Randomized Exploration in Cooperative Multi-Agent Reinforcement Learning

Hao-Lun Hsu*, Weixin Wang*, Miroslav Pajic, Pan Xu

Department of Electrical and Computer Engineering

Duke University

Motivation

- **Thompson Sampling (TS)**
  - Outperform Upper Confidence Bound (UCB) empirically
  - **Not easily scalable to large environments** (multi-agent scenarios)
- **Randomized Exploration**
  - Effective in bandit and single-agent RL
  - **Remain underexplored** in Cooperative MARL

Background and Setup

Parallel MDPs

- M agents interact independently with their respective MDPs
- Share the **same but independent state and action spaces**
- Each agent **might have its unique reward functions and transition kernels**
- Agents and server can communicate to share data

Unified Algorithm Framework

Algorithm Unified Algorithm Framework for Randomized Exploration in Parallel MDPs

for episode $k = 1, ..., K$ **do**
  for agent $m \in \mathcal{M}$ **do**
    Receive initial state $s^k_{m,1}$ and $V^k_{m,H+1}(\cdot) \leftarrow 0$. .
    $\{Q^k_{m,h}(\cdot,\cdot), V^k_{m,h}(\cdot,\cdot)\}_{h=1}^{H} \leftarrow$ **Randomized Exploration** $\triangleleft$ **PHE** or **LMC**
    for step $h = 1, ..., H$ **do**
      $a^k_{m,h} \leftarrow \mathrm{argmax}_{a \in \mathcal{A}}\, Q^k_{m,h}(s^k_{m,h}, a)$.
      Receive $s^k_{m,h+1}$ and $r_{m,h}$, then update local data.
      if **Condition** **then**
        SYNCHRONIZE $\leftarrow$ True.
      end if
    end for
  end for
  if SYNCHRONIZE **then**
    All the agents upload their newly collected local data to **the server**. $U^{\mathrm{loc}}_{m,h}(k)$
    The server gathers all information and sends it back to **each agent**. $U^{\mathrm{ser}}_{h}(k)$
  end if
end for

Synchronization Conditions

Synchronization Conditions

1) synchronize at a constant frequency
2) synchronize at an exponential frequency
3) synchronize when the following **feature mapping $\phi(\cdot,\cdot)$-based** condition is satisfied

$$\log \frac{\det\left({}^{\text{ser}}\mathbf{\Lambda}_h^k + {}^{\text{loc}}\mathbf{\Lambda}_{m,h}^k + \lambda \mathbf{I}\right)}{\det\left({}^{\text{ser}}\mathbf{\Lambda}_h^k + \lambda \mathbf{I}\right)} \geq \frac{\gamma}{(k-k_s)},$$

where $\phi(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, ${}^{\text{ser}}\mathbf{\Lambda}_h^k = \sum_{U_h^{\text{ser}}(k)} \phi(s^l,a^l)\phi(s^l,a^l)^\top$, ${}^{\text{loc}}\mathbf{\Lambda}_{m,h}^k = \sum_{U_{m,h}^{\text{loc}}(k)} \phi(s^l,a^l)\phi(s^l,a^l)^\top$.

- Regression loss with added random Gaussian noises $\epsilon_h^{k,l,n}$ and $\boldsymbol{\xi}_h^{k,n}$ to **perturb reward and regularizer**

$$\widetilde{L}_{m,h}^{k,n}(\mathbf{w}) = \sum_{l=1}^{\mathcal{K}(k)} L\Big(\big(r_h^l + \epsilon_h^{k,l,n}\big) + V_{m,h+1}^k(s'^l), f\big(\mathbf{w}; \phi^l\big)\Big) + \lambda \big\|\mathbf{w} + \boldsymbol{\xi}_h^{k,n}\big\|^2.$$

- Unified Algorithm Framework + PHE $\Rightarrow$ **CoopTS-PHE**

Algorithm Perturbed-History Exploration

1: **for** step $h = H, ..., 1$ **do**
2:     **for** $n = 1, ..., N$ **do**
3:         Sample $\{\epsilon_h^{k,l,n}\}_{l \in [\mathcal{K}(k)]} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2)$ and $\boldsymbol{\xi}_h^{k,n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ independently.
4:         Obtain the perturbed estimated parameter $\widetilde{\mathbf{w}}_{m,h}^{k,n} = \operatorname{argmin} \widetilde{L}_{m,h}^{k,n}(\mathbf{w})$.
5:     **end for**
6:     $Q_{m,h}^k \leftarrow \min\Big\{\max_{n \in [N]} f\big(\widetilde{\mathbf{w}}_{m,h}^{k,n}; \phi\big), H - h + 1\Big\}^+$.
7:     $V_{m,h}^k(\cdot) \leftarrow \max_{a \in \mathcal{A}} Q_{m,h}^k(\cdot, a)$.
8: **end for**
9: Output: $\{Q_{m,h}^k(\cdot,\cdot), V_{m,h}^k(\cdot,\cdot)\}_{h=1}^H$.

Langevin Monte Carlo Exploration

- **Langevin Monte Carlo update**: for iterate $j = 1, \ldots, J_k$, the update is given by

$$\mathbf{w}_{m,h}^{k,j,n} = \mathbf{w}_{m,h}^{k,j-1,n} - \eta_{m,k} \nabla L_{m,h}^{k}\big(\mathbf{w}_{m,h}^{k,j-1,n}\big) + \sqrt{2\eta_{m,k}\beta_{m,k}^{-1}}\,\boldsymbol{\epsilon}_{m,h}^{k,j,n}.$$

- Unified Algorithm Framework + LMC $\Rightarrow$ **CoopTS-LMC**

Algorithm Langevin Monte Carlo Exploration

for step $h = H, \ldots, 1$ **do**
 for $n = 1, \ldots, N$ **do**
  $\mathbf{w}_{m,h}^{k,0,n} = \mathbf{w}_{m,h}^{k-1,J_{k-1},n}$.
  for $j = 1, \ldots, J_k$ **do**
   Sample $\boldsymbol{\epsilon}_{m,h}^{k,j,n} \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$ and obtain $\mathbf{w}_{m,h}^{k,j,n}$ through **LMC update**.
  end for
 end for
 $Q_{m,h}^{k} \leftarrow \min\big\{ \max_{n \in [N]} f\big(\mathbf{w}_{m,h}^{k,J_k,n}; \phi\big), H - h + 1 \big\}^{+}$
 $V_{m,h}^{k}(\cdot) \leftarrow \max_{a \in A} Q_{m,h}^{k}(\cdot, a)$.
end for

Definition

Linear MDP **(linear reward and transition functions)** An MDP$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is a **linear MDP** with feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, if for any $h \in [H]$, there exist d unknown measures $\boldsymbol{\mu}_h = (\mu_h^1, \ldots, \mu_h^d)$ over $\mathcal{S}$ and an unknown vector $\boldsymbol{\theta}_h \in \mathbb{R}^d$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\mathbb{P}_h(\cdot|s, a) = \langle \phi(s, a), \boldsymbol{\mu}_h(\cdot) \rangle, \quad r_h(s, a) = \langle \phi(s, a), \boldsymbol{\theta}_h \rangle.$$

Cumulative Group Regret The learning goal is to minimize the **cumulative group regret** among M agents after K episodes, which is defined as

$$\text{Regret}(K) = \sum_{m \in \mathcal{M}} \sum_{k=1}^{K} \left[V_{m,1}^*(s_{m,1}^k) - V_{m,1}^{\pi_m^k}(s_{m,1}^k) \right].$$

Theoretical Results

Table: Comparison on episodic, non-stationary, linear MDPs. **We define the average regret as the cumulative regret divided by the total number of samples used by the algorithm.** Here d is the feature dimension, H is the episode length, K is the number of episodes, and M is the number of agents in a multi-agent setting.

Setting	Algorithm	Cumulative Group Regret	Average Regret	Randomized Exploration	Generalizable to Deep RL	Communication Complexity
single-agent	OPT-RLSVI [Zanette et al., 2020]	$\widetilde{\mathcal{O}}(d^2H^{5/2}\sqrt{K})$	$\widetilde{\mathcal{O}}(d^{3/2}H^{3/2}\sqrt{1/K})$	Yes	No	–
	LSVI-UCB [Jin et al., 2020]	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{K})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/K})$	No	No	–
	LSVI-PHE [Ishfaq et al., 2021]	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{K})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/K})$	Yes	Yes	–
	LMC-LSVI [Ishfaq et al., 2024]	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{K})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/K})$	Yes	Yes	–
multi-agent	Coop-LSVI [Dubey & Pentland, 2021]	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{MK})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/MK})$	No	No	$\widetilde{\mathcal{O}}(dHM^3)$
	Asyn-LSVI [Min et al., 2023]	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{K})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/K})$	No	No	$\widetilde{\mathcal{O}}(dHM^2)$
	CoopTS-PHE (Ours)	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{MK})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/MK})$	Yes	Yes	$\widetilde{\mathcal{O}}(dHM^2)$
	CoopTS-LMC (Ours)	$\widetilde{\mathcal{O}}(d^{3/2}H^2\sqrt{MK})$	$\widetilde{\mathcal{O}}(d^{3/2}H\sqrt{1/MK})$	Yes	Yes	$\widetilde{\mathcal{O}}(dHM^2)$

N-Chain Experiments

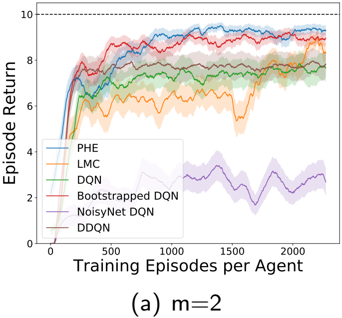
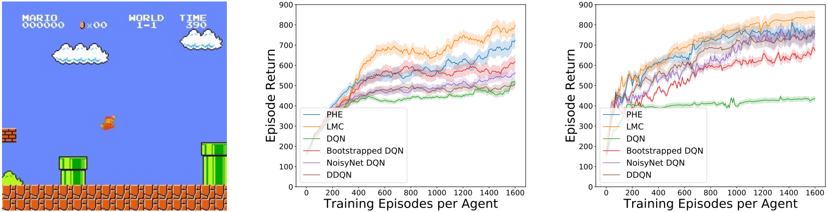


Figure: N-chain task with $N = 25$ states and $m = 2, 3$ agents.

- When $m = 2$, **PHE** achieve **higher average returns** and **LMC eventually catches up**. When $m = 3$, **PHE** and **LMC outperform baselines**. **PHE shows less fluctuation, which supports theoretical results in misspecified settings.**

Super Mario Bro Experiments

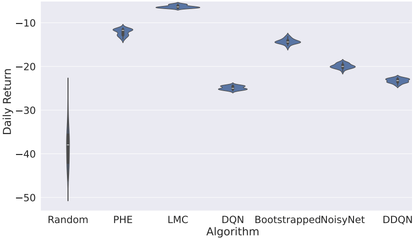


(a) Illustration (b) Parallel MDP (c) Federated Learning

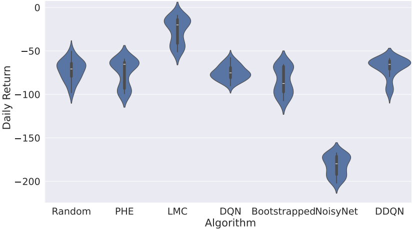
Figure: Super Mario Bros task with $m = 4$ agents: (a) illustration; (b) parallel MDP **(communicate whole data)**; (c) federated learning **(only communicate value function)**.

- **PHE** and **LMC** **outperform baselines in parallel MDP and federated learning settings**. **LMC consistently outperforms PHE**, as the added noise in **PHE** **may not always accurately reflect true posterior in practice**.

Real-world Experiments in Thermal Control



(a) Tampa (hot humid)



(b) Great Falls (cold dry)

Figure: Evaluation performance at Tampa and Great Falls in building energy systems.

- Experiments, trained with parallel data sharing across cities and varying weather, **aim to meet temperature specifications while minimizing electricity use**. **LMC** shows **higher mean returns** in the violin plots.

Summary of Main Contributions

- A unified framework for parallel MDPs + TS-related exploration strategies **PHE** and **LMC** $\rightarrow$ **CoopTS-PHE** and **CoopTS-LMC**.
  - **CoopTS-PHE**: **perturb reward and regularizer (equivalent to TS)**
  - **CoopTS-LMC**: **perform noisy gradient descent (converge to TS)**
- **Regret Upper Bound**: $\widetilde{\mathcal{O}}\left(d^{3/2}H^2\sqrt{M}\left(\sqrt{dM\gamma}+\sqrt{K}\right)\right)$

  Communication Complexity: $\widetilde{\mathcal{O}}\left((d+K/\gamma)MH\right)$
- Extend to **misspecified settings** (approximately linear reward and transition fuctions)
- Extensive experiments on various benchmarks
  - **N-chain** **(require deep exploration)**
  - **Super Mario Bros** **(misspecified setting; federated learning)**
  - **Thermal control problem in building energy systems**

  Outperform existing DQN-based baselines

END

Thank you very much for your valuable time!