Fault-Tolerant Federated Reinforcement Learning with Theoretical Guarantee

Flint Xiaofeng Fan[1,2], Yining Ma[1], Zhongxiang Dai[1], Wei Jing[3], Cheston Tan[2], Bryan Kian Hsiang Low[1]

[1]National University of Singapore
[2]Institute for Infocomm Research, A*STAR
[3]Alibaba DAMO Academy

Outline

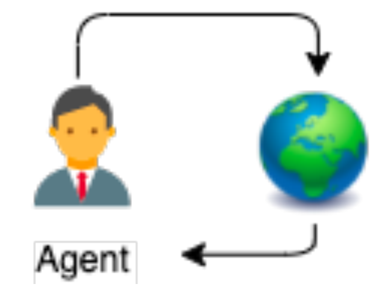
- Motivation & Background
- Problem Setup
- FedPG-BR
- Theoretical Results
- Experiments

Reinforcement Learning (RL)

The optimization problem in RL.

Challenges of RL in real-world applications.

- Poor sample (environment interactions) efficiency.
- One agent owns limited number of samples
  - e.g., patients' medical records



$$D \triangleq \{\tau_i \sim p(\cdot|\boldsymbol{\theta})\}_{i=1}^{B}$$

slow, expensive, fragile

$$\pi_\theta^* = \arg\max_\theta J(\theta) = \int_\tau P(\tau|\theta)R(\tau) = \mathbb{E}_{\boxed{\tau\sim p(\cdot|\theta)}}[R(\tau)]$$

env interaction

Reinforcement Learning (RL)

The optimization problem in RL.

Challenges of RL in real-world applications.

- Poor sample (environment interactions) efficiency
- One agent owns limited number of samples
  - e.g., patients' medical records

$$D \triangleq \{\tau_i \sim p(\cdot|\boldsymbol{\theta})\}_{i=1}^{B}$$

slow, expensive, fragile

$$\pi_\theta^* = \arg\max_\theta J(\theta) = \int_\tau P(\tau|\theta)R(\tau) = \mathbb{E}_{\boxed{\tau\sim p(\cdot|\theta)}}[R(\tau)]$$

env interaction

Observation. Many other agents face the same challenges

Issue. Sharing raw samples is prohibited

Federated Reinforcement Learning (FRL)

Motivation. To build a *better* policy,

- with less trajectories
  - sample efficiency improved
- without sharing trajectories

Applications.

- clinical protocol discovery
- autonomous driving
- IoT devices
- , etc.

a group of self-interested agents

Challenges of FRL

No existing work to provide theoretical guarantee

- Critical drawback due to high sampling cost
- No assurance for practical applications

Vulnerable to random failures or adversarial attacks

- Inherited from Federated Learning
- Poses threats to real-world RL systems

a group of self-interested agents

Challenges of FRL

- No existing work to provide theoretical guarantee
  - Critical drawback due to high sampling cost
  - No assurance for practical applications
- Vulnerable to random failures or adversarial attacks
  - Inherited from Federated Learning
  - Poses threats to real-world RL systems

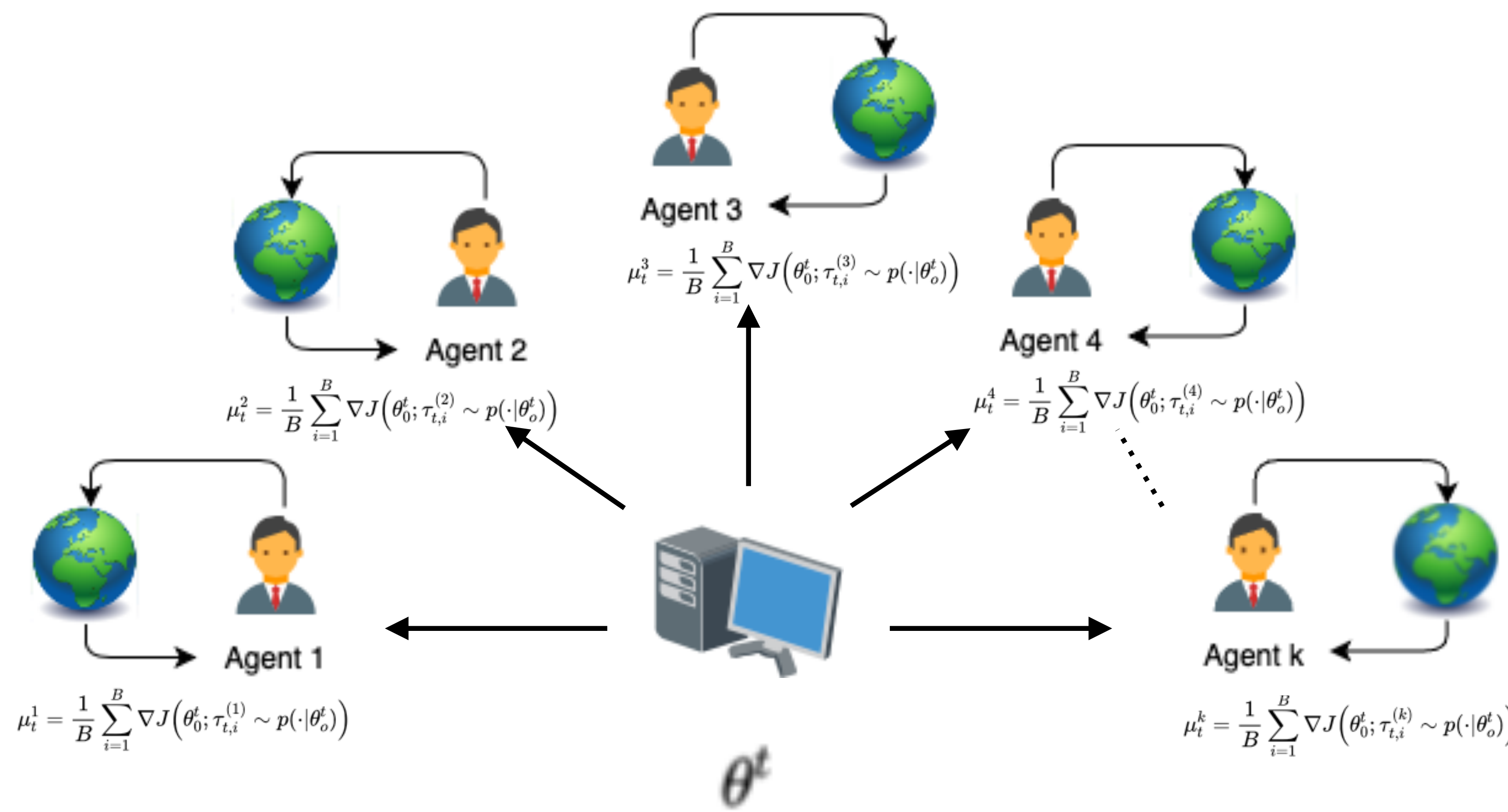
a group of self-interested agents

Simultaneously solved by this work

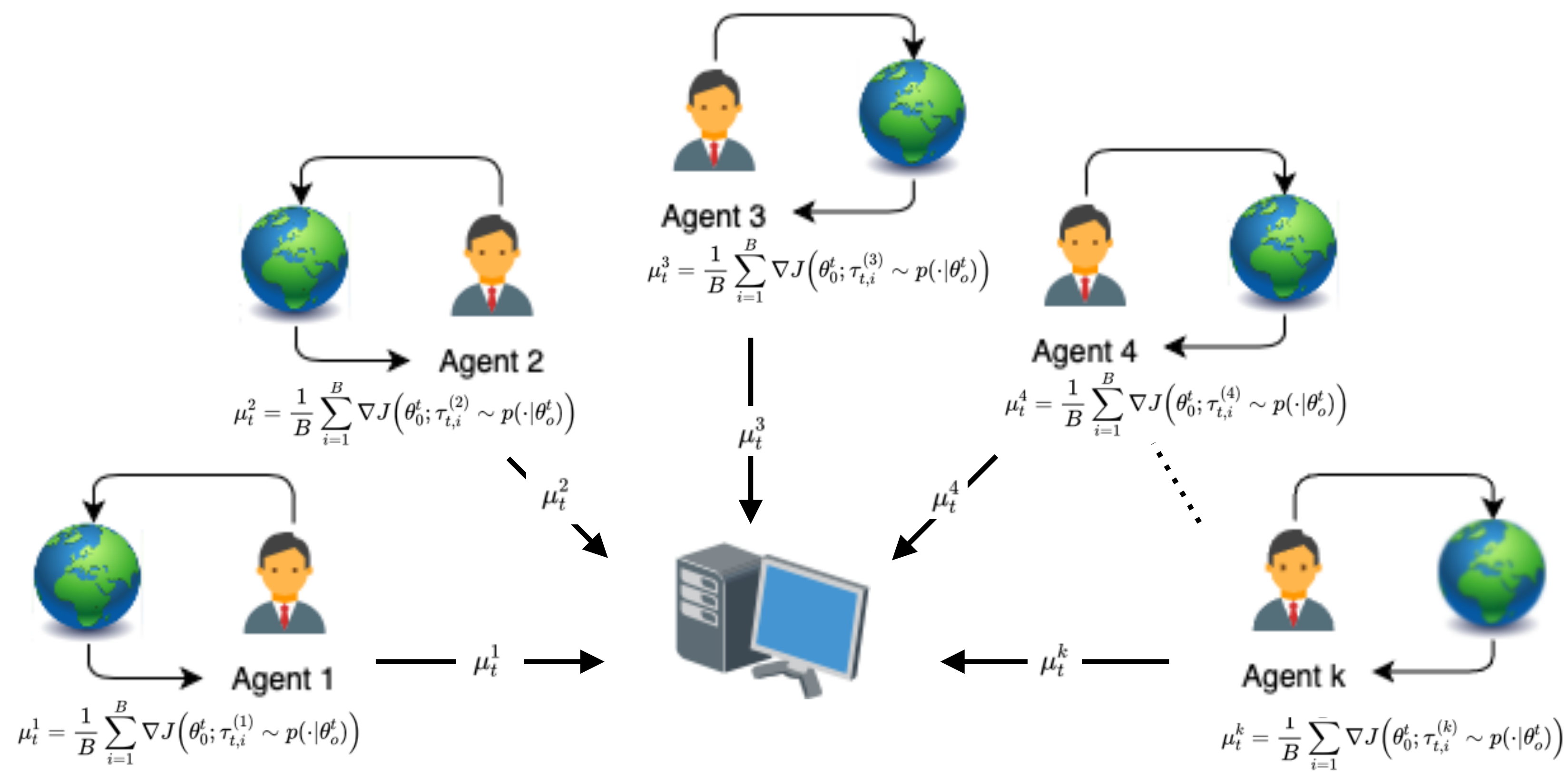
Outline

- Motivation & Background
- Problem Setup
- FedPG-BR
- Theoretical Results
- Experiments

Federated Policy Gradient



Federated Policy Gradient



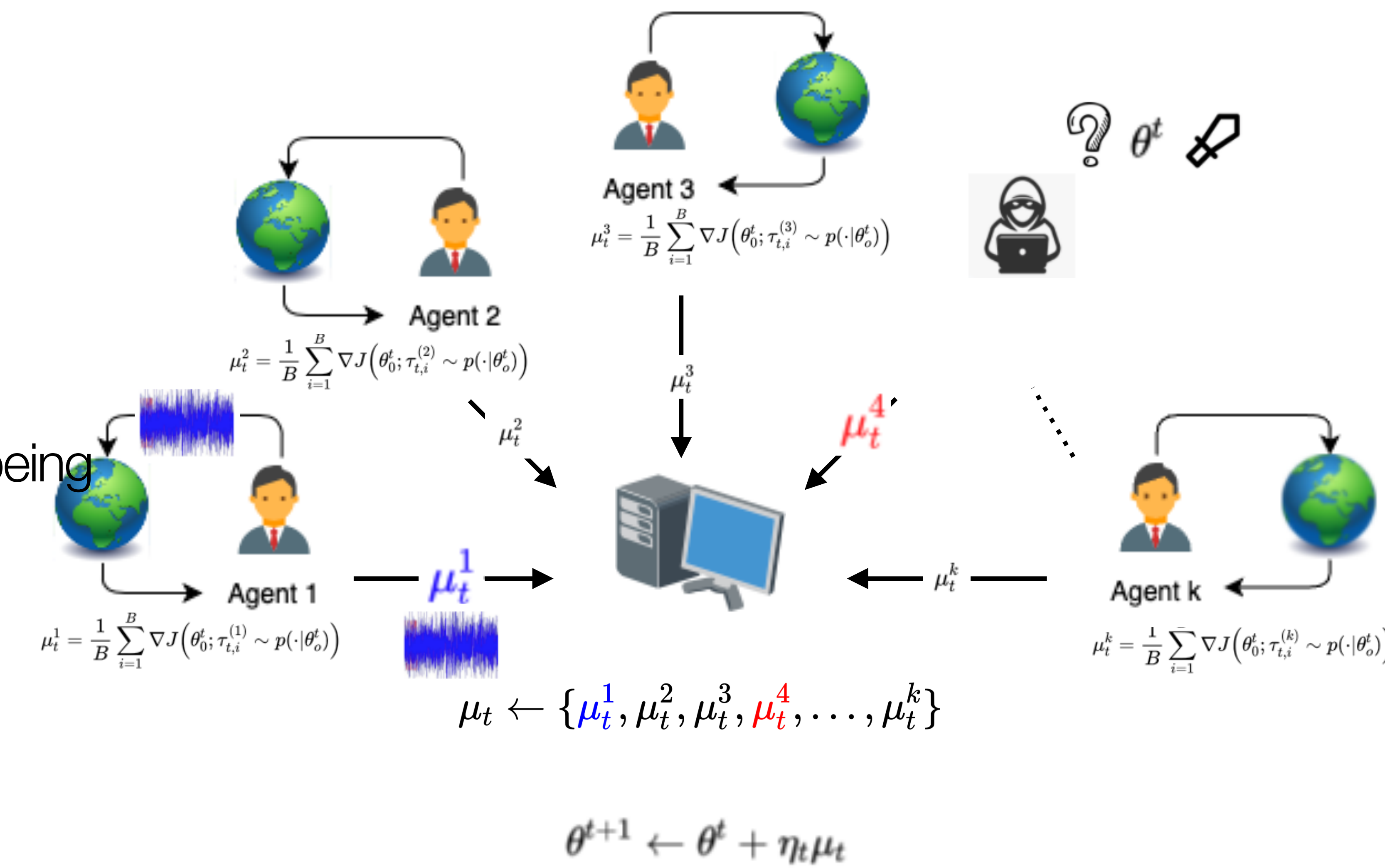
$$\mu_t \leftarrow \{\mu_t^1, \mu_t^2, \mu_t^3, \mu_t^4, \ldots, \mu_t^k\}$$

$$\theta^{t+1} \leftarrow \theta^t + \eta_t \mu_t$$

Practical FRL Systems

The Byzantine Failure Model

- The Byzantine General Problem
- Most stringent fault formalism
- A small fraction[1] of faulty agents being
  - random failures, or
  - adversarial attackers
- The system has no knowledge



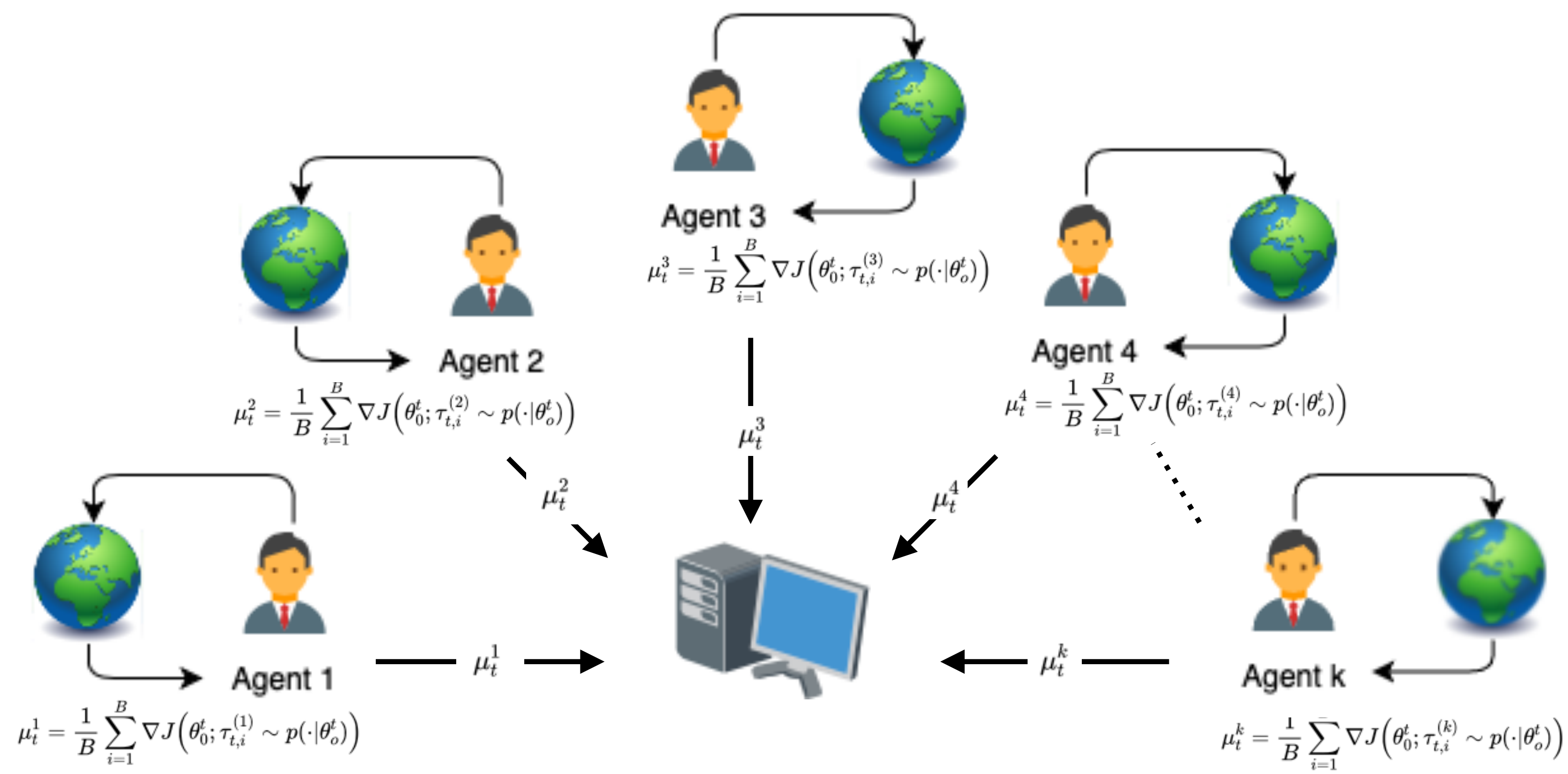
Aim. Build a federated RL policy in this setup with guaranteed improvement

[1]Typically less than half

Outline

- Motivation & Background
- Problem Setup
- FedPG-BR
- Theoretical Results
- Experiments

Federated Policy Gradient using SCSG Optimization



$$\mu_t \leftarrow \{\mu_t^1, \mu_t^2, \mu_t^3, \mu_t^4, \ldots, \mu_t^k\}$$

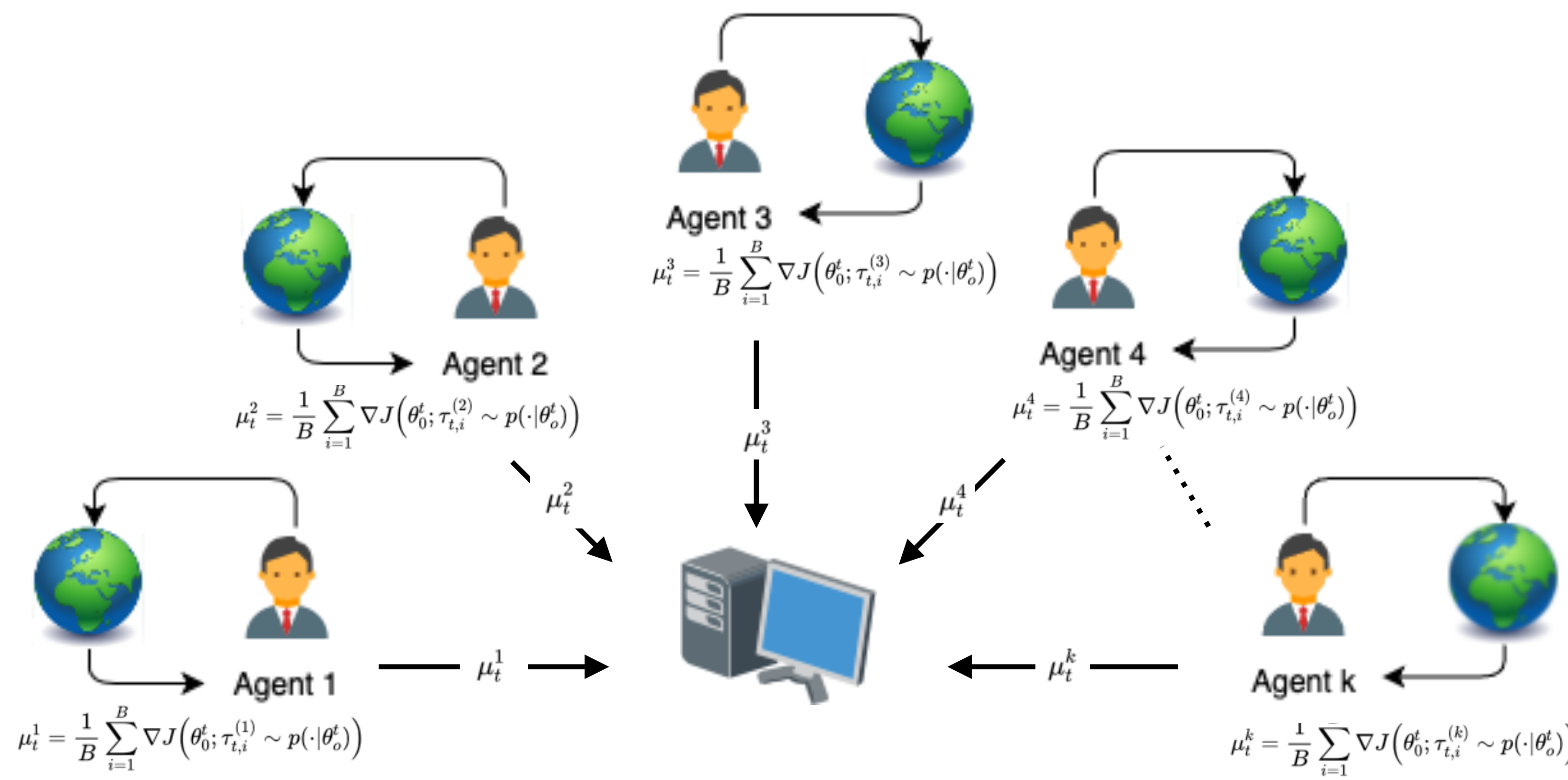
$$\theta^{t+1} \leftarrow \theta^t + \eta_t \mu_t$$

Federated Policy Gradient using SCSG Optimization

Stochastically Controlled

Stochastic Gradient (SCSG)

- Variance-Reduced Optimization
- refined control over the variance



$$\mu_t \leftarrow \{\mu_t^1, \mu_t^2, \mu_t^3, \mu_t^4, \dots, \mu_t^k\}$$

for $n = 0$ **to** $N_t - 1$ **do**

$$v_n^t \triangleq \frac{1}{b_t}\sum_{j=1}^{b_t}\left[\nabla J(\boldsymbol{\theta}_n^t; \tau_{n,j}^t) - \nabla J(\boldsymbol{\theta}_0^t; \tau_{n,j}^t)\right]_{\tau_{n,j}^t \sim p(\cdot|\boldsymbol{\theta}_n^t)} + \mu_t$$

$$\boldsymbol{\theta}_{n+1}^t = \boldsymbol{\theta}_n^t + \eta_t v_n^t$$

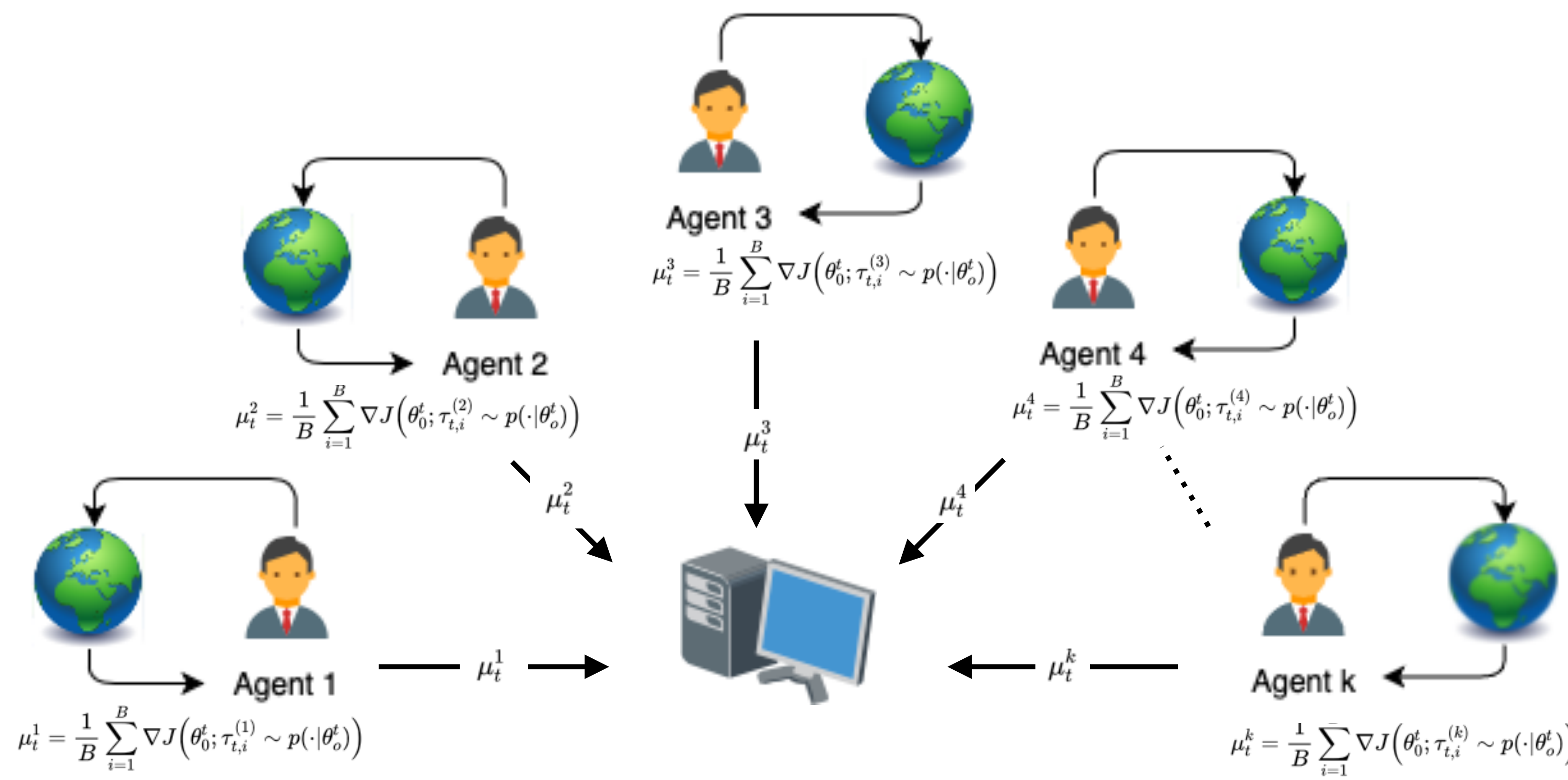
SCSG

Federated Policy Gradient using SCSG Optimization

Stochastically Controlled

Stochastic Gradient (SCSG)

- Variance-Reduced Optimization
- refined control over the variance



$$\mu_t \leftarrow \{\mu_t^1, \mu_t^2, \mu_t^3, \mu_t^4, \ldots, \mu_t^k\}$$

for $n = 0$ **to** $N_t - 1$ **do**

importance weight sampling

$$v_n^t \triangleq \frac{1}{b_t} \sum_{j=1}^{b_t} \left[\nabla J(\boldsymbol{\theta}_n^t; \tau_{n,j}^t) - \omega(\boldsymbol{\theta}_n^t, \boldsymbol{\theta}_0^t; \tau_{n,j}^t) \nabla J(\boldsymbol{\theta}_0^t; \tau_{n,j}^t)\right]_{\tau_{n,j}^t \sim p(\cdot|\boldsymbol{\theta}_n^t)} + \mu_t$$

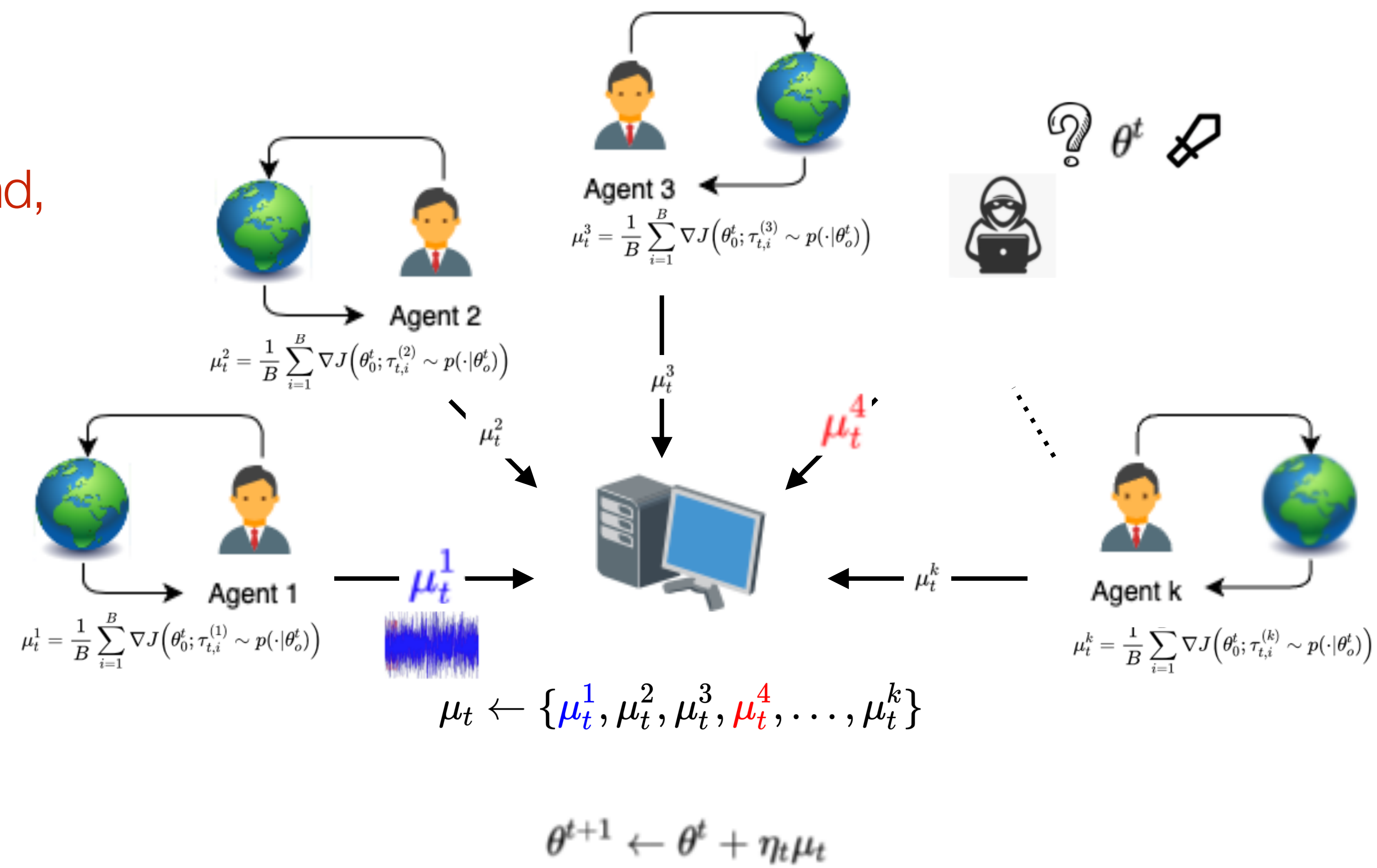
$$\boldsymbol{\theta}_{n+1}^t = \boldsymbol{\theta}_n^t + \eta_t v_n^t$$

SCSG

Handling Byzantine Agents

A gradient-based filter that in each round,

- removes gradients that the server believes are from Byzantine agents



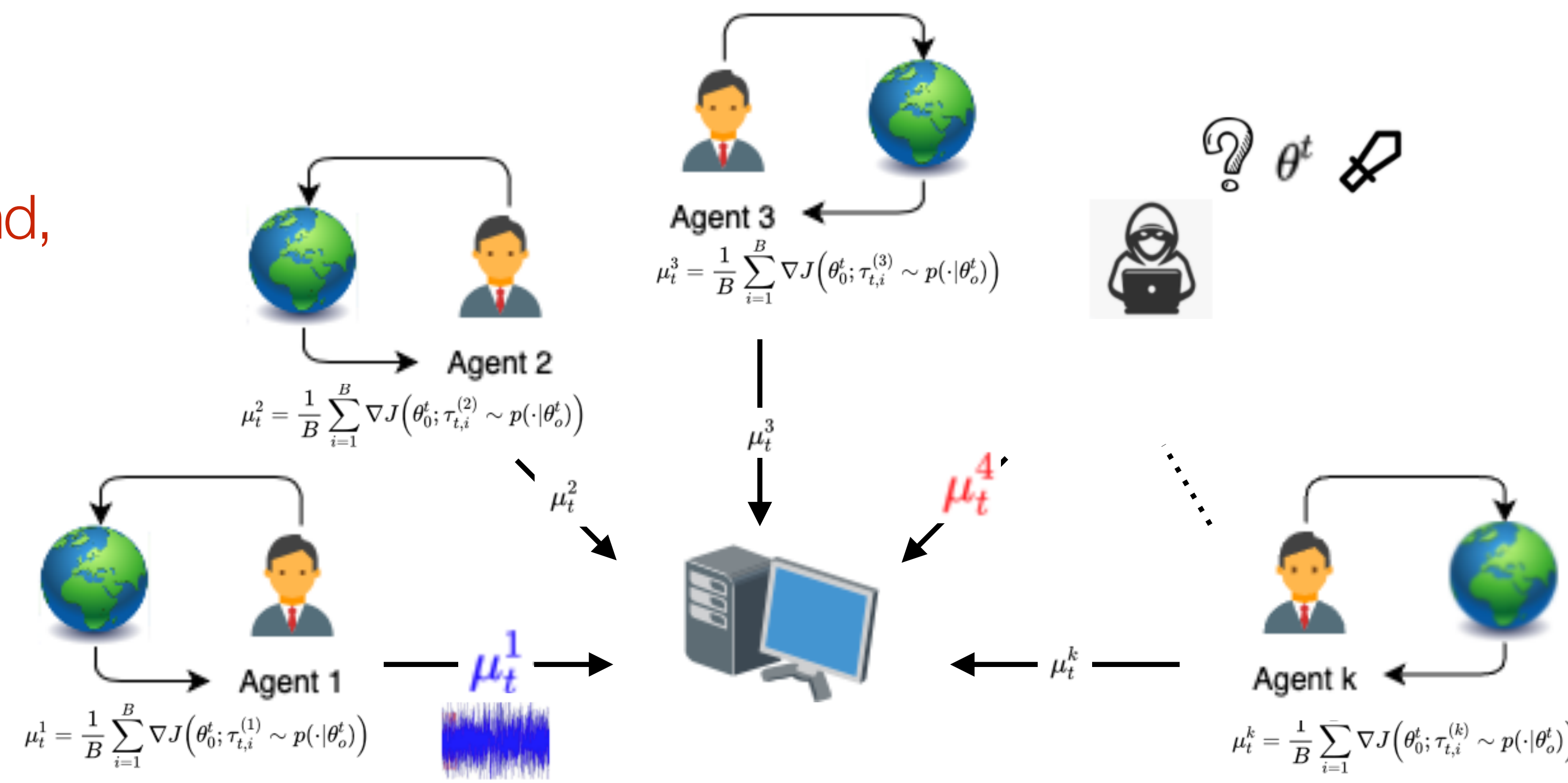
Handling Byzantine Agents

A gradient-based filter that in each round,

- removes gradients that the server believes are from Byzantine agents

Assumptions

- $\alpha < 0.5$
- bounded variance (Assumption 2)



$$\mu_t \leftarrow \{\mu_t^1, \mu_t^2, \mu_t^3, \mu_t^4, \ldots, \mu_t^k\}$$

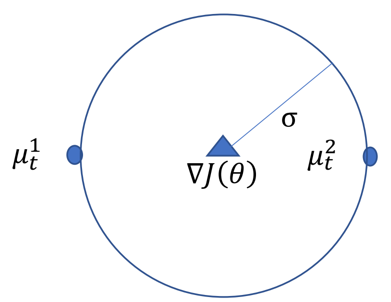
$$\theta^{t+1} \leftarrow \theta^t + \eta_t \mu_t$$

Assumption 2 (On bounded variance of the gradient estimator). *There is a constant σ such that $\|g(\tau|\boldsymbol{\theta}) - \nabla J(\boldsymbol{\theta})\| \leq \sigma$ for any $\tau \sim p(\tau|\boldsymbol{\theta})$ for all policy $\pi_{\boldsymbol{\theta}}$.*

Assumption 2 (On variance of the gradient estimator)

There is a constant σ such that $\|g(\tau|\boldsymbol{\theta}) - \nabla J(\boldsymbol{\theta})\| \leq \sigma$ for any $\tau \sim p(\tau|\boldsymbol{\theta})$ for all policy $\pi_{\boldsymbol{\theta}}$.

$$\|\mu_t^k - \nabla J(\theta_t)\| \leq \sigma, \forall k$$



$$\|\mu_t^1 - \mu_t^2\| \leq 2\sigma, \forall k_1, k_2 \in \mathcal{G}$$

$$\alpha < 0.5$$

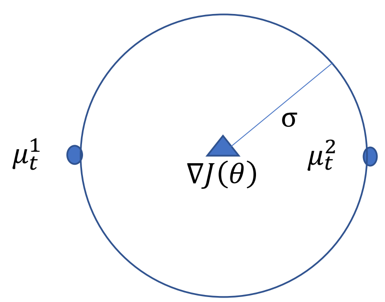
$$S \triangleq \left\{\mu_t^{(k)}\right\} \text{ where } k \in [K]$$

$$\text{s.t. } \left|\left\{k' \in [K] : \left\|\mu_t^{(k')} - \mu_t^{(k)}\right\| \leq 2\sigma\right\}\right| > \frac{K}{2}$$

Assumption 2 (On variance of the gradient estimator)

There is a constant σ such that $\|g(\tau|\boldsymbol{\theta}) - \nabla J(\boldsymbol{\theta})\| \leq \sigma$ for any $\tau \sim p(\tau|\boldsymbol{\theta})$ for all policy $\pi_{\boldsymbol{\theta}}$.

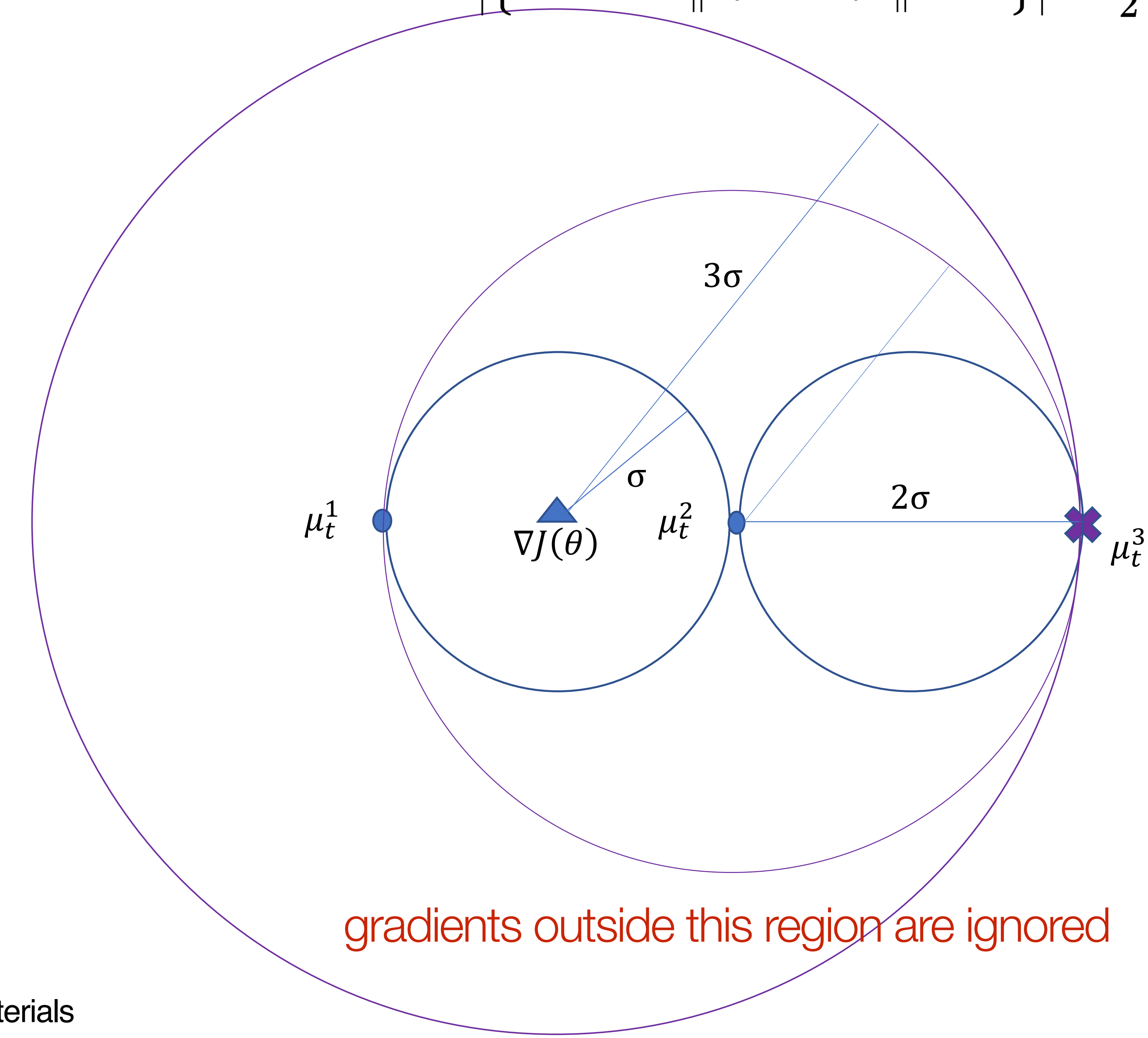
$$\|\mu_t^k - \nabla J(\theta_t)\| \leq \sigma, \forall k$$



$$\|\mu_t^1 - \mu_t^2\| \leq 2\sigma, \forall k_1, k_2 \in \mathcal{G}$$

$$S \triangleq \left\{\mu_t^{(k)}\right\} \text{ where } k \in [K]$$

$$\text{s.t.} \left|\left\{k' \in [K] : \left\|\mu_t^{(k')} - \mu_t^{(k)}\right\| \leq 2\sigma\right\}\right| > \frac{K}{2}$$



For details and proofs, please refer to our supplementary materials

Outline

- Motivation & Background
- Problem Setup
- FedPG-BR
- Theoretical Results
- Experiments

Convergence Guarantee

$$\mathbb{E}[\|\nabla J(\tilde{\boldsymbol{\theta}}_a)\|^2] \leq \frac{2\Psi\left[J(\tilde{\boldsymbol{\theta}}^*) - J(\tilde{\boldsymbol{\theta}}_0)\right]}{TB^{1/3}} + \frac{8\sigma^2}{(1-\alpha)^2 KB} + \frac{96\alpha^2\sigma^2 V}{(1-\alpha)^2 B}$$

Convergence Guarantee

$$\mathbb{E}[\|\nabla J(\tilde{\boldsymbol{\theta}}_a)\|^2] \leq \frac{2\Psi\left[J(\tilde{\boldsymbol{\theta}}^*) - J(\tilde{\boldsymbol{\theta}}_0)\right]}{TB^{1/3}} + \boxed{\frac{8\sigma^2}{(1-\alpha)^2 KB} + \frac{96\alpha^2\sigma^2 V}{(1-\alpha)^2 B}}$$

$$\Big\downarrow K = 1, \alpha = 0$$

$$\boxed{\frac{8\sigma^2}{B}}$$

$$\downarrow$$

full gradient approximation

Convergence Guarantee

$$\mathbb{E}[\|\nabla J(\tilde{\boldsymbol{\theta}}_a)\|^2] \leq \boxed{\frac{2\Psi\left[J(\tilde{\boldsymbol{\theta}}^*) - J(\tilde{\boldsymbol{\theta}}_0)\right]}{TB^{1/3}}} + \boxed{\frac{8\sigma^2}{(1-\alpha)^2 KB} + \frac{96\alpha^2\sigma^2 V}{(1-\alpha)^2 B}}$$

SCSG for non-convex optimization[2]

$$\xrightarrow{K = 1,\ \alpha = 0} \quad \boxed{\frac{8\sigma^2}{B}}$$

full gradient approximation

[2]Lei, Lihua, Cheng Ju, Jianbo Chen, and Michael I. Jordan. NeurIPS2017

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
$K = 1$	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3} K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3} K^{2/3}} + \frac{\alpha^{4/3}}{\epsilon^{5/3}})$

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
$K = 1$	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}} + \frac{\alpha^{4/3}}{\epsilon^{5/3}})$

conventional PG methods using stochastic gradient-based optimization

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
$K = 1$	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}} + \frac{\alpha^{4/3}}{\epsilon^{5/3}})$

recent endeavours in adapting SVRG to PG

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
$K = 1$	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}} + \frac{\alpha^{4/3}}{\epsilon^{5/3}})$

this work
adapting SCSG to PG

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
$K = 1$	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}} + \frac{\alpha^{4/3}}{\epsilon^{5/3}})$

guaranteed to improve as K increases

Sample Complexities

SETTINGS	METHODS	COMPLEXITY
	REINFORCE [39]	$O(1/\epsilon^2)$
	GPOMDP [40]	$O(1/\epsilon^2)$
$K = 1$	SVRPG [18]	$O(1/\epsilon^2)$
	SVRPG [19]	$O(1/\epsilon^{5/3})$
	FedPG-BR	$O(1/\epsilon^{5/3})$
$K > 1, \alpha = 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}})$
$K > 1, \alpha > 0$	**FedPG-BR**	$O(\frac{1}{\epsilon^{5/3}K^{2/3}} + \boxed{\frac{\alpha^{4/3}}{\epsilon^{5/3}}})$

guaranteed to improve as K increases,

in the potential presence of Byzantine agents

additive to the convergence

Outline

- Motivation & Background
- Problem Setup
- FedPG-BR
- Theoretical Results
- Experiments

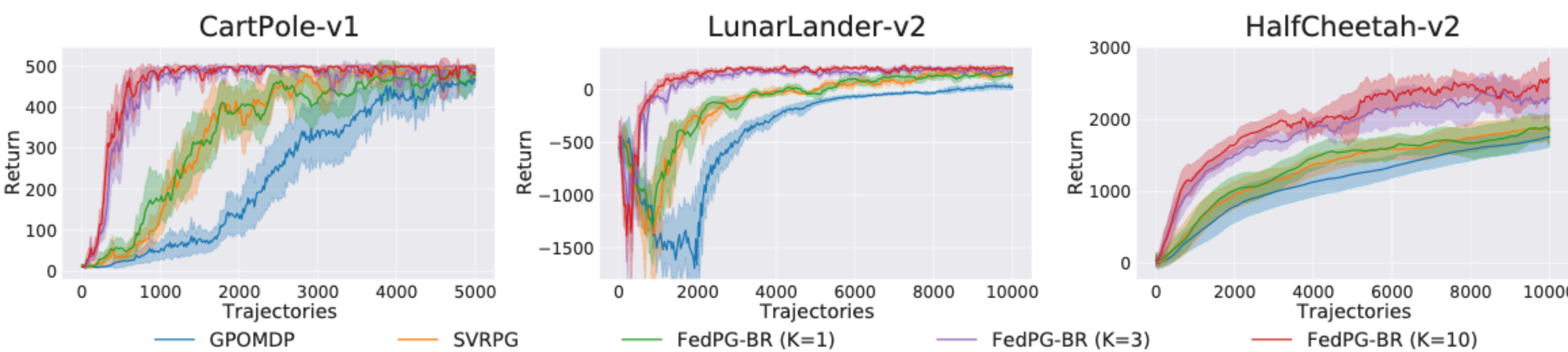
Performance in Ideal Systems with $\alpha = 0$

FedPG-BR and SVRPG performs comparably when $K = 1$ (single-agent)

- both outperforming GPOMDP

The performance of FedPG-BR…

- … is improved significantly with the federation of only $K = 3$ agents
- … is improved even further with $K = 10$ agents



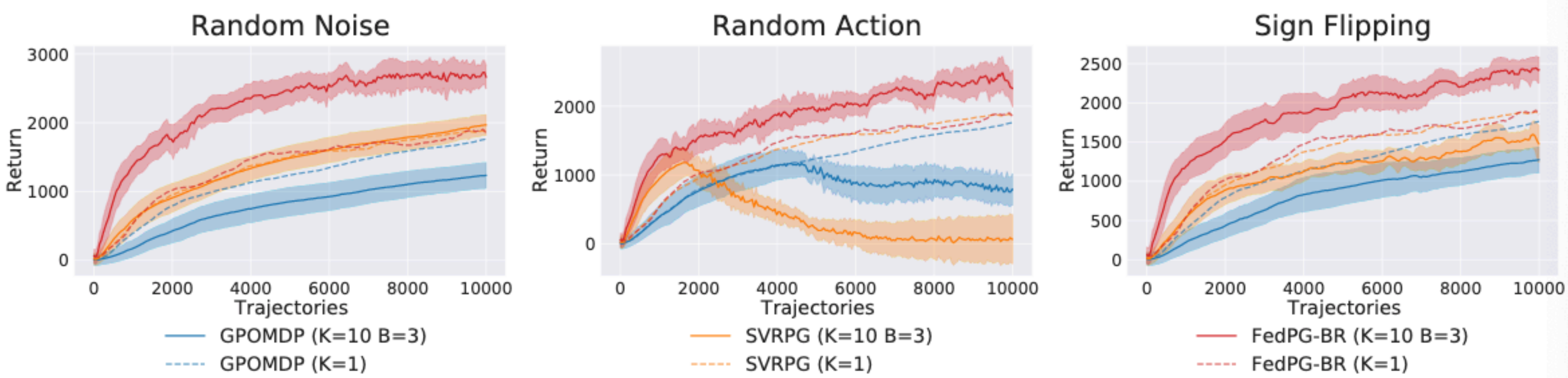
Flint Xiaofeng Fan
National University of Singapore (NUS) & Agency for Science, Technology and Research (A*STAR)

Performance in Practical Systems with $\alpha > 0$

K = 10 agents among which 3 are Byzantine agents being either

- Random Noise. Each Byzantine agent sends a random vector to the server
- Random Action. Every Byzantine agent ignores the policy from the server and takes actions randomly
- Sign Flipping. Each Byzantine agent computes the correct gradient but sends the scaled negative gradient

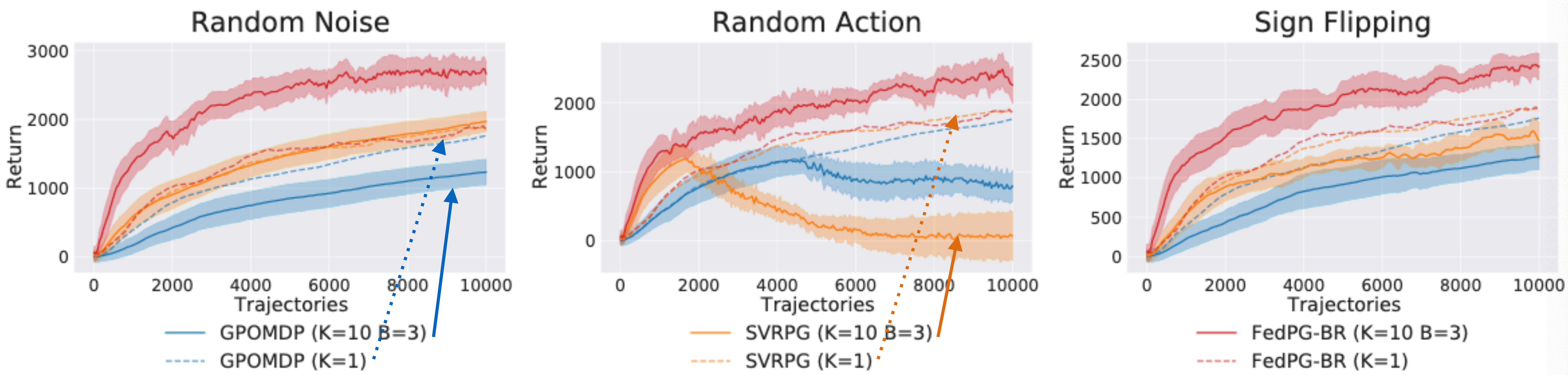
Performance in Practical Systems with $\alpha > 0$



Performance in Practical Systems with $\alpha > 0$

With the presence of Byzantine agents, the performance of federation of…

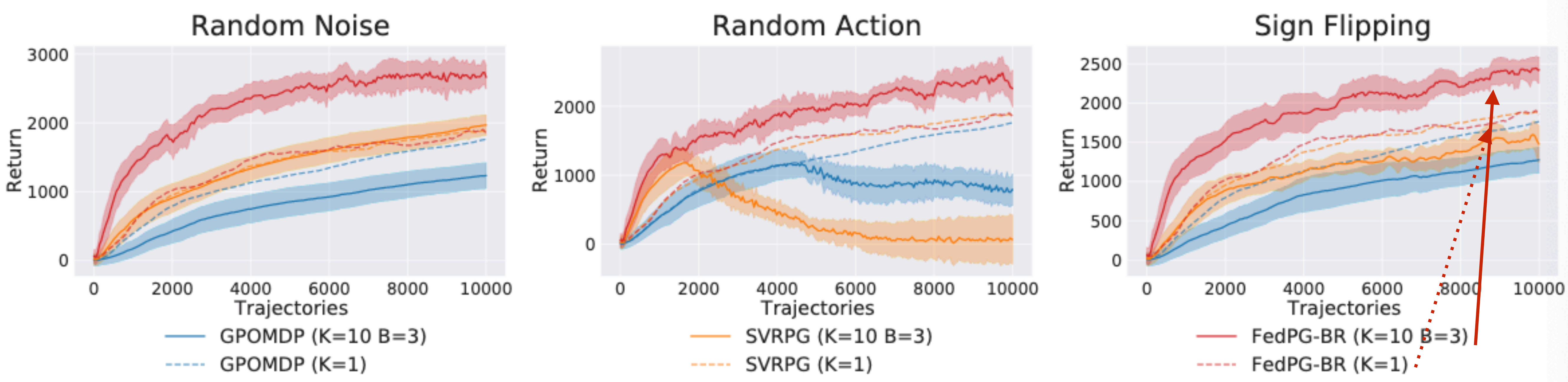
- … GPOMDP and SVRPG are worse than that in the single-agent setup



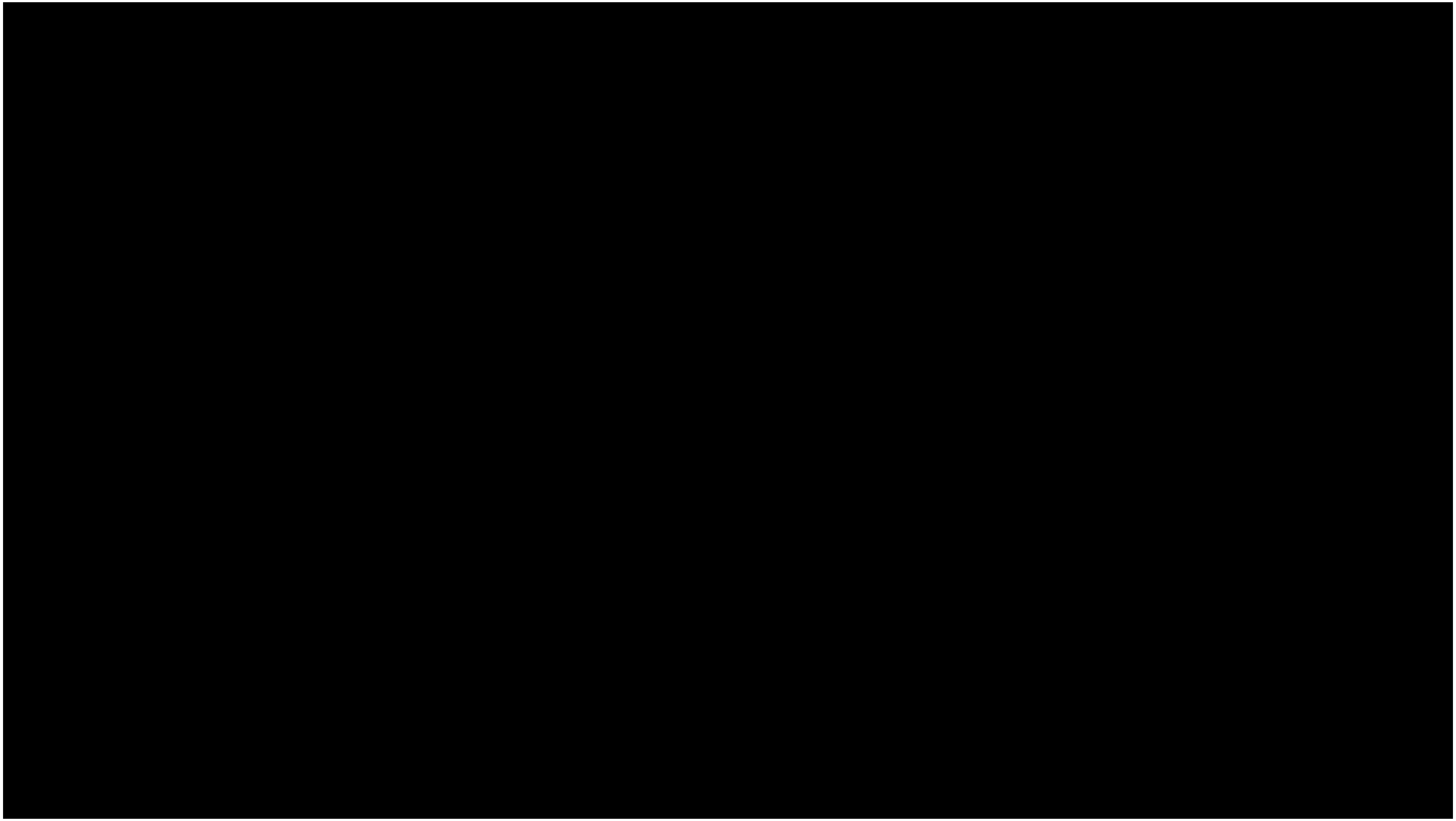
Performance in Practical Systems with $\alpha > 0$

With the presence of Byzantine agents, the performance of federation of…

- … GPOMDP and SVRPG are worse than that in the single-agent setup
- … FedPG-BR(K=10 B=3) is robust against all 3 types of Byzantine agents
  - significantly outperforms its single-agent setup



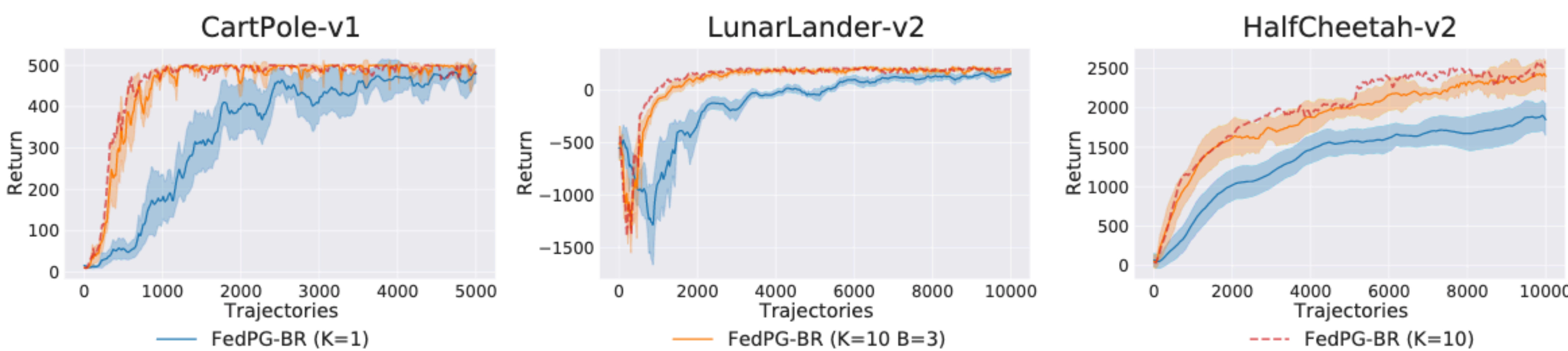
Performance in Practical Systems with $\alpha > 0$



Fault-Tolerant Federated Reinforcement Learning with Theoretical Guarantee

Thank you!

Performance of FedPG-BR against more sophisticated attacks



Performance of FedPG-BR against more sophisticated attacks