

Decentralized Federated Policy Gradient with Byzantine Fault-Tolerance and Provably Fast Convergence

Philip Jordan¹ Florian Grötschla¹ Flint Xiaofeng Fan² Roger Wattenhofer¹

¹ ETH Zürich

² National University of Singapore

Decentralized Federated Policy Gradient with Byzantine Fault-Tolerance and Provably Fast Convergence

Philip Jordan¹ Florian Grötschla¹ Flint Xiaofeng Fan² Roger Wattenhofer¹

¹ ETH Zürich

² National University of Singapore

Outline

Outline

1) Motivation

Outline

- 1) Motivation
- 2) Setting & Related Work

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations

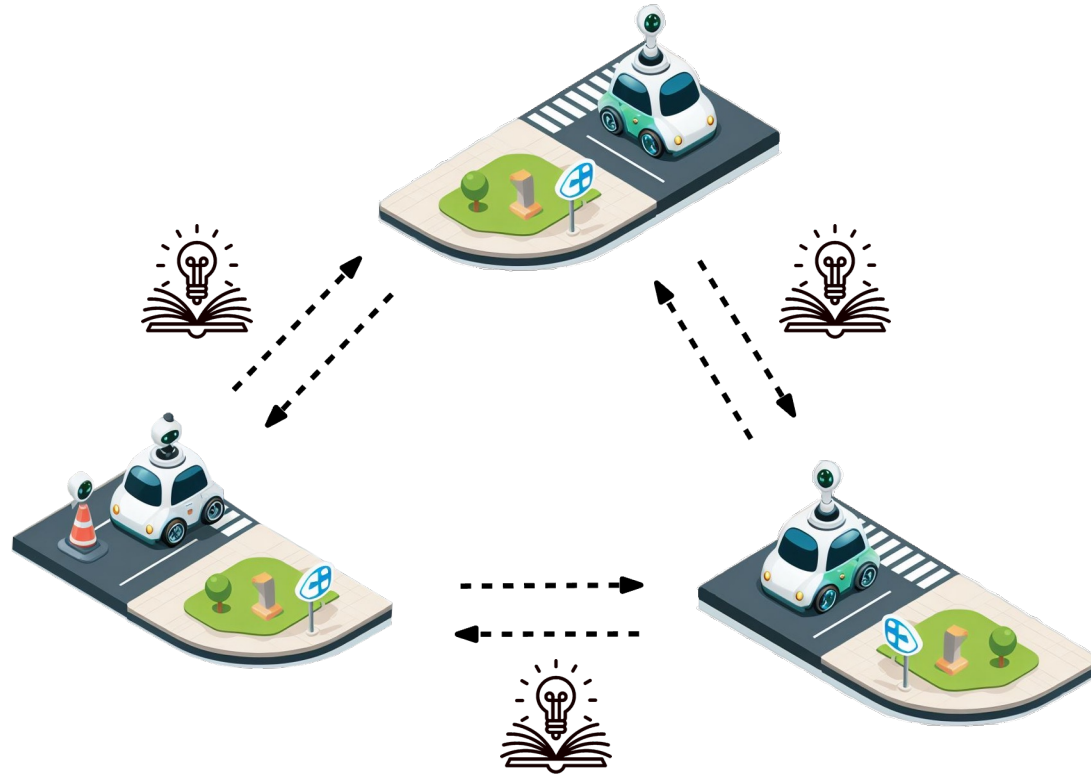
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

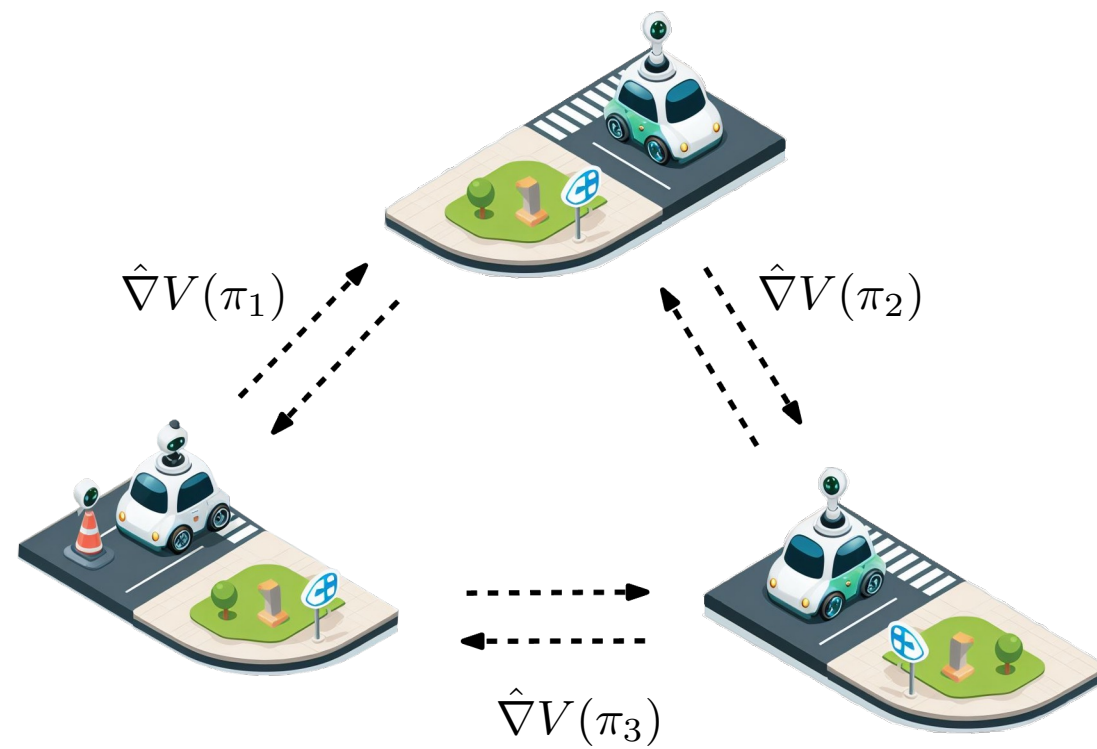
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

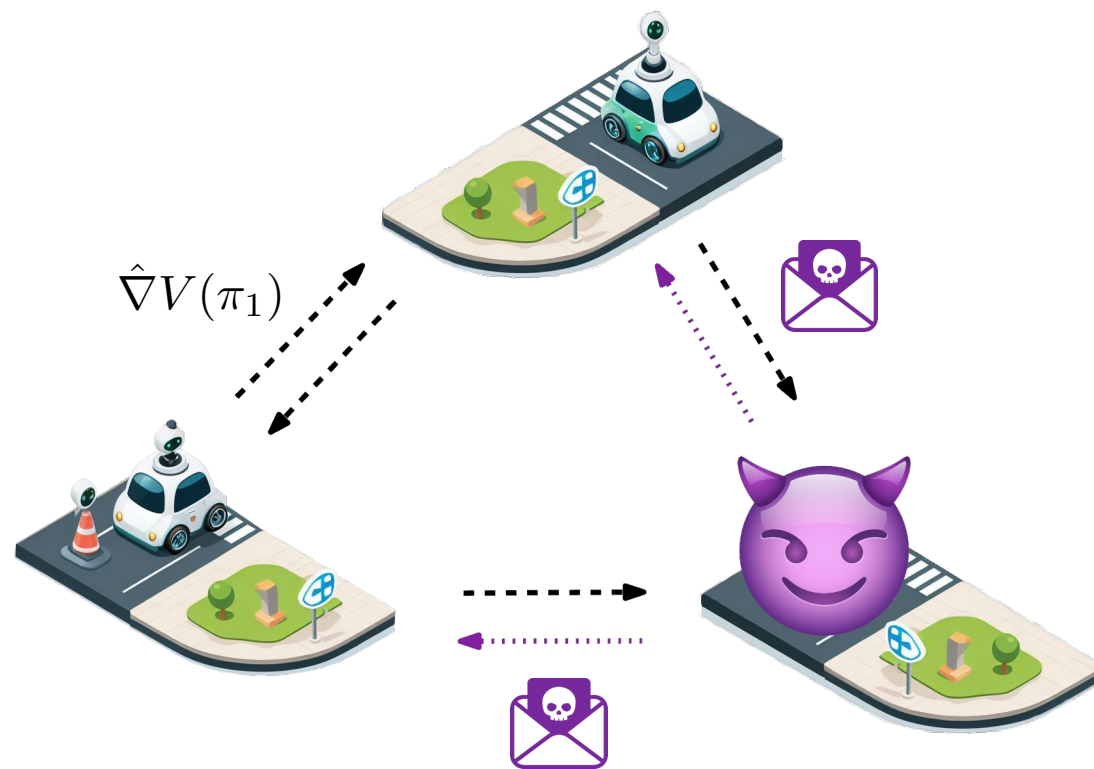
Motivation



Motivation



Motivation



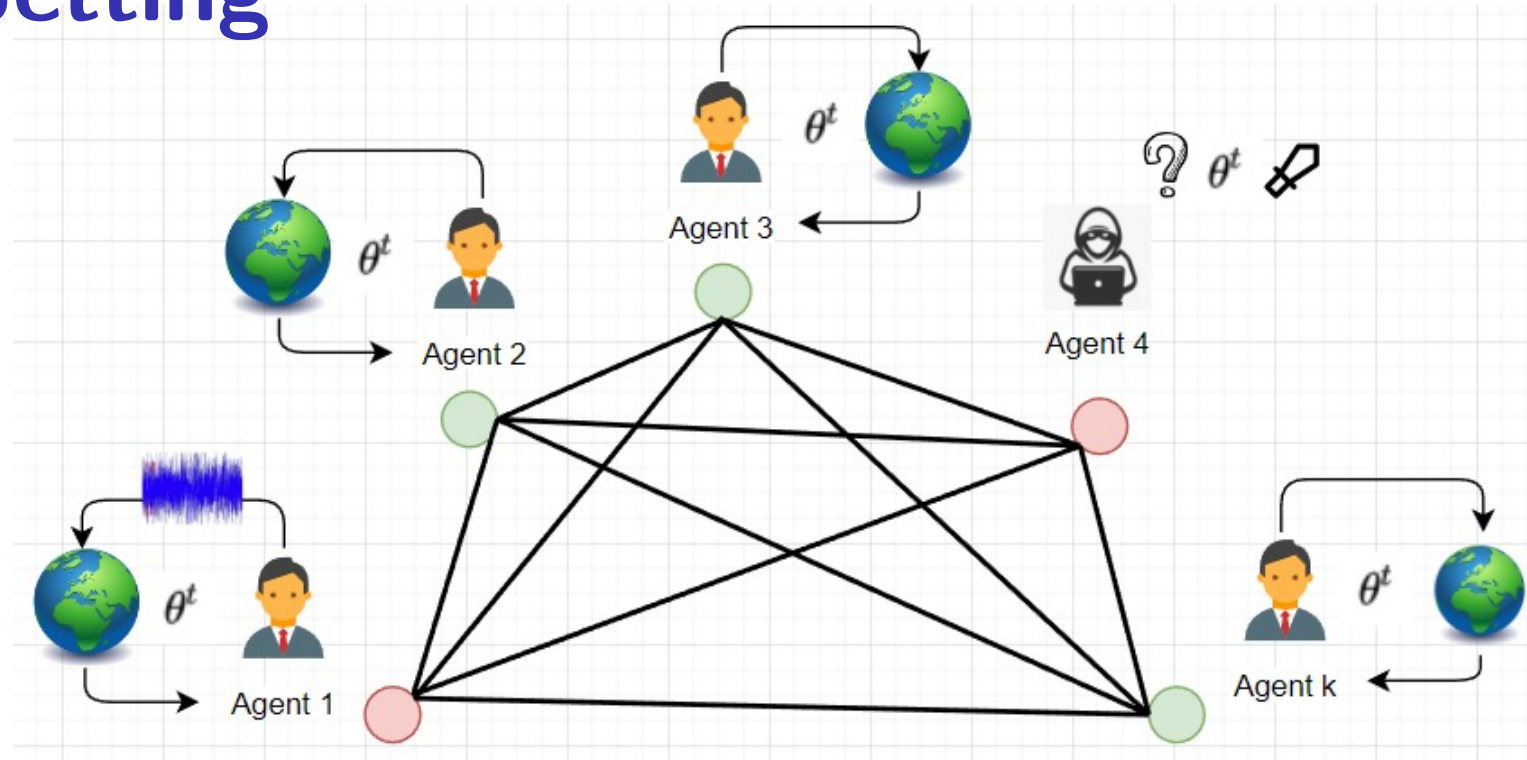
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Outline

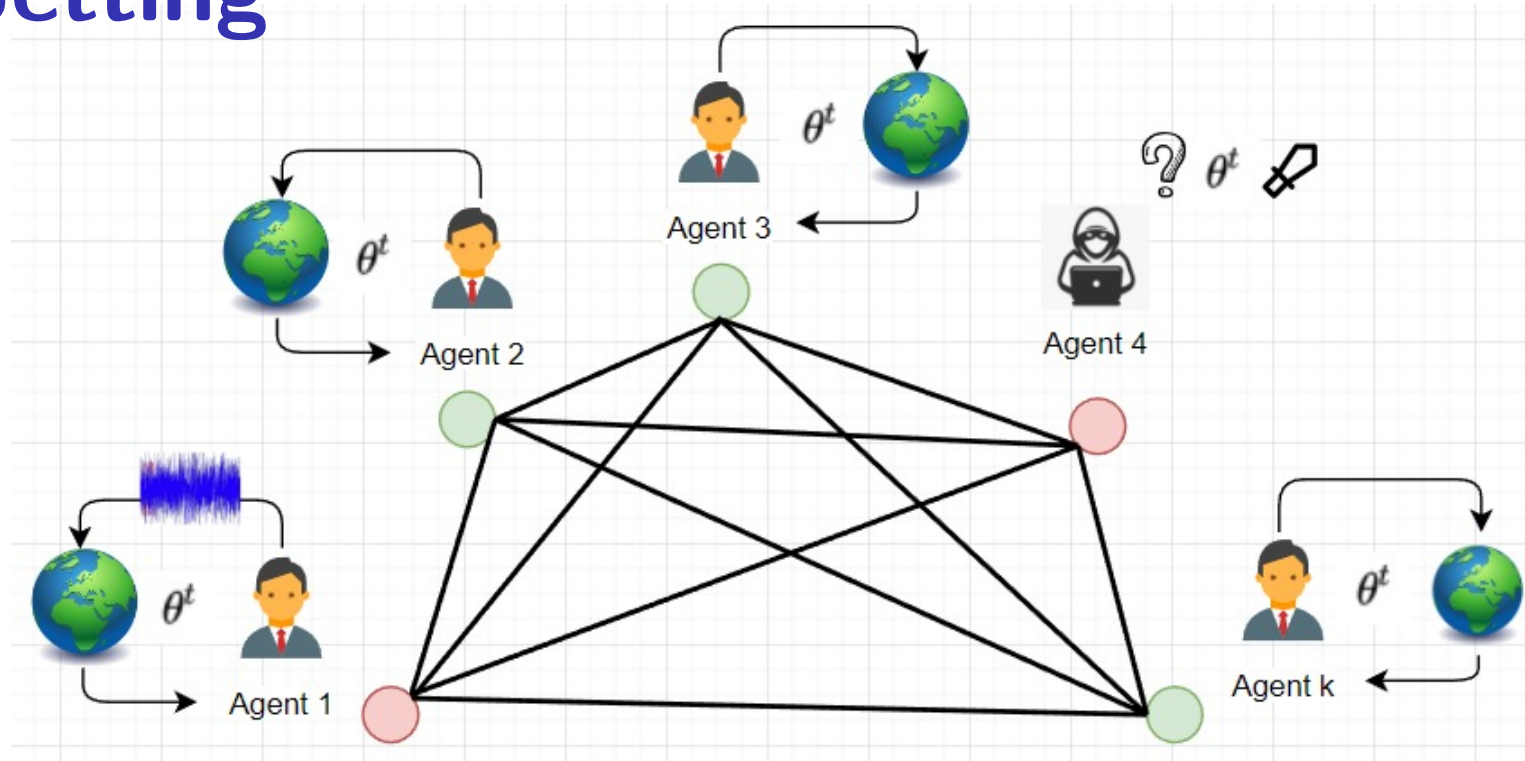
- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Problem Setting



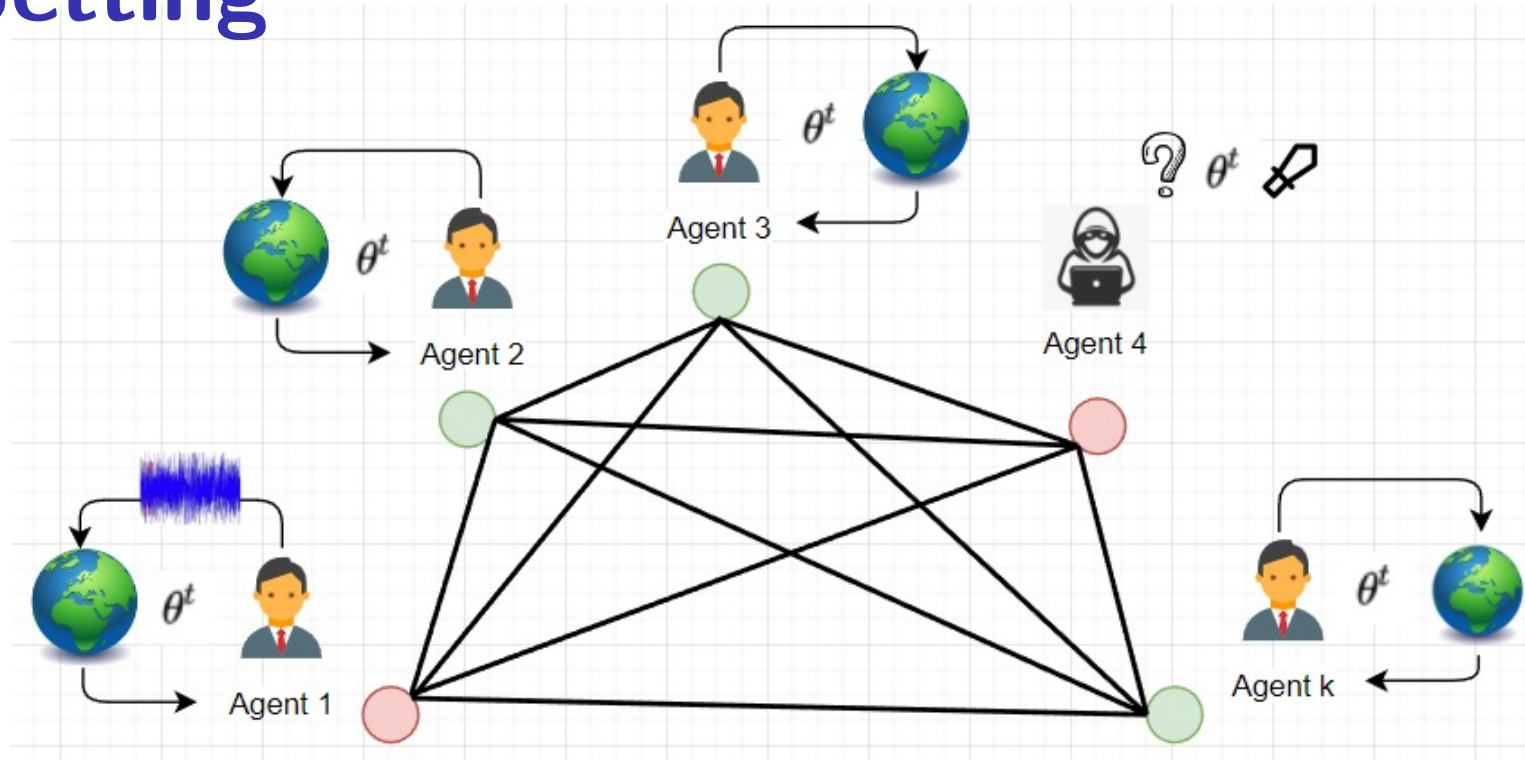
➤ K agents, each...

Problem Setting



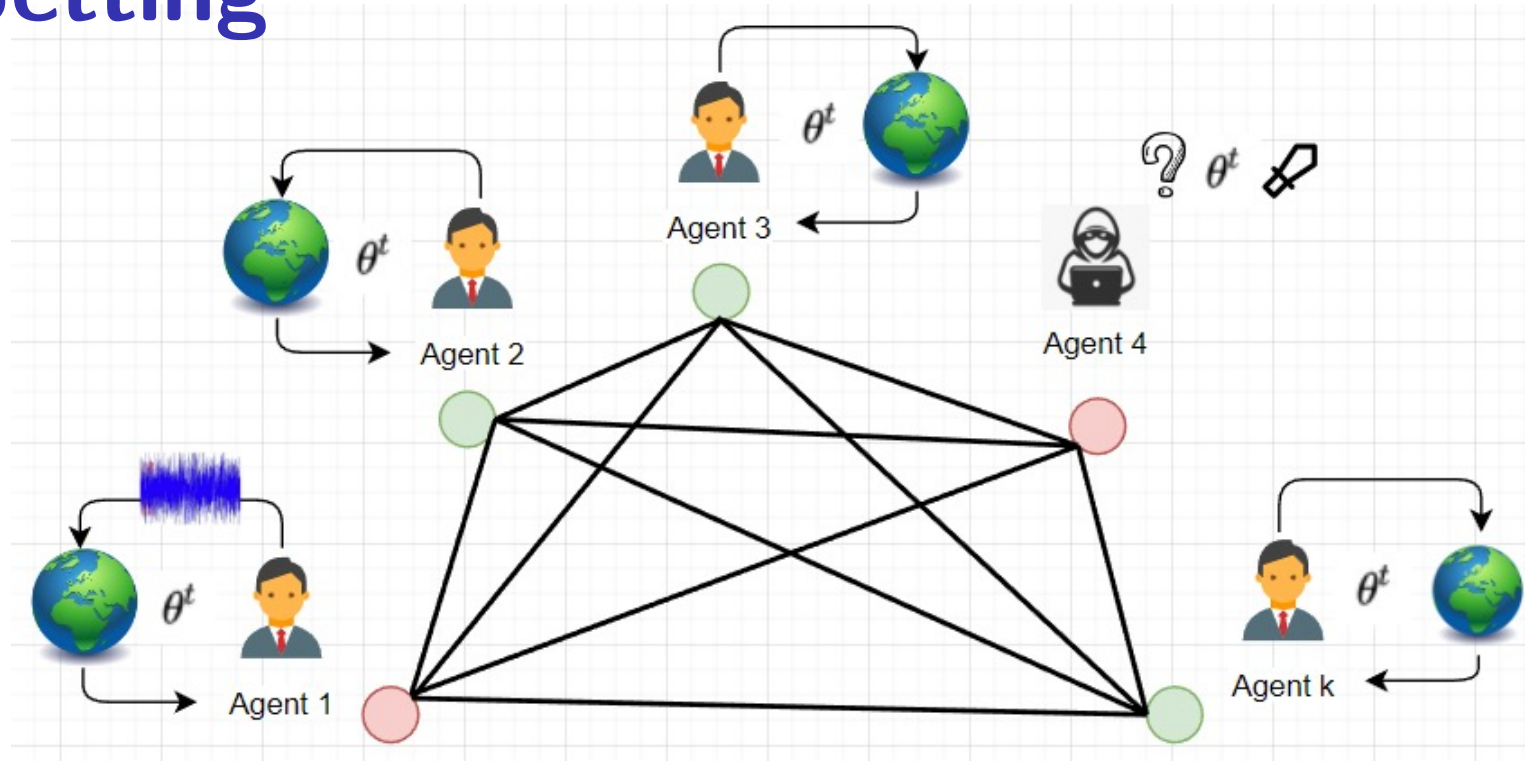
- K agents, each...
 - ...interacts with MDP $\mathcal{M} := \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho\} \rightarrow$ homogeneous local environments

Problem Setting



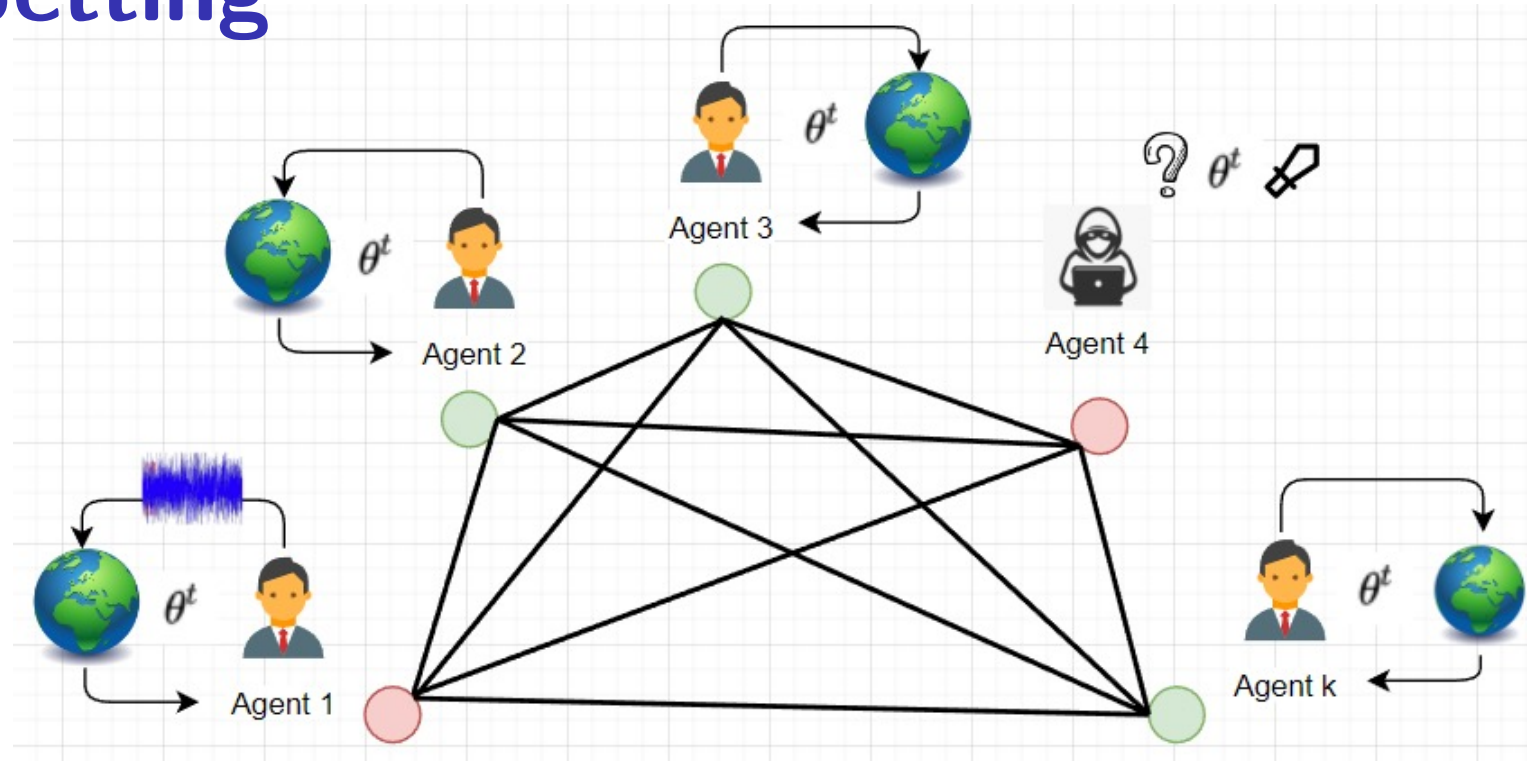
- K agents, each...
 - ...interacts with MDP $\mathcal{M} := \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho\} \rightarrow$ homogeneous local environments
 - ...maintains $\pi_{\theta^{(i)}}$ with local parameters $\theta^{(i)} \in \mathbb{R}^d$

Problem Setting



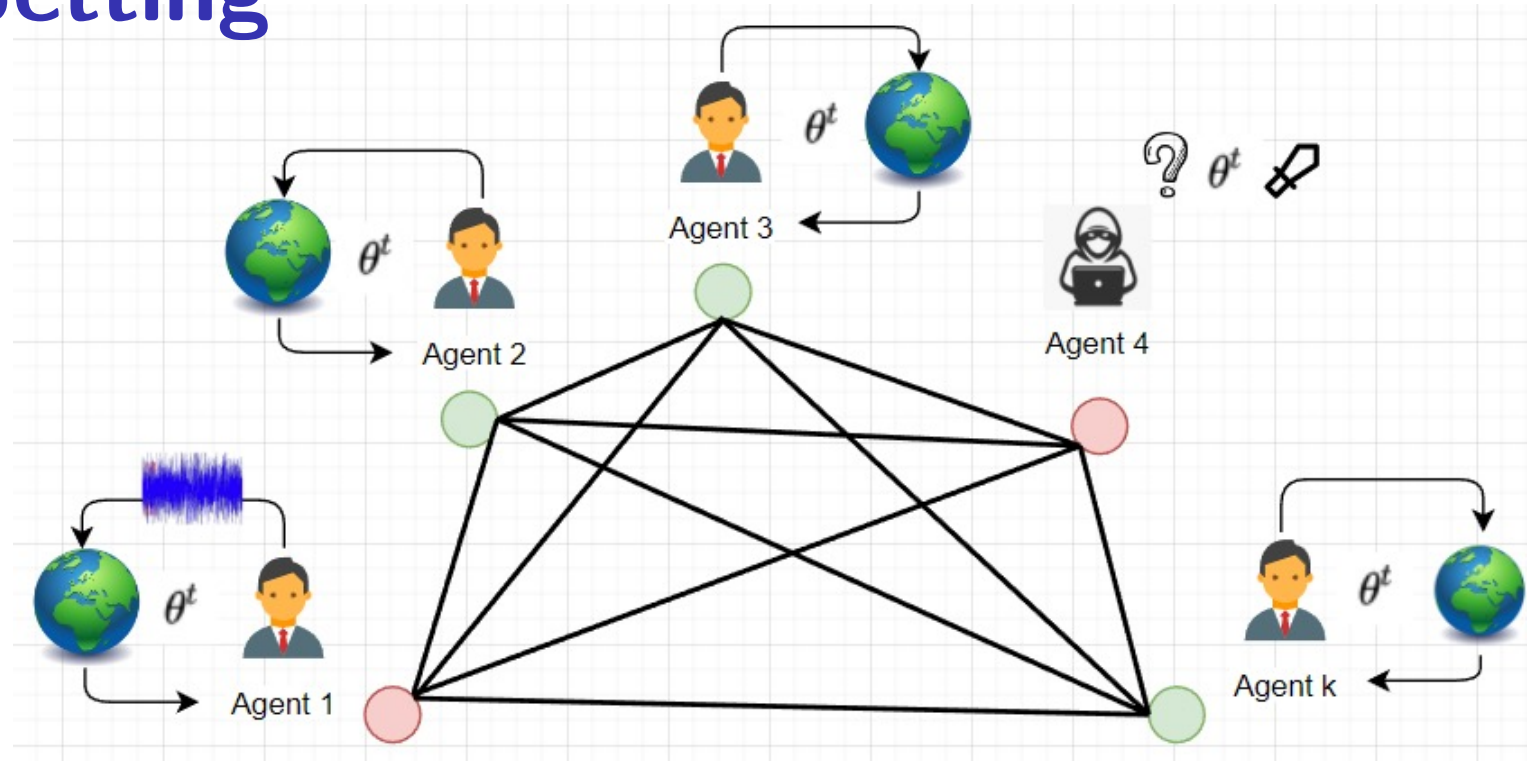
- K agents, each...
 - ...interacts with MDP $\mathcal{M} := \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho\} \rightarrow$ homogeneous local environments
 - ...maintains $\pi_{\theta^{(i)}}$ with local parameters $\theta^{(i)} \in \mathbb{R}^d$
 - ... aims to find $\arg \max_{\theta^{(i)} \in \mathbb{R}^d} J(\theta^{(i)}) := \mathbb{E}_{\tau \sim p(\cdot | \theta^{(i)})} \left[\sum_{t=0}^{H-1} \gamma^t \mathcal{R}(s_t, a_t) \mid \mathcal{M} \right]$

Problem Setting



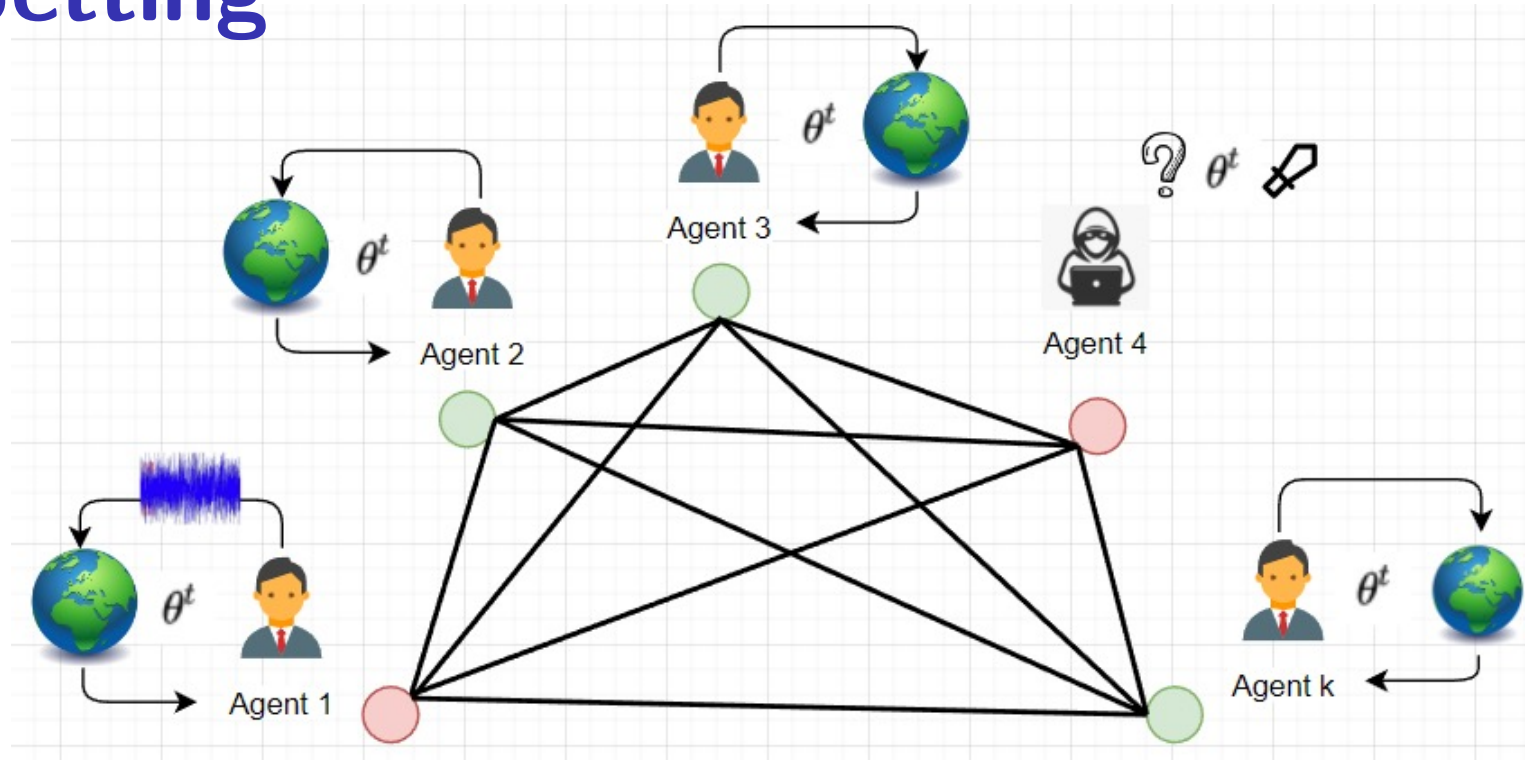
➤ communication:

Problem Setting



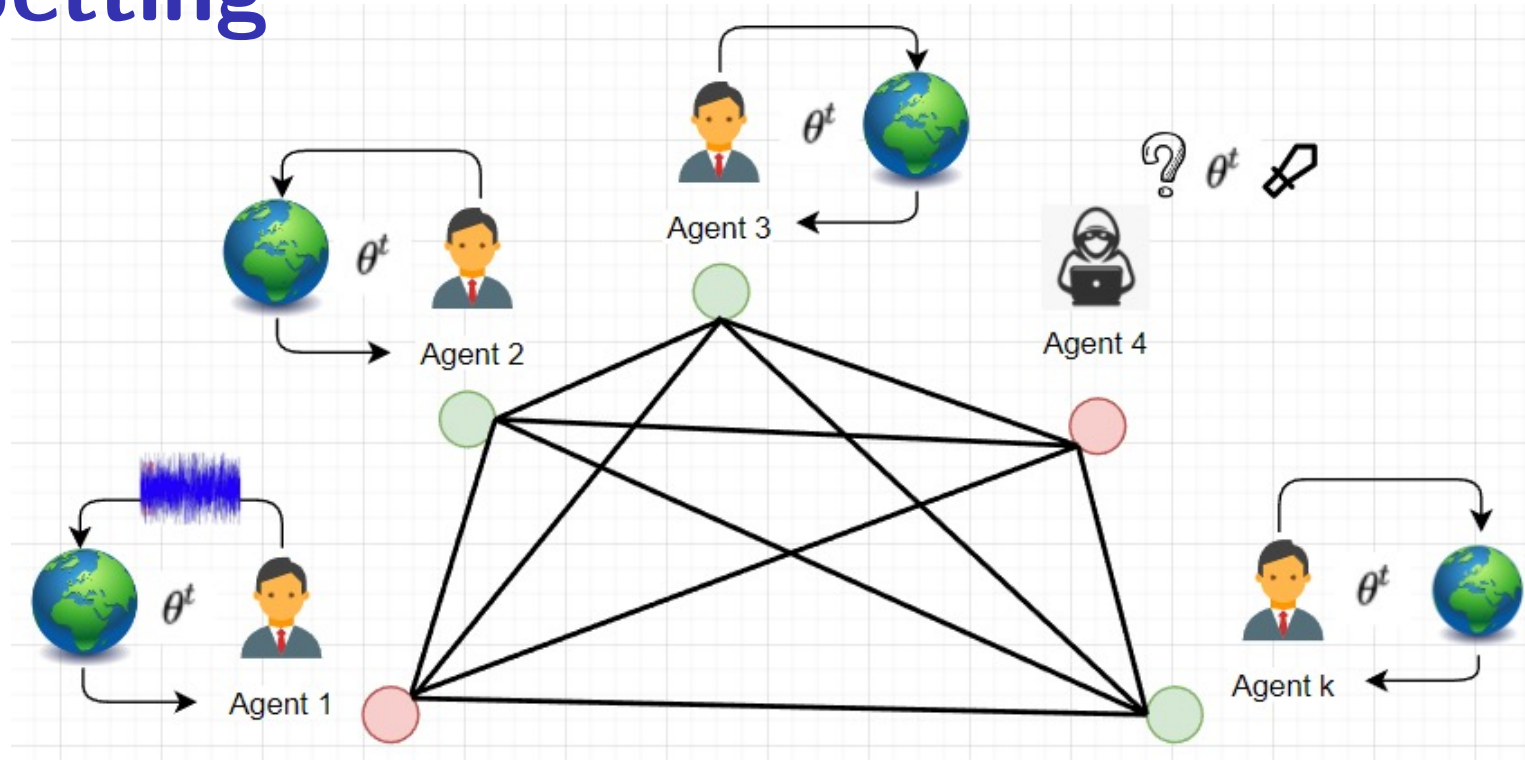
- communication:
 - synchronous, round-based

Problem Setting



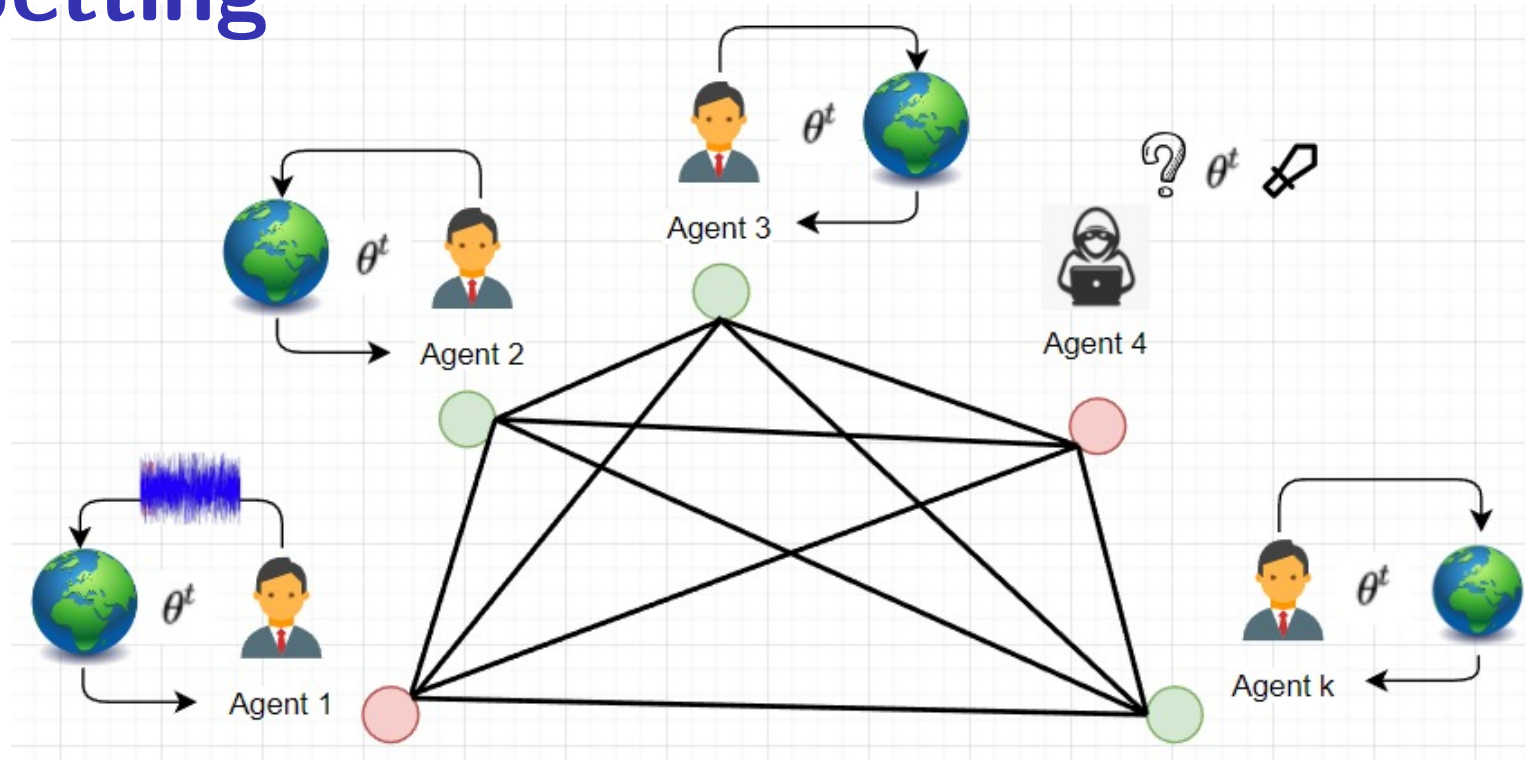
- communication:
 - synchronous, round-based
 - all-to-all, i.e., $\mathcal{O}(K^2)$ messages per round

Problem Setting



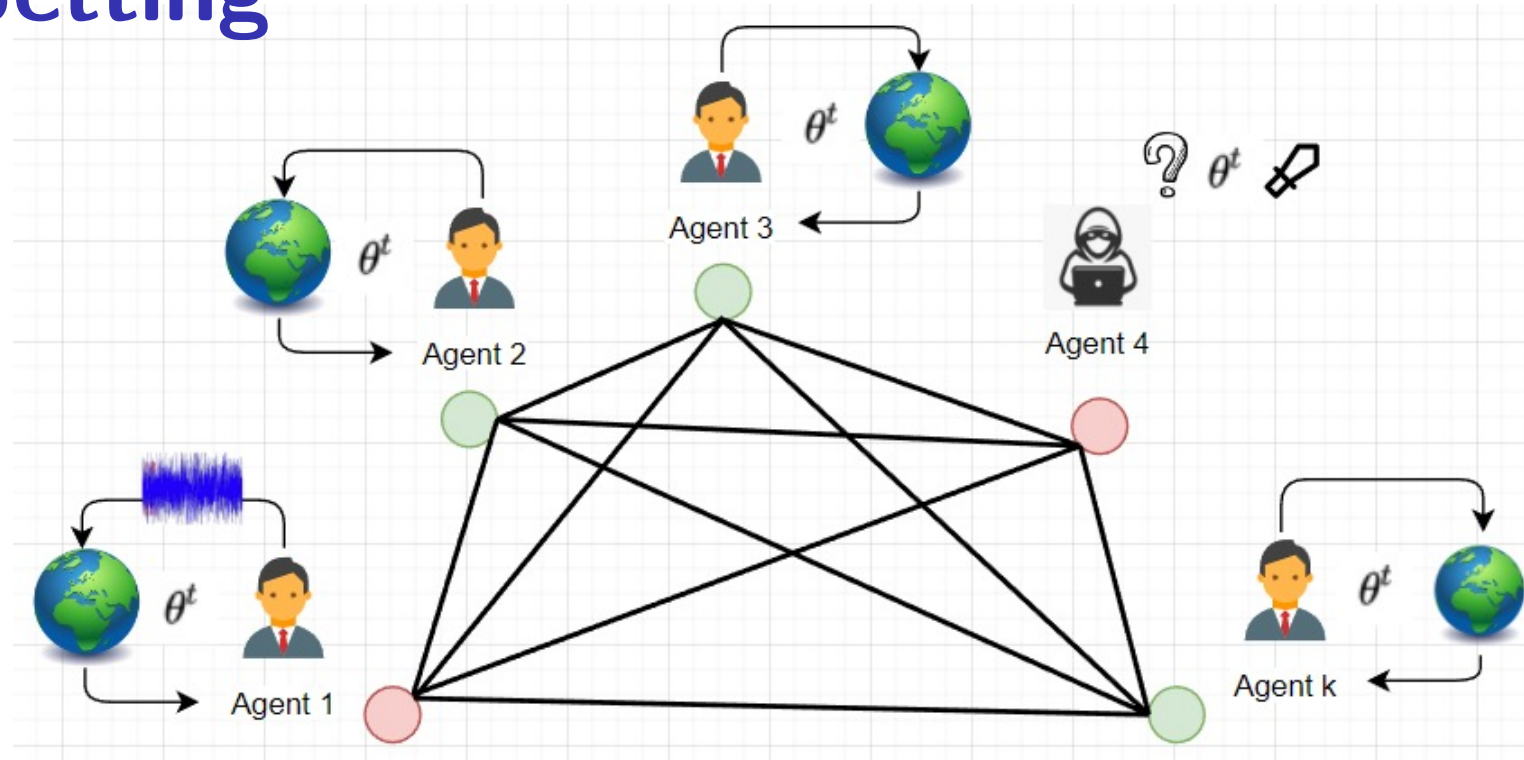
- up to α -fraction of agents may be Byzantine where $\alpha < 1/4$

Problem Setting



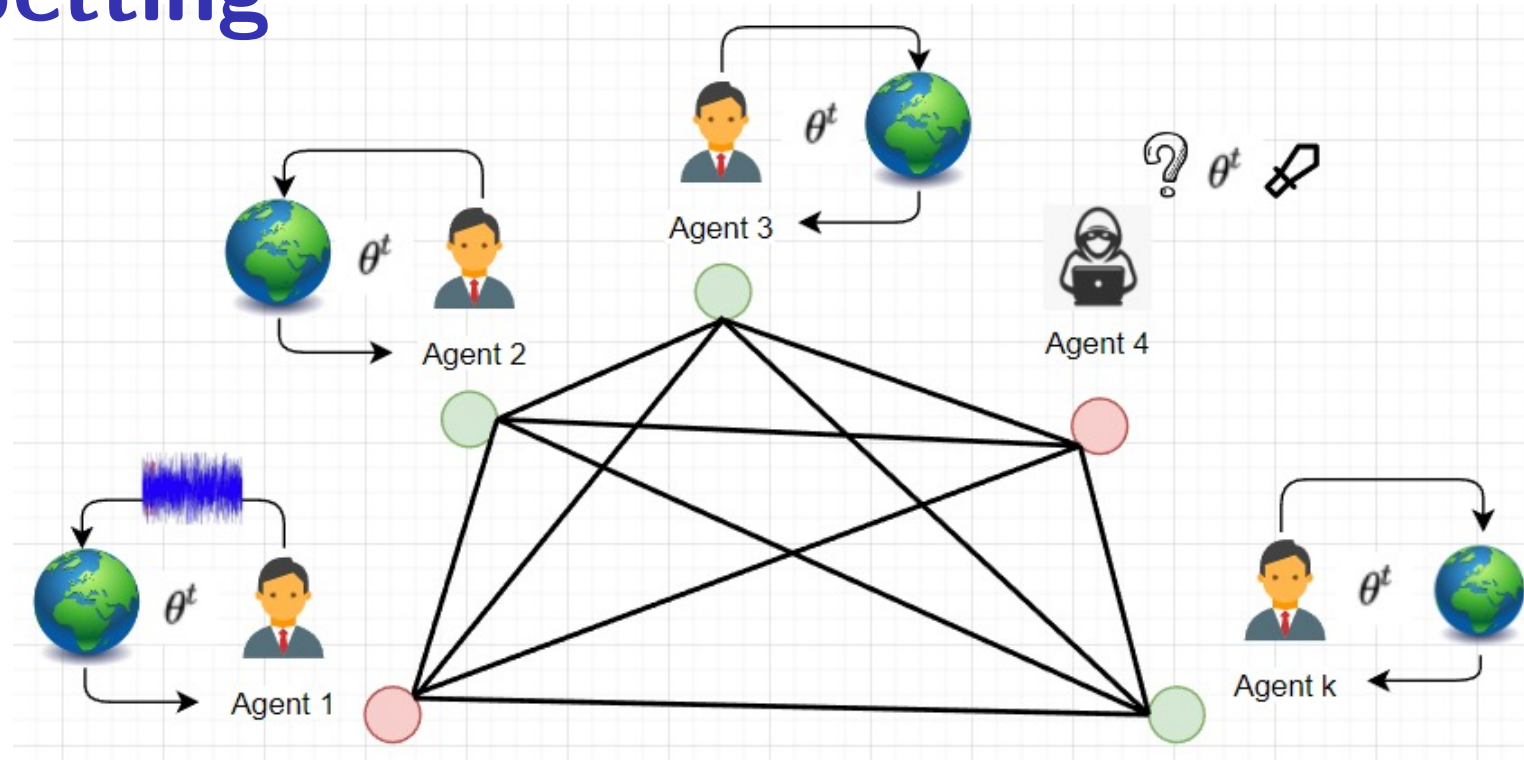
- up to α -fraction of agents may be Byzantine where $\alpha < 1/4$
 - set of Byzantines may change each round (no use of remembering past)

Problem Setting



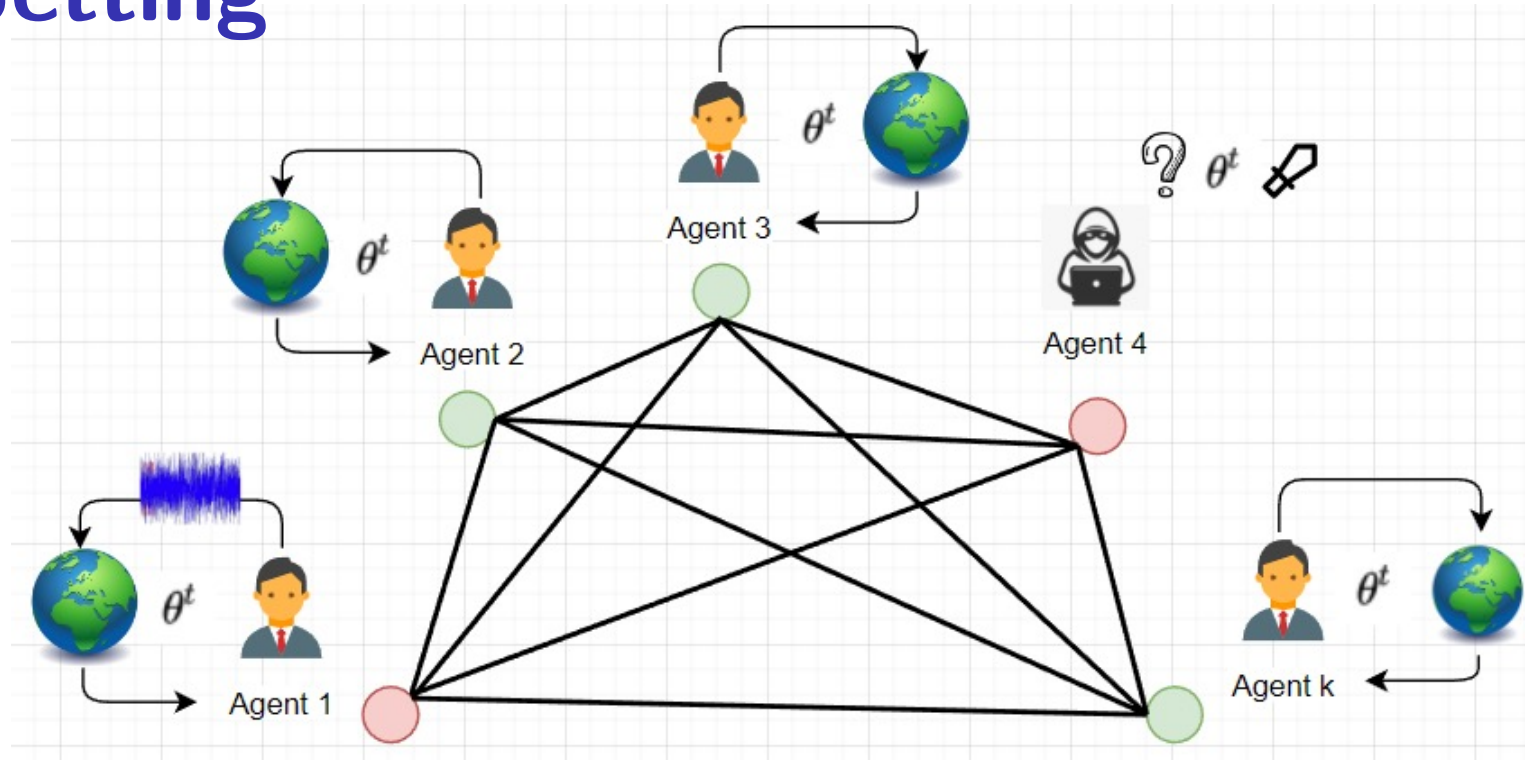
- up to α -fraction of agents may be Byzantine where $\alpha < 1/4$
 - set of Byzantines may change each round (no use of remembering past)
- Byzantine agents may...

Problem Setting



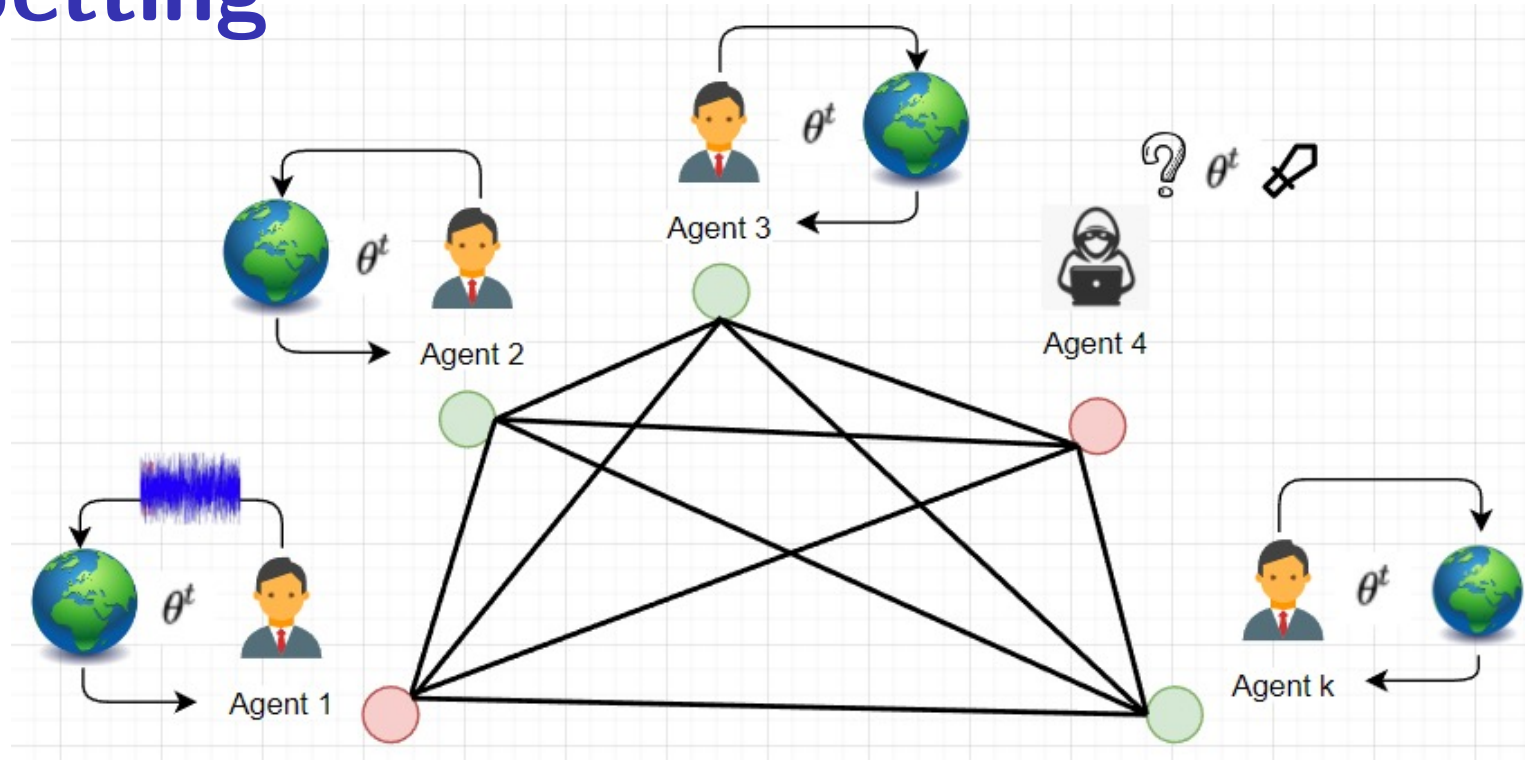
- up to α -fraction of agents may be Byzantine where $\alpha < 1/4$
 - set of Byzantines may change each round (no use of remembering past)
- Byzantine agents may...
 - ...send arbitrary messages, modelling e.g. communication failure, malicious behavior, ...

Problem Setting



- up to α -fraction of agents may be Byzantine where $\alpha < 1/4$
 - set of Byzantines may change each round (no use of remembering past)
- Byzantine agents may...
 - ...send arbitrary messages, modelling e.g. communication failure, malicious behavior, ...
 - ...collude (full information of algorithm, who is Byz., etc.)

Problem Setting



- no trusted server/coordinator among agents!

Related Work & Contribution

	centralized	decentralized
non-fault-tolerant	FEDPG [FMD ⁺ 21]	MDPGT [JLT ⁺ 22]
fault-tolerant	FEDPG-BR [FMD ⁺ 21]	DECBYZ-PG <i>this work</i>

Outline

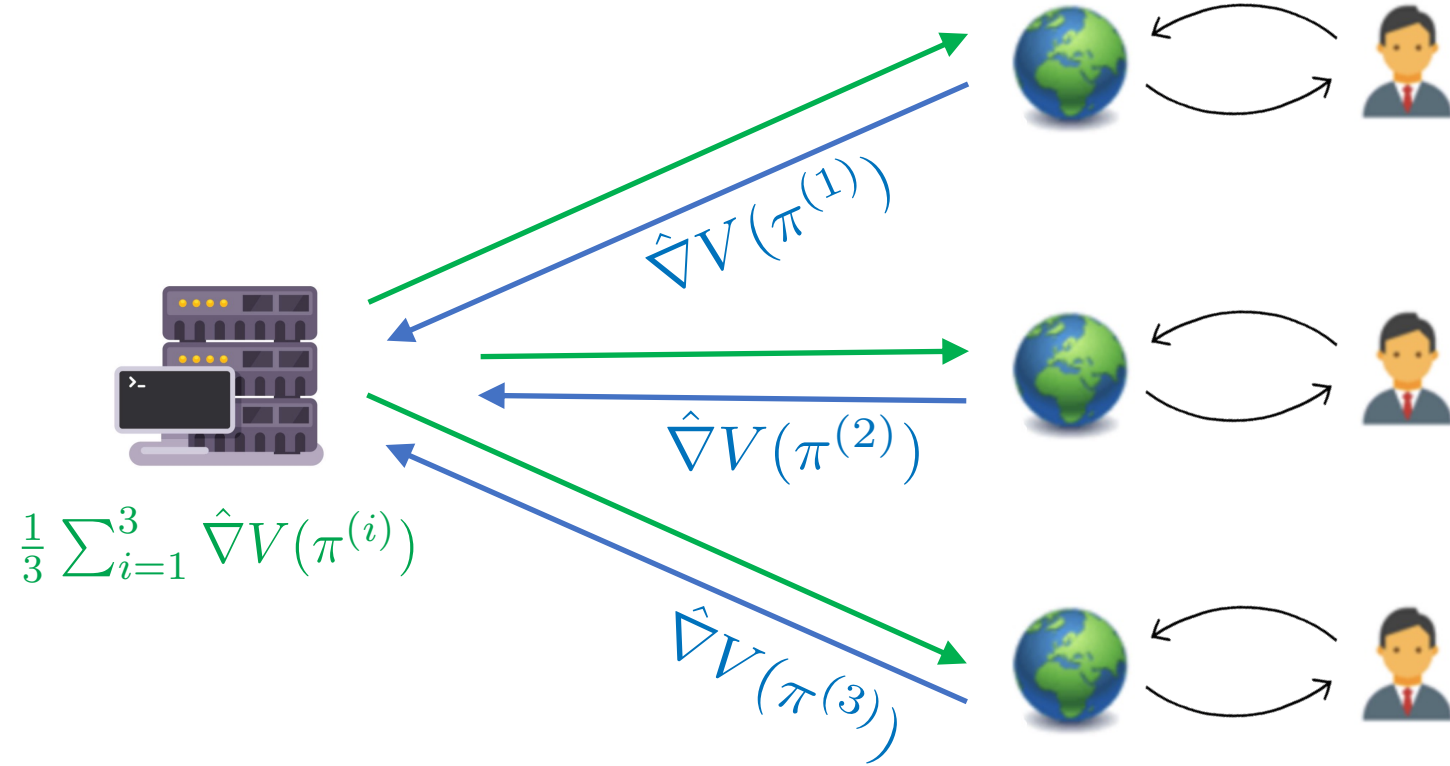
- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Outline

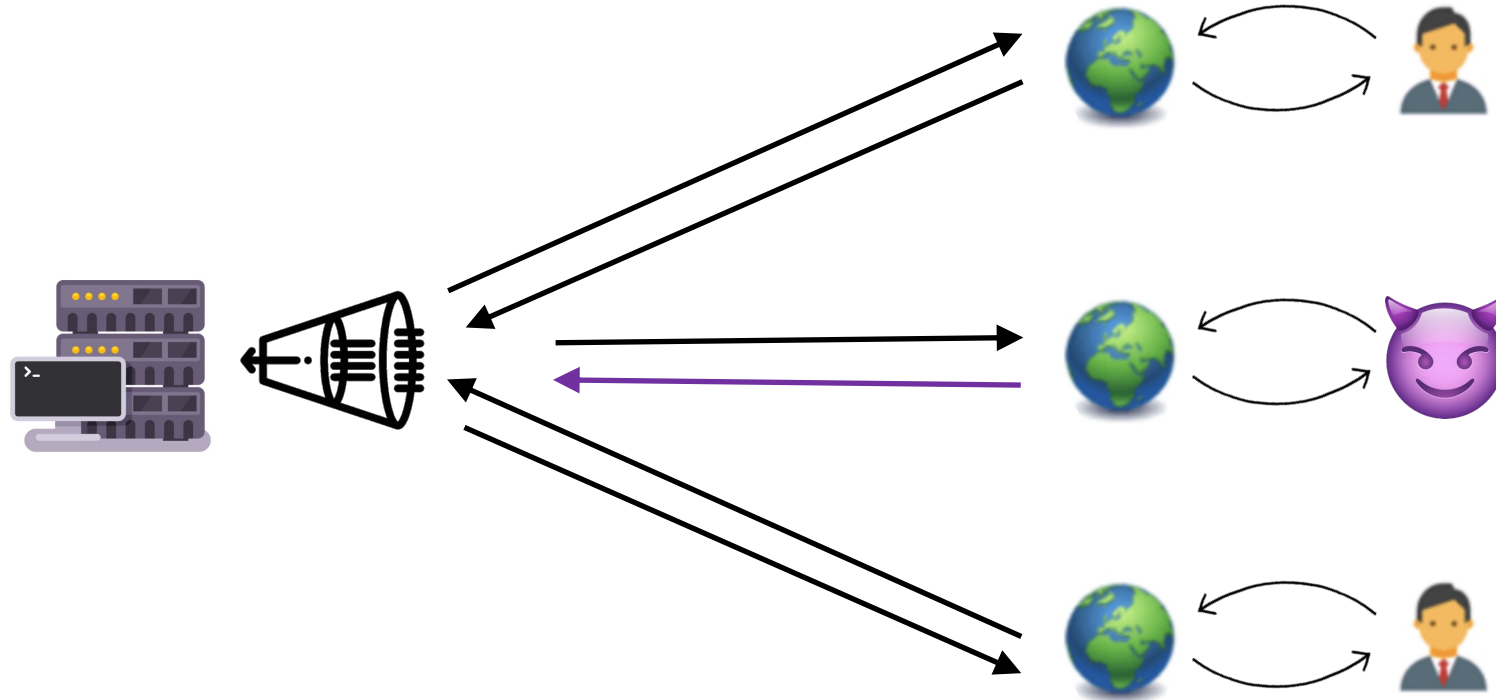
- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method**
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Centralized Federated PG

Centralized Federated PG



Centralized Byz.-tolerant Federated PG



Robust Aggregation

Definition

Aggregate is an (C_{ra}, α) -robust aggregator if for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- any $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,

Robust Aggregation

Definition

Aggregate is an (C_{ra}, α) -robust aggregator if for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- any $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta} := \mathbf{Aggregate}(\theta^{(1)}, \dots, \theta^{(K)})$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

Robust Aggregation

Definition

Aggregate is an (C_{ra}, α) -robust aggregator if for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- any $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta} := \mathbf{Aggregate}(\theta^{(1)}, \dots, \theta^{(K)})$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

$$\mathbb{E} \left[\|\hat{\theta} - \bar{\theta}\|^2 \right] \leq \frac{C_{ra}\alpha}{|\mathcal{H}|(|\mathcal{H}| - 1)} \sum_{i,l \in \mathcal{H}} \mathbb{E} \left[\|\theta^{(i)} - \theta^{(l)}\|^2 \right] .$$

Robust Aggregation

Definition

Aggregate is an (C_{ra}, α) -robust aggregator if for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- any $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta} := \mathbf{Aggregate}(\theta^{(1)}, \dots, \theta^{(K)})$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

$$\mathbb{E} \left[\|\hat{\theta} - \bar{\theta}\|^2 \right] \leq \frac{C_{ra}\alpha}{|\mathcal{H}|(|\mathcal{H}| - 1)} \sum_{i,l \in \mathcal{H}} \mathbb{E} \left[\|\theta^{(i)} - \theta^{(l)}\|^2 \right] .$$

➤ shown to hold when $\alpha < 1/2$ for certain median-like operations (see [GHRR23])

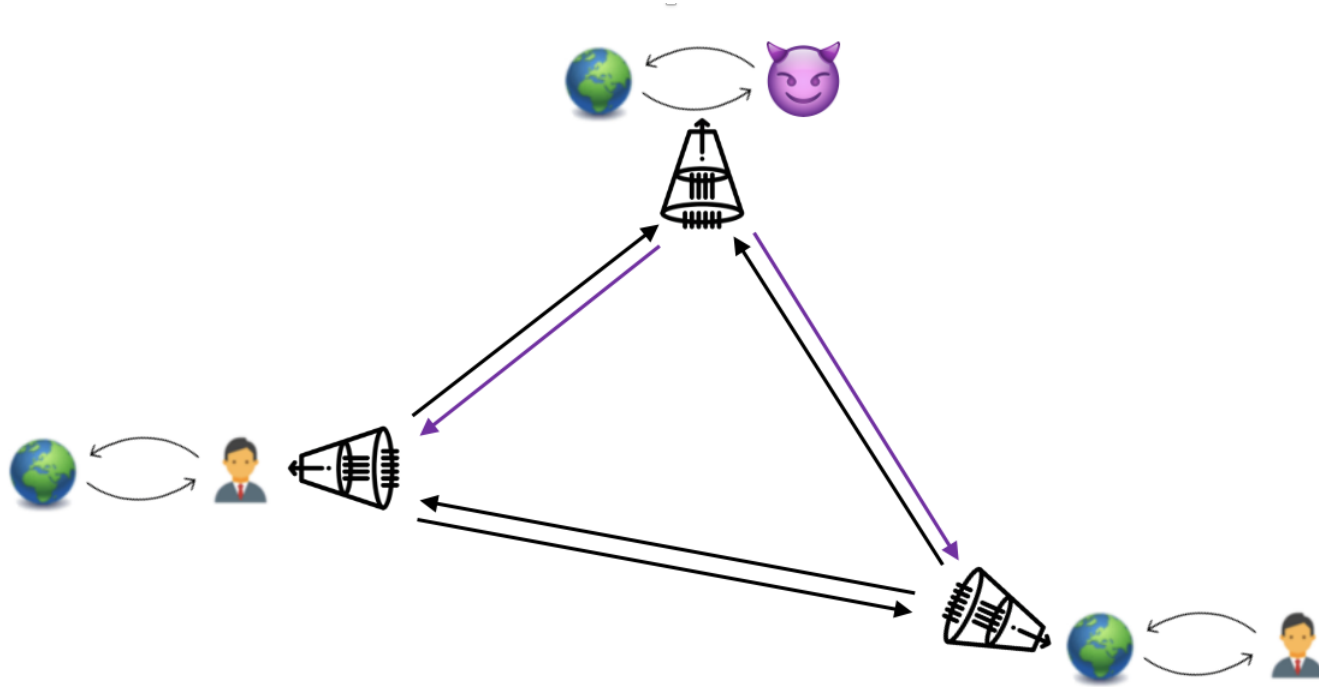
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method**
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

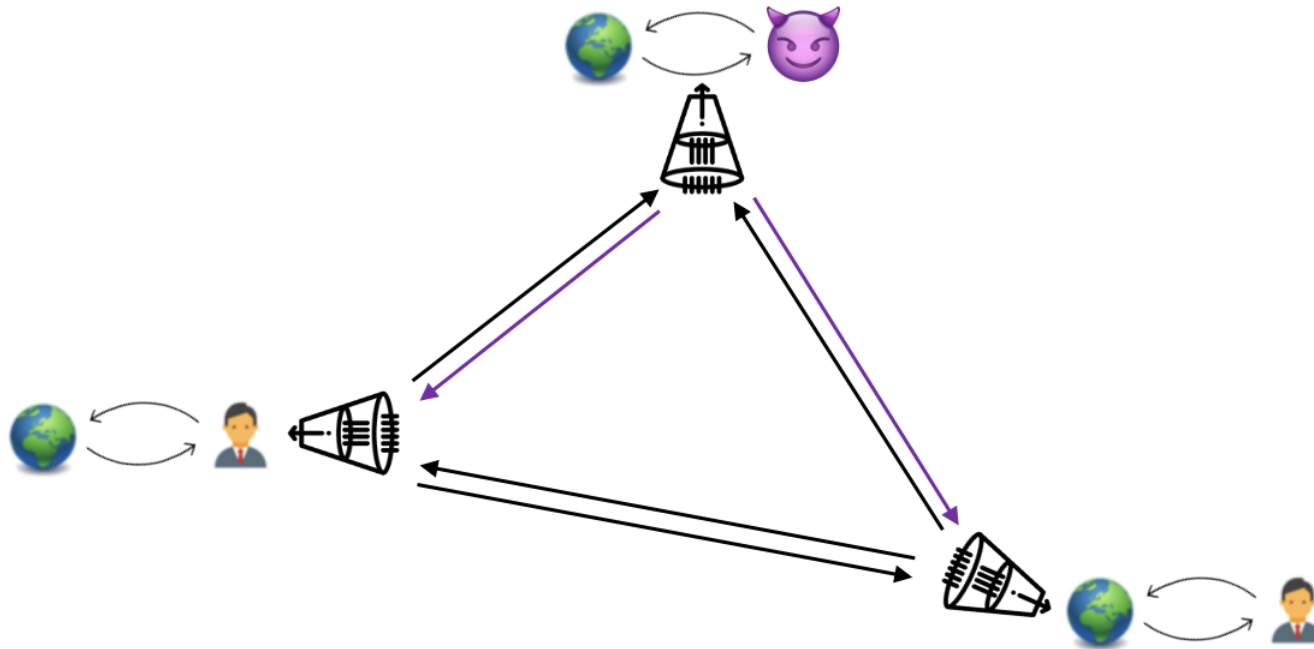
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method**
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Decentralized Byz.-tolerant PG (Idea)



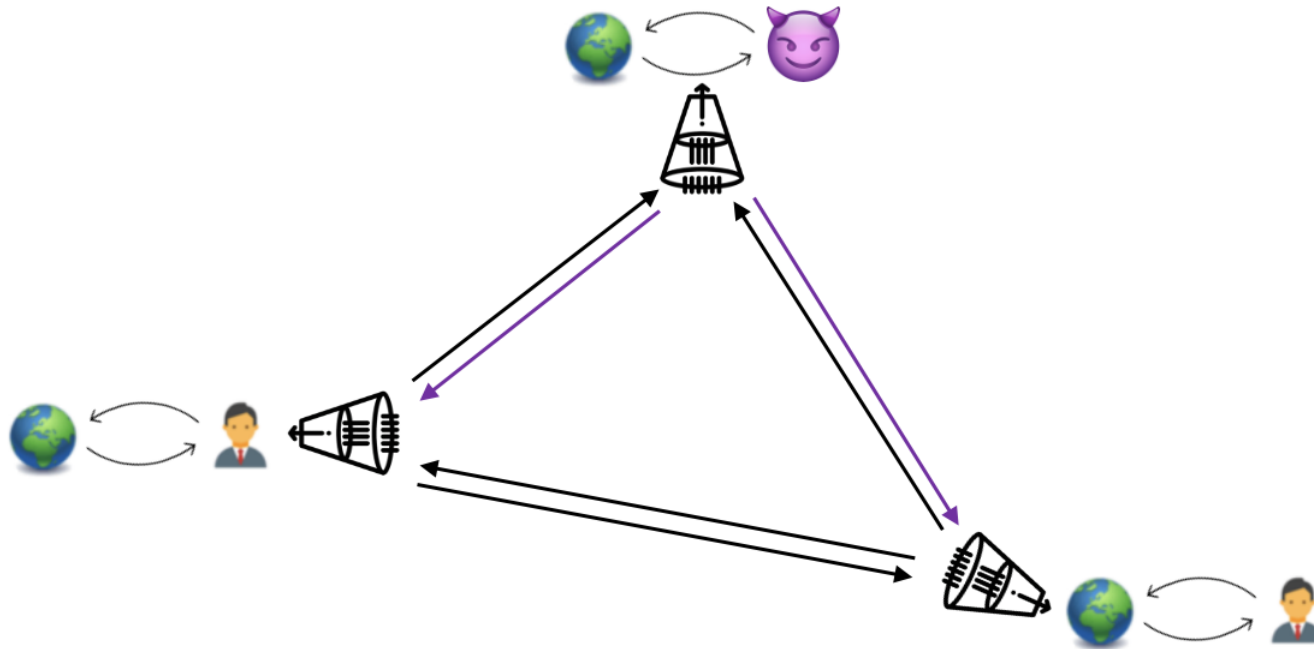
Decentralized Byz.-tolerant PG (Idea)



Algorithm DECBYZ-PG at k -th agent

- 1: initialize $\theta_0^{(k)} \leftarrow \theta_0$
 - 2: **for** $t = 0$ **to** $T - 1$ **do**
 - 3: $\tilde{v}_t^{(k)} \leftarrow$ estimate of $\nabla J \left(\theta_t^{(k)} \right)$
 - 4: $v_t^{(k)} \leftarrow \mathbf{Aggregate} \left(\left\{ \left\{ \tilde{v}_t^{(k')} \right\} \right\}_{k'=1}^K \right)$
 - 5: $\theta_{t+1}^{(k)} \leftarrow \theta_t^{(k)} + \eta v_t^{(k)}$
 - 6:
 - 7: **output:** $\theta_T^{(k)}$
-

Decentralized Byz.-tolerant PG (Idea)

**Algorithm** DECByZ-PG at k -th agent

1: initialize $\theta_0^{(k)} \leftarrow \theta_0$

2: **for** $t = 0$ **to** $T - 1$ **do**

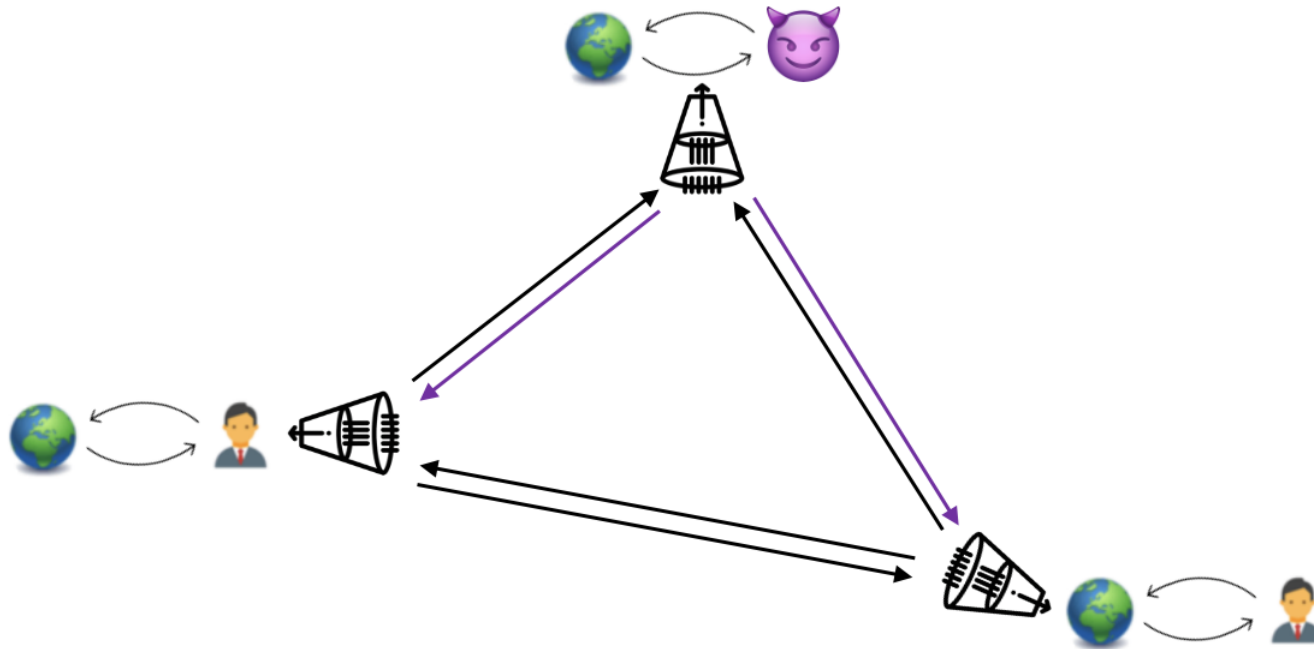
3: $\tilde{v}_t^{(k)} \leftarrow \text{estimate of } \nabla J \left(\theta_t^{(k)} \right)$
$$4: \quad v_t^{(k)} \leftarrow \mathbf{Aggregate} \left(\left\{ \left\{ \tilde{v}_t^{(k')} \right\} \right\}_{k'=1}^K \right)$$
5: $\theta_{t+1}^{(k)} \leftarrow \theta_t^{(k)} + \eta v_t^{(k)}$

6:

7: **output:** $\theta_T^{(k)}$

- **Problem:** Byzantine agent can make $\theta_{t+1}^{(k)}$'s diverge by sending inconsistent gradients!

Decentralized Byz.-tolerant PG (Idea)



Algorithm DECBYZ-PG at k -th agent

- 1: initialize $\theta_0^{(k)} \leftarrow \theta_0$
 - 2: **for** $t = 0$ **to** $T - 1$ **do**
 - 3: $\tilde{v}_t^{(k)} \leftarrow$ estimate of $\nabla J(\theta_t^{(k)})$
 - 4: $v_t^{(k)} \leftarrow \mathbf{Aggregate} \left(\left\{ \left\{ \tilde{v}_t^{(k')} \right\}_{k'=1}^K \right\} \right)$
 - 5: $\tilde{\theta}_{t+1}^{(k)} \leftarrow \theta_t^{(k)} + \eta v_t^{(k)}$
 - 6: $\theta_{t+1}^{(k)} \leftarrow \mathbf{Agree} \left(\left\{ \left\{ \tilde{\theta}_{t+1}^{(k')} \right\}_{k'=1}^K \right\} \right)$
 - 7: **output:** $\theta_T^{(k)}$
-

- **Problem:** Byzantine agent can make $\theta_{t+1}^{(k)}$'s diverge by sending inconsistent gradients!
- **Solution:** Byzantine-tolerant agreement mechanism

Byzantine-tolerant Agreement

Definition

Agree is a (C_{avg}, α) -averaging agreement if after κ rounds of communication, for ...

Byzantine-tolerant Agreement

Definition

Agree is a (C_{avg}, α) -averaging agreement if after κ rounds of communication, for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- honest agents $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,

Byzantine-tolerant Agreement

Definition

Agree is a (C_{avg}, α) -averaging agreement if after κ rounds of communication, for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- honest agents $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta}^{(k)} := \mathbf{Agree}(\theta^{(1)}, \dots, \theta^{(K)})$ at $k \in [K]$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

Byzantine-tolerant Agreement

Definition

Agree is a (C_{avg}, α) -averaging agreement if after κ rounds of communication, for ...

- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- honest agents $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta}^{(k)} := \mathbf{Agree}(\theta^{(1)}, \dots, \theta^{(K)})$ at $k \in [K]$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

$$\max_{i, l \in \mathcal{H}} \|\hat{\theta}^{(i)} - \hat{\theta}^{(l)}\| \leq \frac{\max_{i, l \in \mathcal{H}} \|\theta^{(i)} - \theta^{(l)}\|}{2^\kappa} \quad (\text{bounded diameter})$$

$$\|\bar{\hat{\theta}} - \bar{\theta}\| \leq C_{avg} \cdot \max_{i, l \in \mathcal{H}} \|\theta^{(i)} - \theta^{(l)}\| \quad (\text{bounded shift})$$

Byzantine-tolerant Agreement

Definition

Agree is a (C_{avg}, α) -averaging agreement if after κ rounds of communication, for ...

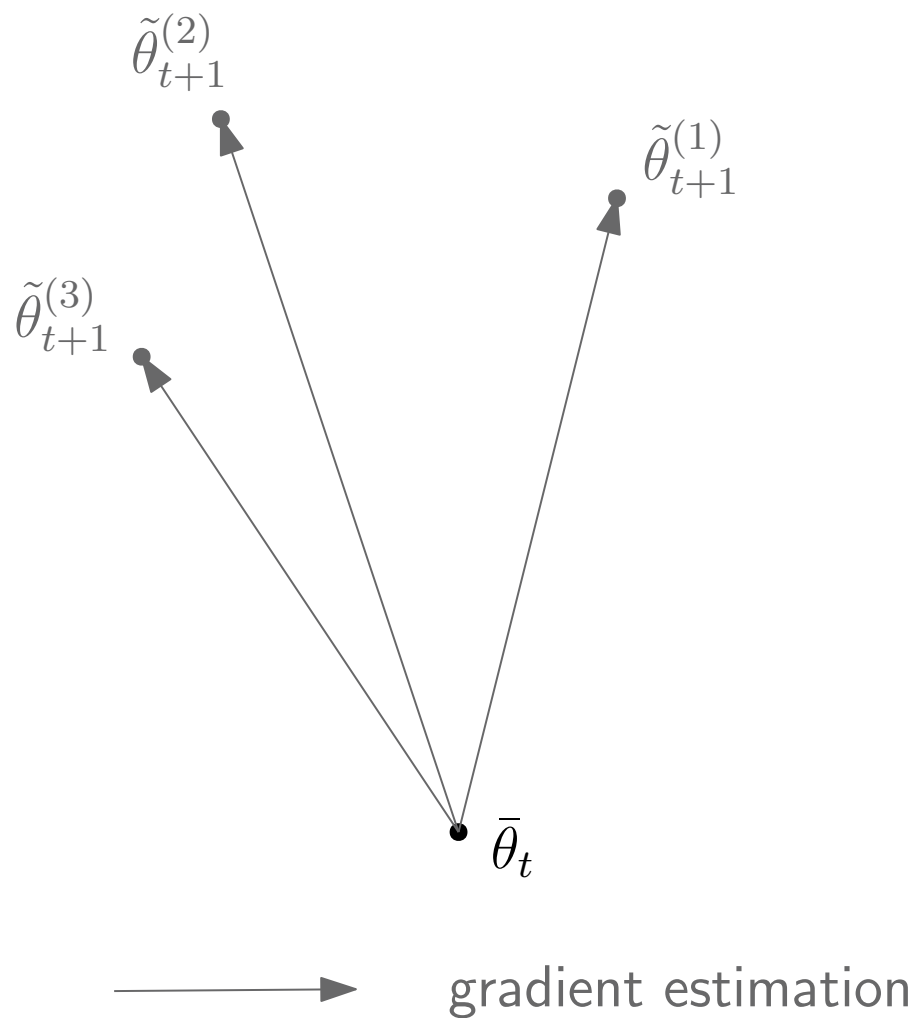
- any inputs $\langle \theta^{(k)} \rangle_{k \in [K]}$,
- honest agents $\mathcal{H} \subseteq [K]$ with $|\mathcal{H}| \geq (1 - \alpha)K$,
- $\hat{\theta}^{(k)} := \mathbf{Agree}(\theta^{(1)}, \dots, \theta^{(K)})$ at $k \in [K]$, and
- $\bar{\theta} := \frac{1}{K} \sum_{k=1}^K \theta^{(k)}$,

$$\max_{i, l \in \mathcal{H}} \|\hat{\theta}^{(i)} - \hat{\theta}^{(l)}\| \leq \frac{\max_{i, l \in \mathcal{H}} \|\theta^{(i)} - \theta^{(l)}\|}{2^\kappa} \quad (\text{bounded diameter})$$

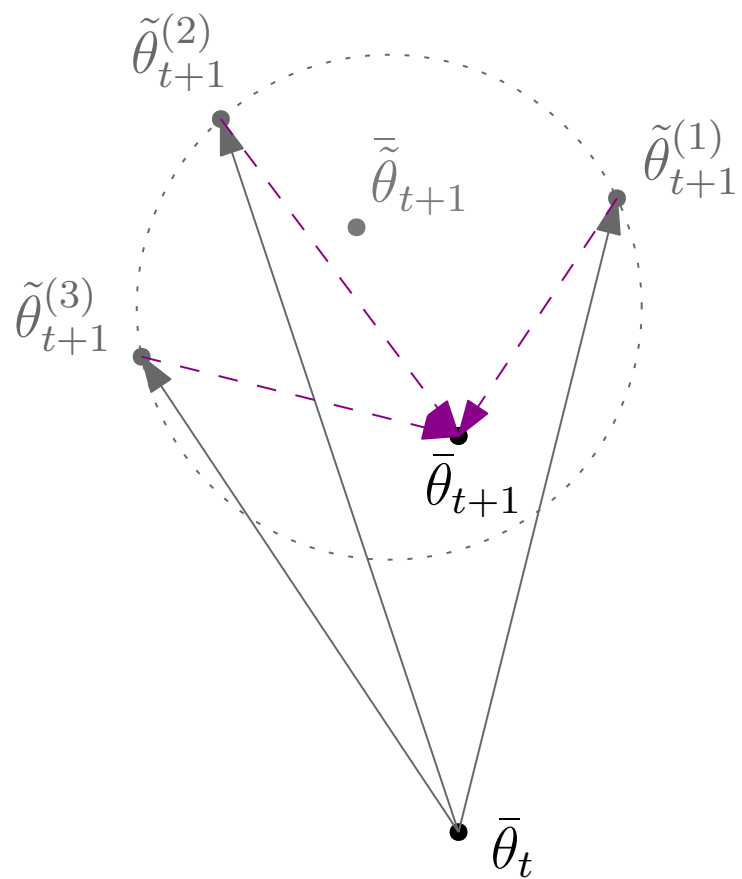
$$\|\bar{\hat{\theta}} - \bar{\theta}\| \leq C_{avg} \cdot \max_{i, l \in \mathcal{H}} \|\theta^{(i)} - \theta^{(l)}\| \quad (\text{bounded shift})$$

- shown to hold when $\alpha < 1/4$ for repeatedly taking averages over "nearest neighbors" (see [EMFG⁺21])

Technical Insight / Intuition

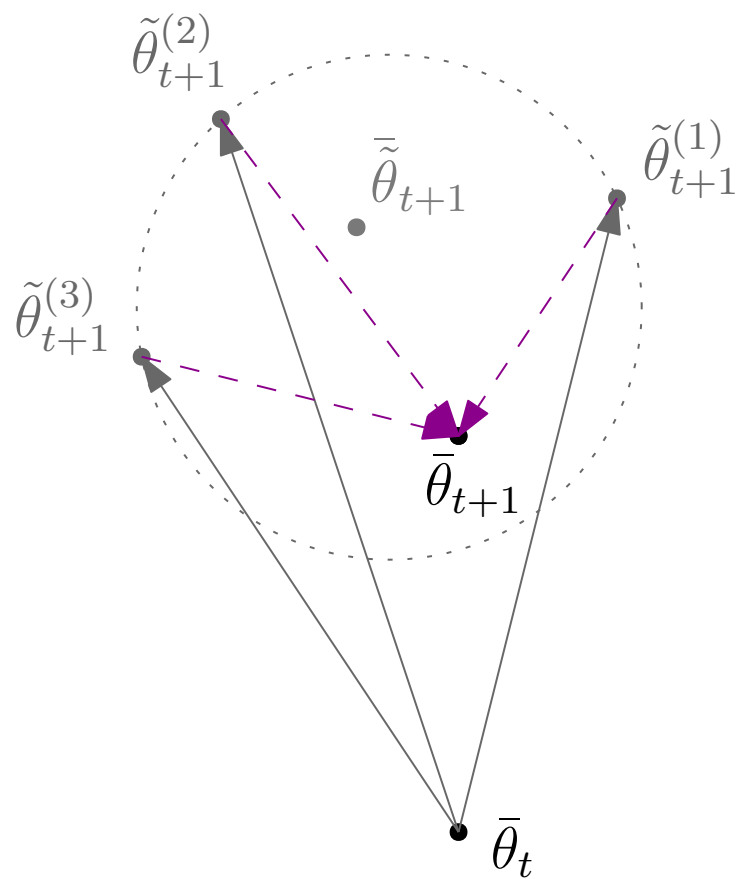


Technical Insight / Intuition



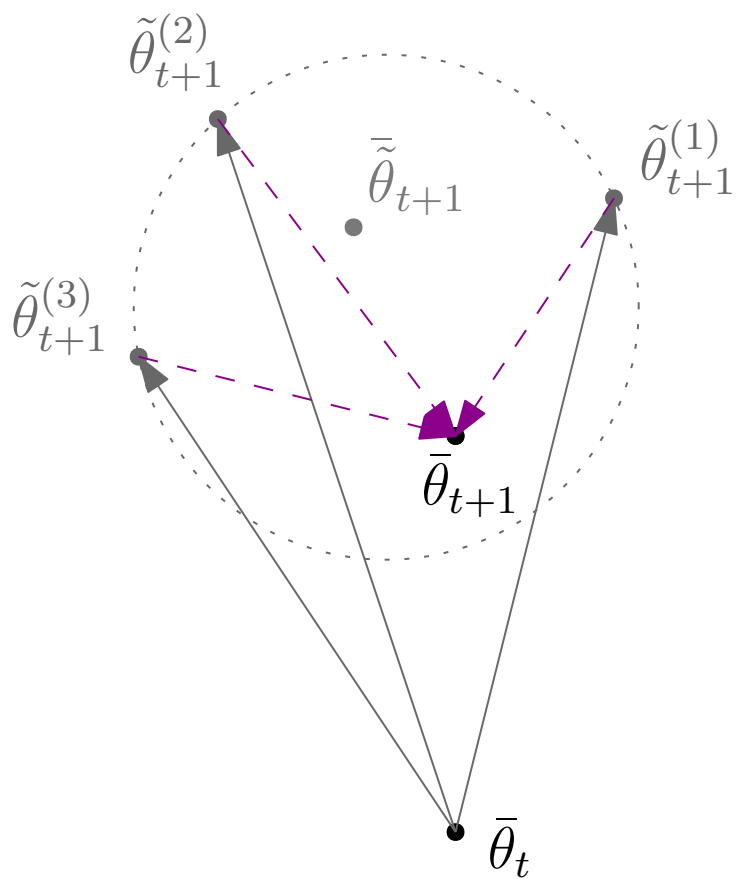
—→ gradient estimation
- - - → agreement

Technical Insight / Intuition



$$\mathbb{E} \left[\text{diameter of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right] \leq O(K^2) \cdot \mathbb{E} \left[\text{pw. distance of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right]$$

Technical Insight / Intuition



$$\mathbb{E} \left[\text{diameter of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right] \leq O(K^2) \cdot \mathbb{E} \left[\text{pw. distance of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right]$$

$$\mathbb{E} \left[\text{diameter of "most" of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right] \leq O(1) \cdot \mathbb{E} \left[\text{pw. distance of } \tilde{\theta}_{t+1}^{(k)} \text{'s} \right]$$

—————▶ gradient estimation
-----▶ agreement

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method**
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result**
- 6) Simulations
- 7) Future Work

Convergence Result

- **K -agent α -tolerant ϵ -approximate solution:**

For $\mathcal{G}_T \subset [K]$ honest agents at round T , we want $\forall k \in \mathcal{G}_T, \mathbb{E}[\|\nabla J(\theta_T^{(k)})\|^2] \leq \epsilon^2$ where $\theta_T^{(k)}$ is agent k 's output.

Convergence Result

- **K -agent α -tolerant ϵ -approximate solution:**

For $\mathcal{G}_T \subset [K]$ honest agents at round T , we want $\forall k \in \mathcal{G}_T, \mathbb{E}[\|\nabla J(\theta_T^{(k)})\|^2] \leq \epsilon^2$ where $\theta_T^{(k)}$ is agent k 's output.

Theorem

For suitably chosen η, κ, N, B, p , the expected sample complexity for DECByzPG to achieve a K -agent α -tolerant ϵ -approximate solution is

$$\mathcal{O} \left(\frac{\alpha^{3/2}}{\epsilon^4} + \frac{\alpha^{1/2}}{K\epsilon^4} + \frac{\alpha^{1/2}}{K^{1/2}\epsilon^3} + \frac{1}{K\epsilon^3} \right).$$

Convergence Result

- **K -agent α -tolerant ϵ -approximate solution:**

For $\mathcal{G}_T \subset [K]$ honest agents at round T , we want $\forall k \in \mathcal{G}_T, \mathbb{E}[\|\nabla J(\theta_T^{(k)})\|^2] \leq \epsilon^2$ where $\theta_T^{(k)}$ is agent k 's output.

Theorem

For suitably chosen η, κ, N, B, p , the expected sample complexity for DECByzPG to achieve a K -agent α -tolerant ϵ -approximate solution is

$$\mathcal{O} \left(\frac{\alpha^{3/2}}{\epsilon^4} + \frac{\alpha^{1/2}}{K\epsilon^4} + \frac{\alpha^{1/2}}{K^{1/2}\epsilon^3} + \frac{1}{K\epsilon^3} \right).$$

- Case $\alpha = 0$: $\mathcal{O}(K^{-1}\epsilon^{-3})$
- Case $\alpha = \mathcal{O}(K^{-1})$: $\mathcal{O}(K^{-3/2}\epsilon^{-4} + K^{-1}\epsilon^{-3})$

Convergence Result

- **K -agent α -tolerant ϵ -approximate solution:**

For $\mathcal{G}_T \subset [K]$ honest agents at round T , we want $\forall k \in \mathcal{G}_T, \mathbb{E}[\|\nabla J(\theta_T^{(k)})\|^2] \leq \epsilon^2$ where $\theta_T^{(k)}$ is agent k 's output.

Theorem

For suitably chosen η, κ, N, B, p , the expected sample complexity for DECByzPG to achieve a K -agent α -tolerant ϵ -approximate solution is

$$\mathcal{O} \left(\frac{\alpha^{3/2}}{\epsilon^4} + \frac{\alpha^{1/2}}{K\epsilon^4} + \frac{\alpha^{1/2}}{K^{1/2}\epsilon^3} + \frac{1}{K\epsilon^3} \right).$$

- Case $\alpha = 0$: $\mathcal{O}(K^{-1}\epsilon^{-3})$
- Case $\alpha = \mathcal{O}(K^{-1})$: $\mathcal{O}(K^{-3/2}\epsilon^{-4} + K^{-1}\epsilon^{-3})$
- **in comparison**, best decentralized non-Byz.-tolerant: $\mathcal{O}(K^{-1}\epsilon^{-3})$

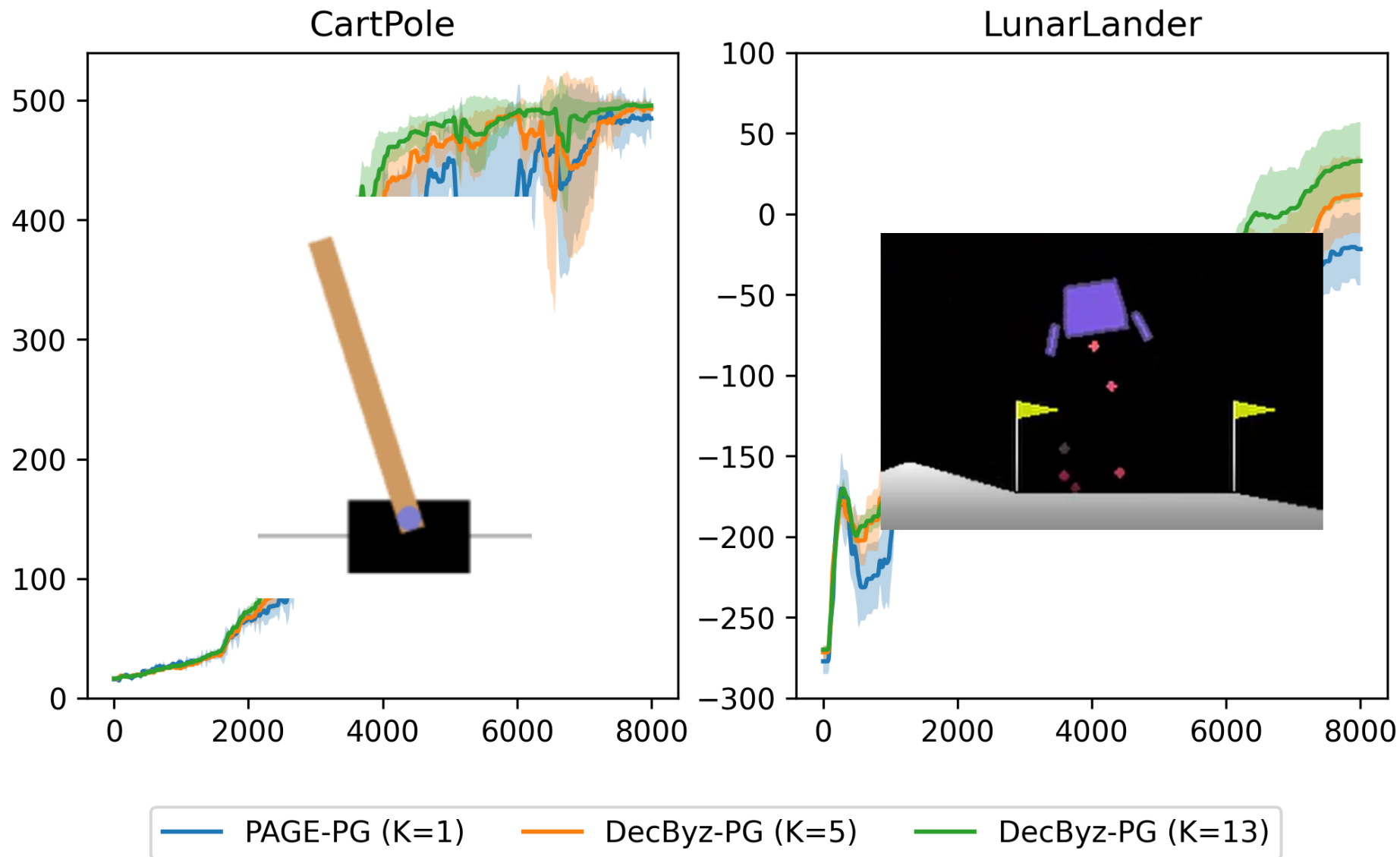
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result**
- 6) Simulations
- 7) Future Work

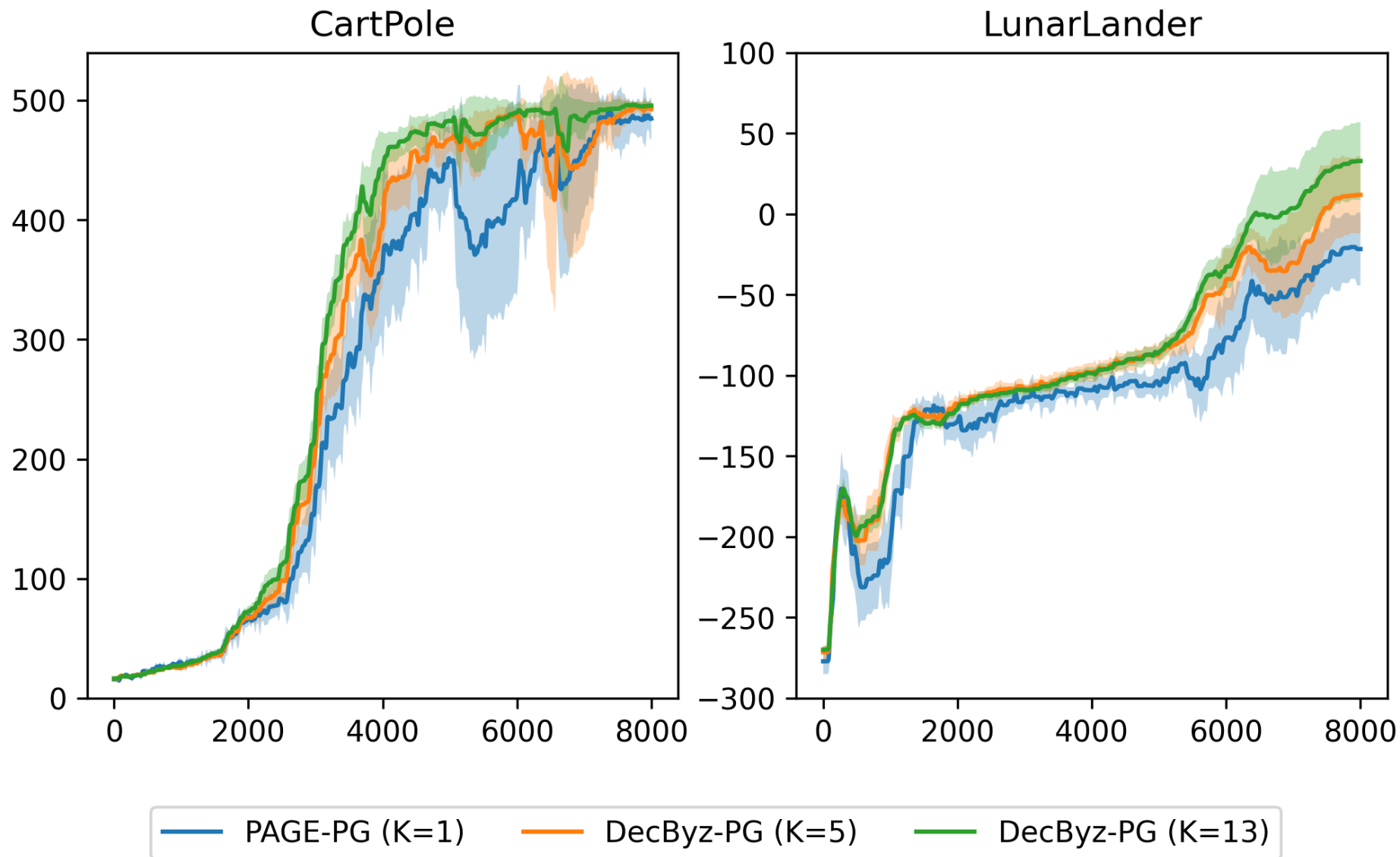
Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations**
- 7) Future Work

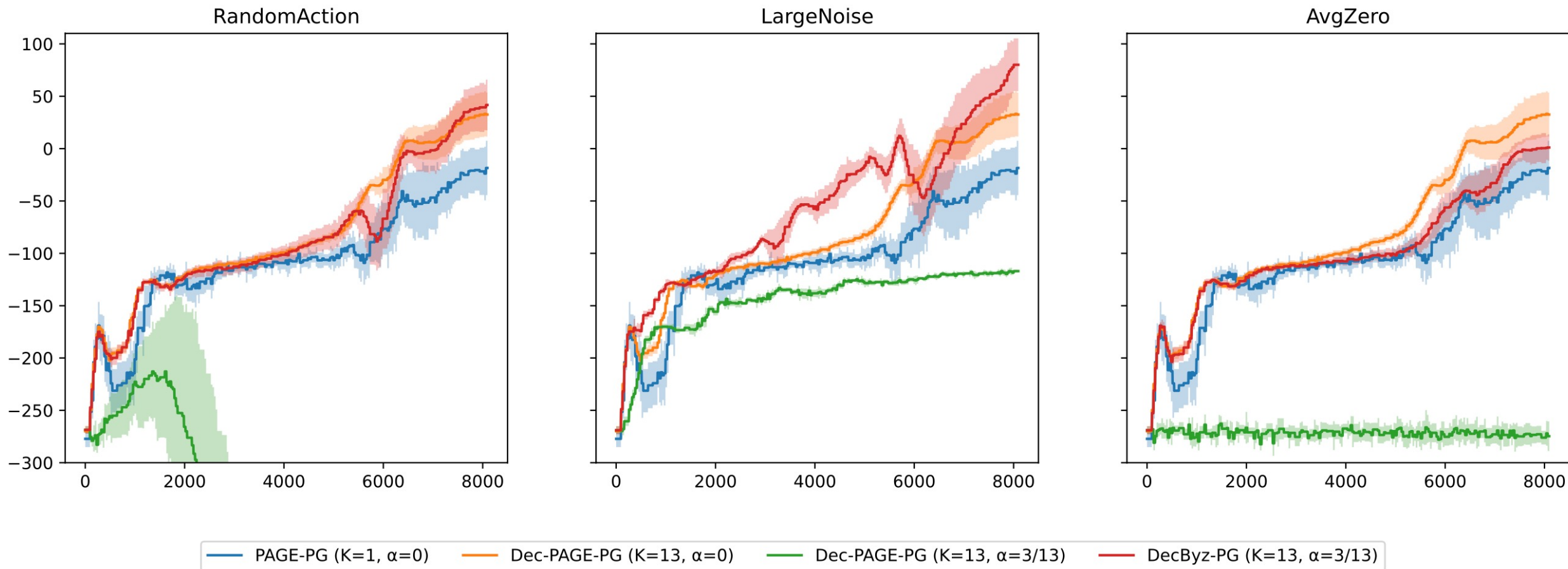
Simulations (speed-up)



Simulations (speed-up)



Simulations (fault-tolerance, LunarLander)



Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations**
- 7) Future Work

Outline

- 1) Motivation
- 2) Setting & Related Work
- 3) Warm-up: Centralized Method
- 4) Decentralized Method
- 5) Convergence Result
- 6) Simulations
- 7) Future Work

Future Work

.

.

.

.

Future Work

- can we tolerate more than $\frac{1}{4}$ -fraction of Byz. agents?

Future Work

- can we tolerate more than $\frac{1}{4}$ -fraction of Byz. agents?
- can message complexity of $\mathcal{O}(K^2)$ be reduced?

Future Work

- can we tolerate more than $\frac{1}{4}$ -fraction of Byz. agents?
- can message complexity of $\mathcal{O}(K^2)$ be reduced?
- heterogeneous environments (effect on fault-tolerance & convergence)

Future Work

- can we tolerate more than $\frac{1}{4}$ -fraction of Byz. agents?
- can message complexity of $\mathcal{O}(K^2)$ be reduced?
- heterogeneous environments (effect on fault-tolerance & convergence)
- additional (differential) privacy guarantees attainable by adding noise to messages?

References I



El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyen Hoang, and Sébastien Rouault.

Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning).

Advances in Neural Information Processing Systems, 34:25044–25057, 2021.



Xiaofeng Fan, Yining Ma, Zhongxiang Dai, Wei Jing, Cheston Tan, and Bryan Kian Hsiang Low.

Fault-tolerant federated reinforcement learning with theoretical guarantee.

In *Advances in Neural Information Processing Systems*, volume 34, pages 1007–1021, 2021.



Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel.

Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top.

In *The Eleventh International Conference on Learning Representations*, 2023.

References II



Zhanhong Jiang, Xian Yeow Lee, Sin Yong Tan, Kai Liang Tan, Aditya Balu, Young M Lee, Chinmay Hegde, and Soumik Sarkar.

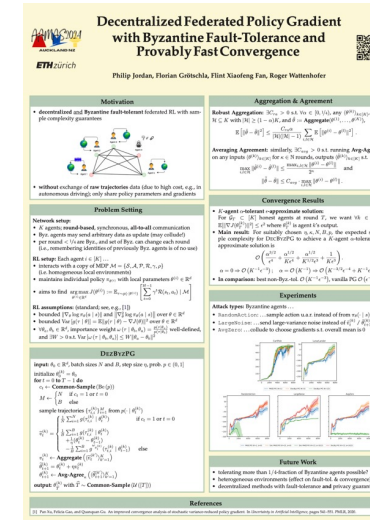
Mdpgt: momentum-based decentralized policy gradient tracking.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9377–9385, 2022.

Paper:



Come by our poster!



Code: <https://github.com/philip-jordan/decentralized-byzantine-RL>

Decentralized Algorithm

Algorithm 2 DECByzPG at the k -th agent

- 1: **input:** $\theta_0 \in \mathbb{R}^d$, large batch size N , small batch size B , step size η , probability $p \in (0, 1]$
 - 2: initialize $\theta_0^{(k)} = \theta_0$
 - 3: **for** $t = 0$ **to** $T - 1$ **do**
 - 4: $c_t \leftarrow \mathbf{Common-Sample}(\text{Be}(p))$
 - 5: sample trajectories $\{\tau_{t,i}^{(k)}\}_{i=1}^M$ from $p(\cdot \mid \theta_t^{(k)})$ where $M = \begin{cases} N & \text{if } c_t = 1 \text{ or } t = 0 \\ B & \text{else} \end{cases}$
 - 6:
$$\tilde{v}_t^{(k)} = \begin{cases} \frac{1}{N} \sum_{i=1}^N g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) & \text{if } c_t = 1 \text{ or } t = 0 \\ \frac{1}{B} \sum_{i=1}^B g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) + \frac{1}{\eta}(\theta_t^{(k)} - \theta_{t-1}^{(k)}) - \frac{1}{B} \sum_{i=1}^B g^{\omega_{\theta_t^{(k)}}}(\tau_{t,i}^{(k)} \mid \theta_{t-1}^{(k)}) & \text{else} \end{cases}$$
 - 7: $v_t^{(k)} \leftarrow \mathbf{Aggregate}(\langle \tilde{v}_t^{(k')} \rangle_{k'=1}^K)$
 - 8: $\tilde{\theta}_{t+1}^{(k)} = \theta_t^{(k)} + \eta v_t^{(k)}$
 - 9: $\theta_{t+1}^{(k)} \leftarrow \mathbf{Agree}_K(\langle \tilde{\theta}_{t+1}^{(k')} \rangle_{k'=1}^K)$
 - 10: **output:** $\theta_{\hat{T}}^{(k)}$ with $\hat{T} \sim \mathbf{Common-Sample}(\mathcal{U}([T]))$
-

Decentralized Algorithm

Algorithm 2 DECByzPG at the k -th agent

- 1: **input:** $\theta_0 \in \mathbb{R}^d$, large batch size N , small batch size B , step size η , probability $p \in (0, 1]$
 - 2: initialize $\theta_0^{(k)} = \theta_0$
 - 3: **for** $t = 0$ **to** $T - 1$ **do**
 - 4: $c_t \leftarrow \mathbf{Common-Sample}(\text{Be}(p))$
 - 5: sample trajectories $\{\tau_{t,i}^{(k)}\}_{i=1}^M$ from $p(\cdot \mid \theta_t^{(k)})$ where $M = \begin{cases} N & \text{if } c_t = 1 \text{ or } t = 0 \\ B & \text{else} \end{cases}$
 - 6:
$$\tilde{v}_t^{(k)} = \begin{cases} \frac{1}{N} \sum_{i=1}^N g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) & \text{if } c_t = 1 \text{ or } t = 0 \\ \frac{1}{B} \sum_{i=1}^B g(\tau_{t,i}^{(k)} \mid \theta_t^{(k)}) + \frac{1}{\eta}(\theta_t^{(k)} - \theta_{t-1}^{(k)}) - \frac{1}{B} \sum_{i=1}^B g^{\omega_{\theta_t^{(k)}}}(\tau_{t,i}^{(k)} \mid \theta_{t-1}^{(k)}) & \text{else} \end{cases}$$
 - 7: $v_t^{(k)} \leftarrow \mathbf{Aggregate}(\langle \tilde{v}_t^{(k')} \rangle_{k'=1}^K)$
 - 8: $\tilde{\theta}_{t+1}^{(k)} = \theta_t^{(k)} + \eta v_t^{(k)}$
 - 9: $\theta_{t+1}^{(k)} \leftarrow \mathbf{Agree}_K(\langle \tilde{\theta}_{t+1}^{(k')} \rangle_{k'=1}^K)$
 - 10: **output:** $\theta_{\hat{T}}^{(k)}$ with $\hat{T} \sim \mathbf{Common-Sample}(\mathcal{U}([T]))$
-

Simulations (fault-tolerance, CartPole)

