and B_j is some exploration bonus.

Execute $\sigma(\pi_k)$ for one episode.

Compute Bernstein-style transition confidence set $\mathcal{P}_{k+1}$.

Optimistic Value Functions: denote by $Q^{\pi,\mathcal{P},c}$ (or $V^{\pi,\mathcal{P},c}$) the optimistic action-value function (or value function) w.r.t policy π, transition confidence set $\mathcal{P}$, and cost function c.

PO Algorithms for SSP: Stochastic Environments

Algorithm Design: simply set $B_k(s, a, h) = 0$, and

- **Action-value estimator** $\widetilde{Q}_k = Q^{\pi_k, \mathcal{P}_k, \widetilde{c}_k}$ for some cost function $\widetilde{c}_k$:

$$\widetilde{c}_k = (1 + \lambda \widehat{Q}_k(s, a, h))\widehat{c}_k(s, a, h) + e_k(s, a, h),$$

  where $\widehat{Q}_k = Q^{\pi_k, \mathcal{P}_k, \widehat{c}_k}$, $\widehat{c}_k$ is some standard cost estimator, and e_k is some correction term specified below.

PO Algorithms for SSP: Stochastic Environments

Algorithm Design: simply set $B_k(s,a,h)=0$, and

- **Action-value estimator** $\widetilde{Q}_k = Q^{\pi_k,\mathcal{P}_k,\widetilde{c}_k}$ for some cost function $\widetilde{c}_k$:

$$\widetilde{c}_k = (1+\lambda\widehat{Q}_k(s,a,h))\widehat{c}_k(s,a,h) + e_k(s,a,h),$$

  where $\widehat{Q}_k = Q^{\pi_k,\mathcal{P}_k,\widehat{c}_k}$, $\widehat{c}_k$ is some standard cost estimator, and e_k is some correction term specified below.
- **Optimistic cost estimator** $\widehat{c}_k$ is some standard Bernstein-style optimistic cost estimator, such that $\widehat{c}_k(s,a)\le c(s,a)$ with high probability.
- **Correction term** $e_k(s,a,h)$ is 0 for stochastic costs (SC); $(8\iota\sqrt{\widehat{c}_k(s,a,h)/k}+\beta'\widehat{Q}_k(s,a,h))\mathbb{I}\{h\le H\}$ for stochastic adversary with full information; and $\beta\widehat{Q}_k(s,a,h)\mathbb{I}\{h\le H\}$ for stochastic adversary with bandit feedback.

PO Algorithms for SSP: Stochastic Environments

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(B_\star S\sqrt{AK})$ *with stochastic costs;*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star K} + DS\sqrt{AK})$ *with stochastic adversary, full information; and*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star SAK} + DS\sqrt{AK})$ *with stochastic adversary, bandit feedback.*

PO Algorithms for SSP: Stochastic Environments

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(B_\star S\sqrt{AK})$ *with stochastic costs;*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star K} + DS\sqrt{AK})$ *with stochastic adversary, full information; and*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star SAK} + DS\sqrt{AK})$ *with stochastic adversary, bandit feedback.*

Analysis Highlights

- A new correction term $\lambda\widehat{c}_k(s,a,h)\widehat{Q}_k(s,a,h)$ to deal with transition estimation error, which requires a regret bound starting from any state-action pair that PO enjoys.

PO Algorithms for SSP: Stochastic Environments

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(B_\star S\sqrt{AK})$ *with stochastic costs;*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star K} + DS\sqrt{AK})$ *with stochastic adversary, full information; and*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star SAK} + DS\sqrt{AK})$ *with stochastic adversary, bandit feedback.*

Analysis Highlights

- A new correction term $\lambda\widehat{c}_k(s,a,h)\widehat{Q}_k(s,a,h)$ to deal with transition estimation error, which requires a regret bound starting from any state-action pair that PO enjoys.
- Carefully designed correction term e_k to deal with cost estimation error under different feedback types.

PO Algorithms for SSP: Stochastic Environments

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(B_\star S\sqrt{AK})$ *with stochastic costs;*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star K} + DS\sqrt{AK})$ *with stochastic adversary, full information; and*
$R_K = \tilde{\mathcal{O}}(\sqrt{DT_\star SAK} + DS\sqrt{AK})$ *with stochastic adversary, bandit feedback.*

Analysis Highlights

- A new correction term $\lambda\widehat{c}_k(s,a,h)\widehat{Q}_k(s,a,h)$ to deal with transition estimation error, which requires a regret bound starting from any state-action pair that PO enjoys.
- Carefully designed correction term e_k to deal with cost estimation error under different feedback types.
- An improved PO analysis that reduces the cost of policy update from $\tilde{\mathcal{O}}(\sqrt{K})$ to $\tilde{\mathcal{O}}(K^{1/4})$.

PO Algorithms for SSP: Adversarial Environments

Algorithm Design (Full Information)

- **Action-value estimator** $\widetilde{Q}_k = Q^{\pi_k, \mathcal{P}_k, \widetilde{c}_k}$, where $\widetilde{c}_k = (1 + \lambda \widehat{Q}_k(s,a,h))c_k(s,a,h)$ and $\widehat{Q}_k = Q^{\pi_k, \mathcal{P}_k, c_k}$.
- **Dilated bonus** $B_k = B^{\pi_k, \mathcal{P}_k, b_k}$, where $b_k(s,a,h) = 2\eta \sum_a \pi_k(a|s,h)\widetilde{A}_k(s,a,h)^2$, $\widetilde{A}_k(s,a,h) = \widetilde{Q}_k(s,a,h) - \widetilde{V}_k(s,h)$, $\widetilde{V}_k = V^{\pi_k, \mathcal{P}_k, \widetilde{c}_k}$, and $B^{\pi,\mathcal{P},b}$ is defined as: $B^{\pi,\mathcal{P},b}(s,a,H+1) = b(s,a,H+1)$ and

$$B^{\pi,\mathcal{P},b}(s,a,h) = b(s,a,h) + \left(1 + \frac{1}{H'}\right) \max_{\widehat{P} \in \mathcal{P}} \widehat{P}_{s,a,h}\left(\sum_{a'} \pi(a'|\cdot,\cdot) B^{\pi,\mathcal{P},b}(\cdot, a', \cdot)\right),$$

where $H' = \frac{8(H+1)\ln(2K)}{1-\gamma}$ is the dilated coefficient.

PO Algorithms for SSP: Adversarial Environments

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(T_\star\sqrt{DK} + \sqrt{S^2 ADT_\star K})$.

Analysis Highlights

- A shifting argument to obtain a refined stability term w.r.t the advantage function.
- Dilated bonus + correction term $\lambda\widehat{Q}_k(s,a,h)c_k(s,a,h)$ to transform the stability term into a term of order $\tilde{\mathcal{O}}(\eta DT_\star K)$ ($\tilde{\mathcal{O}}(\eta T_\star T_{\max}^2 K)$ by vanilla analysis).

PO Algorithms for SSP: Adversarial Environments

Algorithm Design and Analysis (Bandit Feedback): mainly follows (Luo et al., 2021) with components adapted to the stacked discounted MDP.

Theorem

With the instantiation above, we have $R_K = \tilde{\mathcal{O}}(\sqrt{S^2 A T_{\max}^5 K})$.

Conclusion

We propose the first set of PO algorithms for SSP.

1. Stacked Discounted Approximation (SDA): approximate SSP by a simpler MDP with small bias, and learns nearly stationary policy.
2. Near optimal regret bounds in various settings.

	Regret	Time	Space	Feedback
[CEMR21]	$B_\star\sqrt{SAK}$	$S^3A^2T_{\max}$	$S^2AT_{\max}$	SC
Our work	$B_\star S\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
[CL21]	$\sqrt{DT_\star K} + DS\sqrt{AK}$	$\text{poly}(S, A, T_{\max}) \cdot K$	$S^2AT_{\max}$	SAF
Our work	$\sqrt{DT_\star K} + DS\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
[CL21]	$\sqrt{SADT_\star K} + DS\sqrt{AK}$	$\text{poly}(S, A, T_{\max}) \cdot K$	$S^2AT_{\max}$	SAB
Our work	$\sqrt{SADT_\star K} + DS\sqrt{AK}$	$S^2AT_{\max}K$	S^2A	
[CL21]	$\sqrt{S^2ADT_\star K}$	$\text{poly}(S, A, T_{\max}) \cdot K$	$S^2AT_{\max}$	AF
Our work	$\sqrt{(S^2A + T_\star)DT_\star K}$	$S^2AT_{\max}K$	S^2A	
[CL21]	$\sqrt{S^3A^2DT_\star K}$	$\text{poly}(S, A, T_{\max}) \cdot K$	$S^2AT_{\max}$	AB
Our work	$\sqrt{S^2AT_{\max}^5 K}$	$\text{poly}(S, A, T_{\max}) \cdot K$	S^2A	