Structure-Exploiting Multi-Agent Reinforcement Learning for Networked Systems

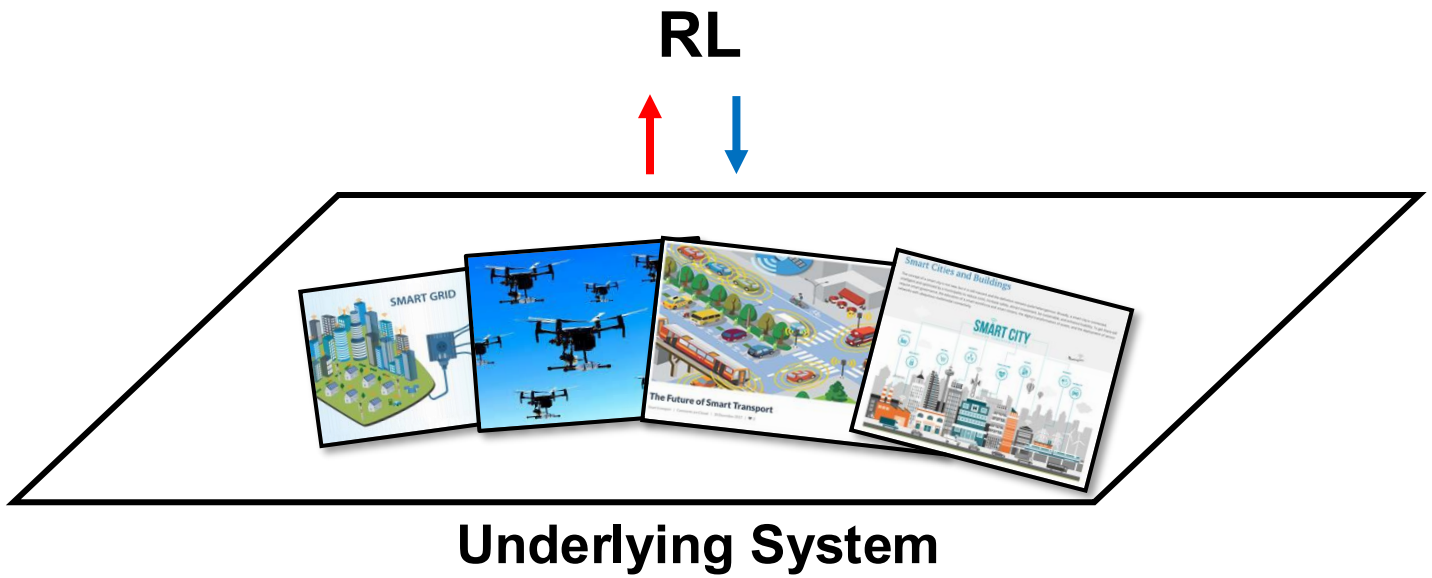
Guannan Qu
Associate Professor of ECE
Carnegie Mellon University

SIGMETRICS, June 8, 2026

Collaborators: Alex Deweese, Jiaoyang Li, Adam Wierman, Na Li, Yiheng Lin, Emile Anand, Pan Xu, Yizhou Zhang, Zaiwei Chen, Longbo Huang, Guanya Shi, Chaoyi Pan, Zeji Yi..

Picture credit: IEEE, qrowd, ARC advisory group

Opportunities of RL for networked systems?



Model Unavailability

- Unknown system dynamics
- Hard-to-model behaviors
- Uncertain disturbances

Optimize Performance

- Deep RL can help improve performance for non-convex control problems

Online Adaptivity

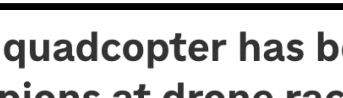
- Online self-adapt to time-varying system changes

There are already some success stories…

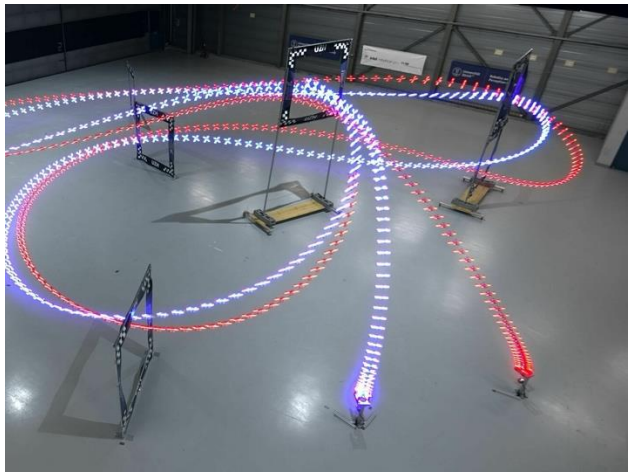
An AI quadcopter has beaten human champions at drone racing

AUGUST 30, 2023 · 2:36 PM ET

HEARD ON ALL THINGS CONSIDERED



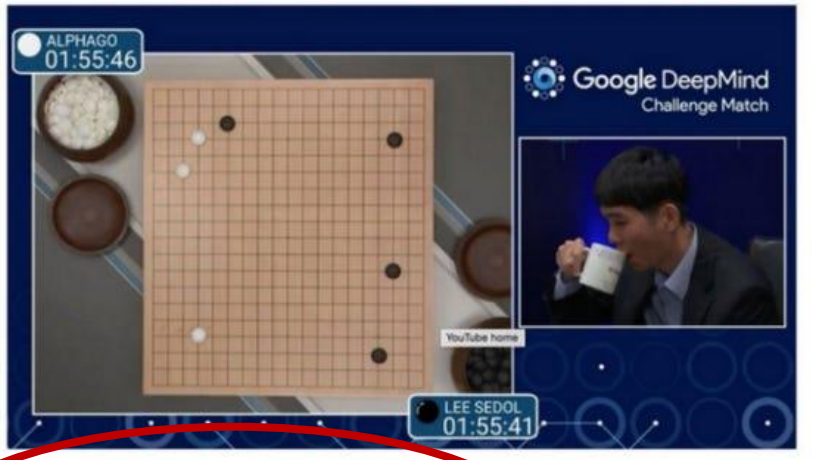
Geoff Brumfiel



[Source: FT, NPR]

… but also many challenges and hurdles

Scalability



AlphaGO
1202 CPUs, 176 GPUs,
100+ Scientists.

Lee Se-dol
1 Human Brain,
1 Coffee.

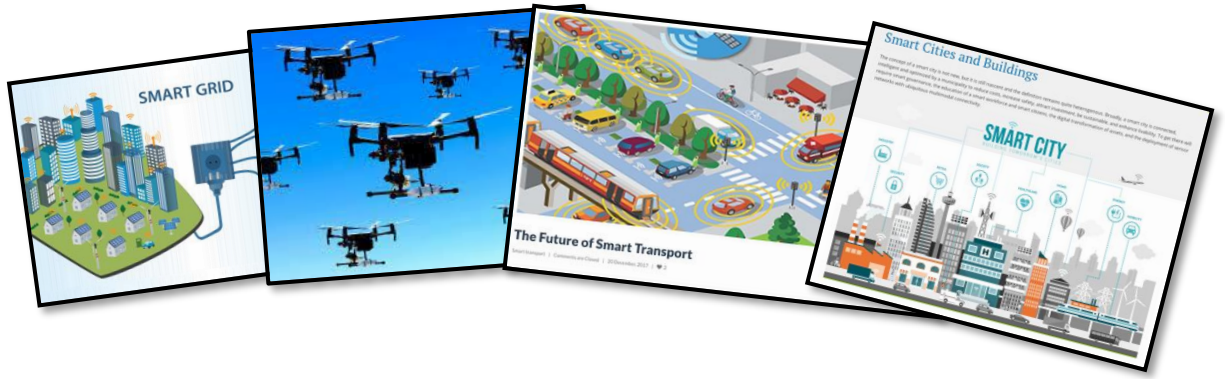
The StarCraft Multi-Agent Challenge

(a) 2c_vs_64zg (b) 3s_vs_5z (c) corridor

Figure 1: Screenshots of three SMAC scenarios.

Name	Ally Units	Enemy Units
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots
3s5z	3 Stalkers & 5 Zealots	3 Stalkers & 5 Zealots
1c3s5z	1 Colossus, 3 Stalkers & 5 Zealots	1 Colossus, 3 Stalkers & 5 Zealots
5m_vs_6m	5 Marines	6 Marines
10m_vs_11m	10 Marines	11 Marines
27m_vs_30m	27 Marines	30 Marines
3s5z_vs_3s6z	3 Stalkers & 5 Zealots	3 Stalkers & 6 Zealots
MMM2	1 Medivac, 2 Marauders & 7 Marines	1 Medivac, 3 Marauders & 8 Marines
2s_vs_1sc	2 Stalkers	1 Spine Crawler
3s_vs_3z	3 Stalkers	3 Zealots
6h_vs_8z	6 Hydralisks	8 Zealots
bane_vs_bane	20 Zerglings & 4 Banelings	20 Zerglings & 4 Banelings
2c_vs_64zg	2 Colossi	64 Zerglings
corridor	6 Zealots	24 Zerglings

Existing work rarely exists 100 agents



Existing MARL Theories are often pessimistic for these applications

State space is factored

$s = (s_1, s_2, \ldots, s_n)$

i.e. exponentially large in the number of agents

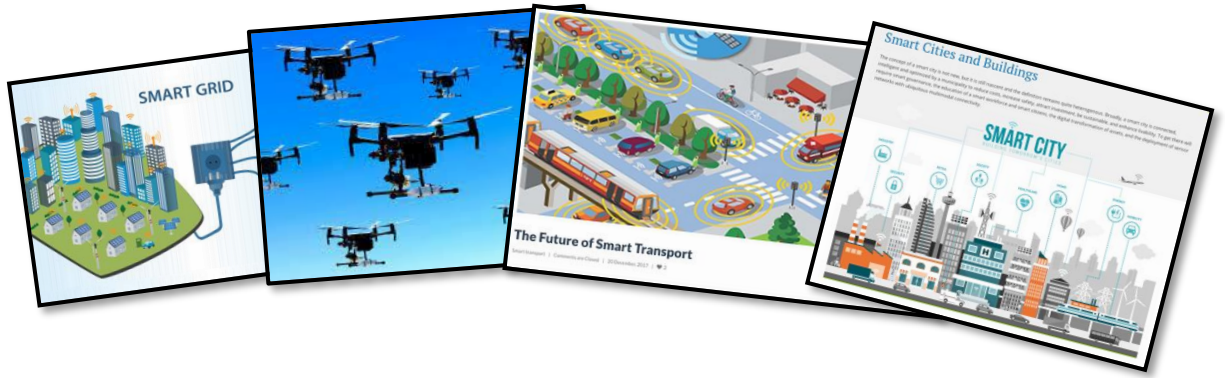
Factored MDP is intractable

[Kearns and Koller, 1999; Guestrin et al., 2003]

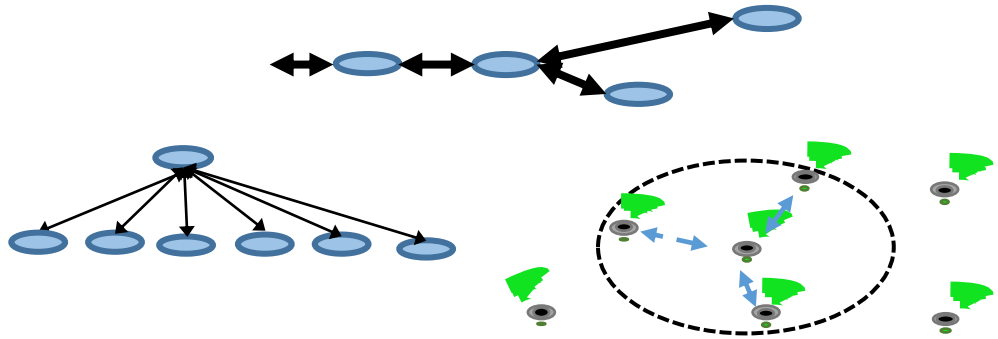
Each agent only has partial observation

Dec-POMDP is general NEXP-complete

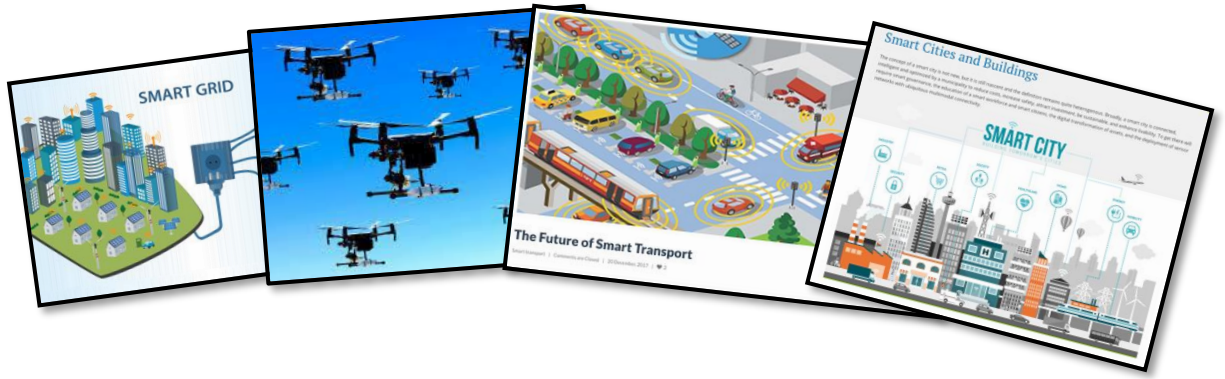
[Oliehoek, 2012; Oliehoek et al., 2016].



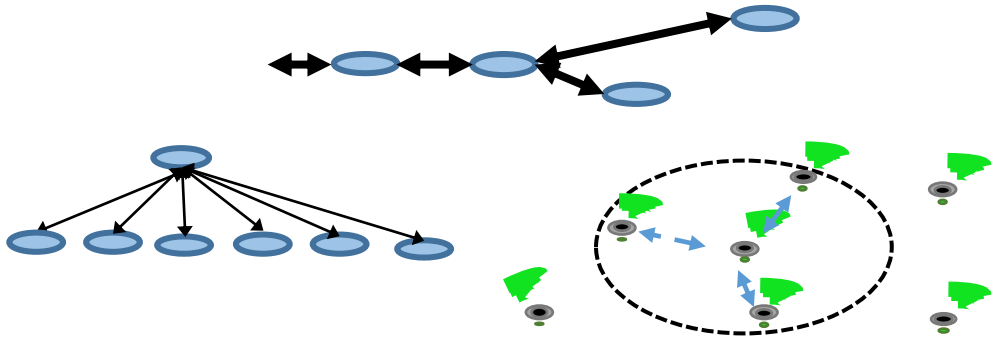
Extract abstract structural property



$$x_{t+1} = f_{physics}(x_t, u_t)$$



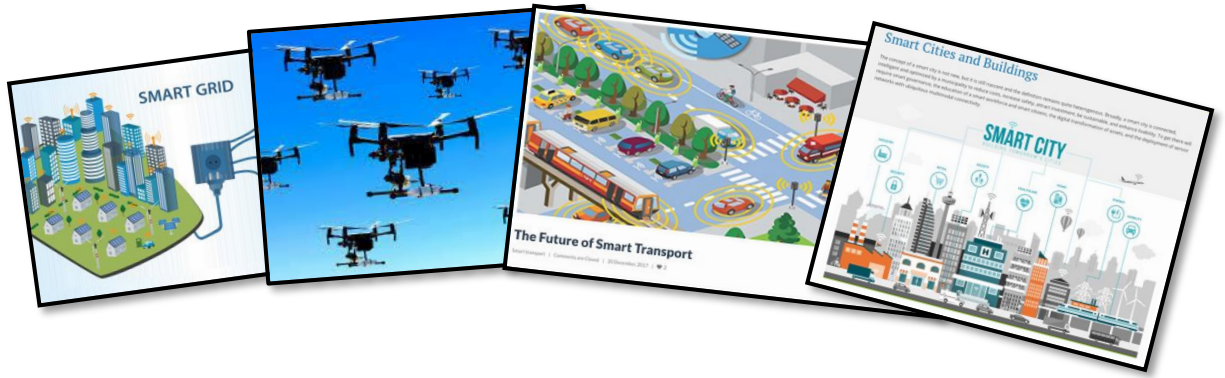
Extract abstract structural property



$$x_{t+1} = f_{physics}(x_t, u_t)$$

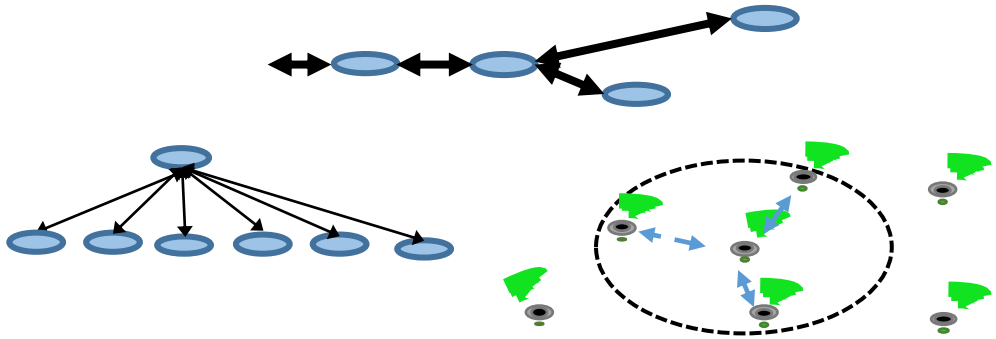
Exploit structure to obtain "positive" MARL theory

MARL Algorithm
with theoretical guarantee
that **avoids intractability**



Apply to applications

Extract abstract structural property

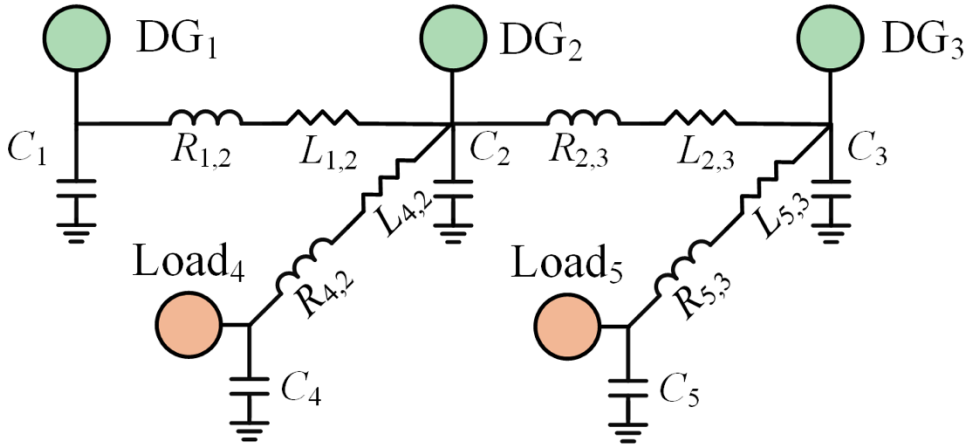


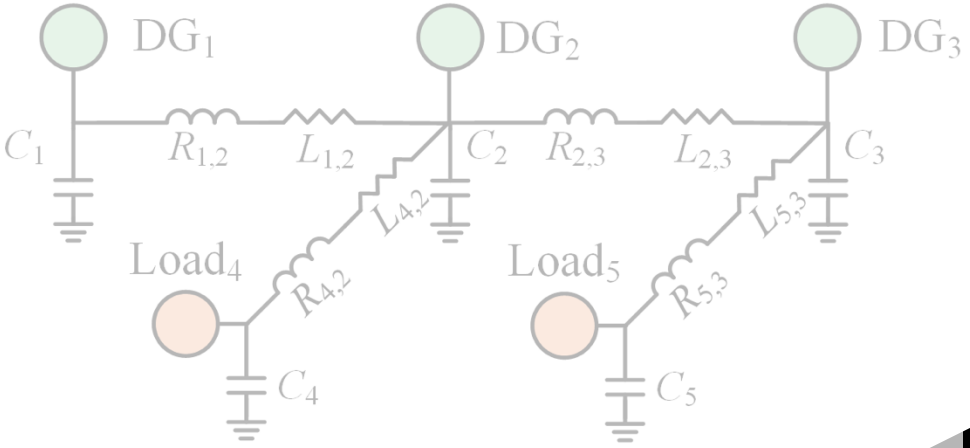
$$x_{t+1} = f_{physics}(x_t, u_t)$$

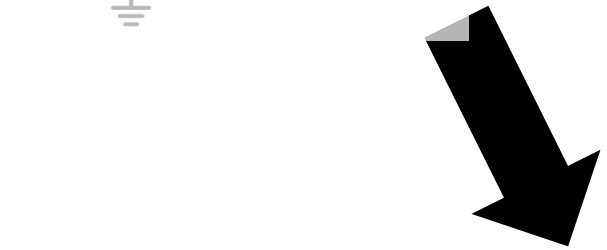
MARL Algorithm
with theoretical guarantee
that **avoids intractability**

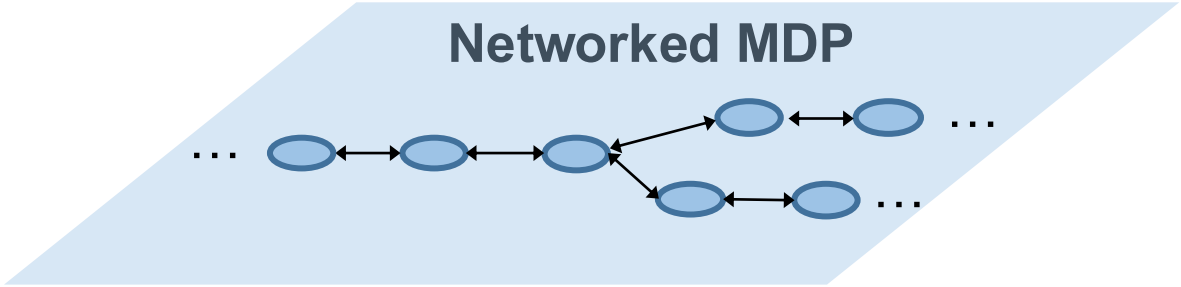
Exploit structure to obtain
"positive" MARL theory

Power Network

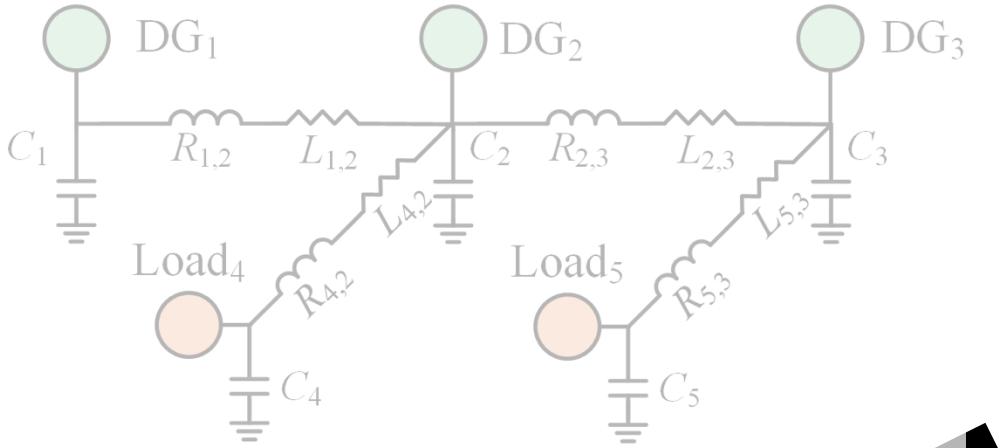


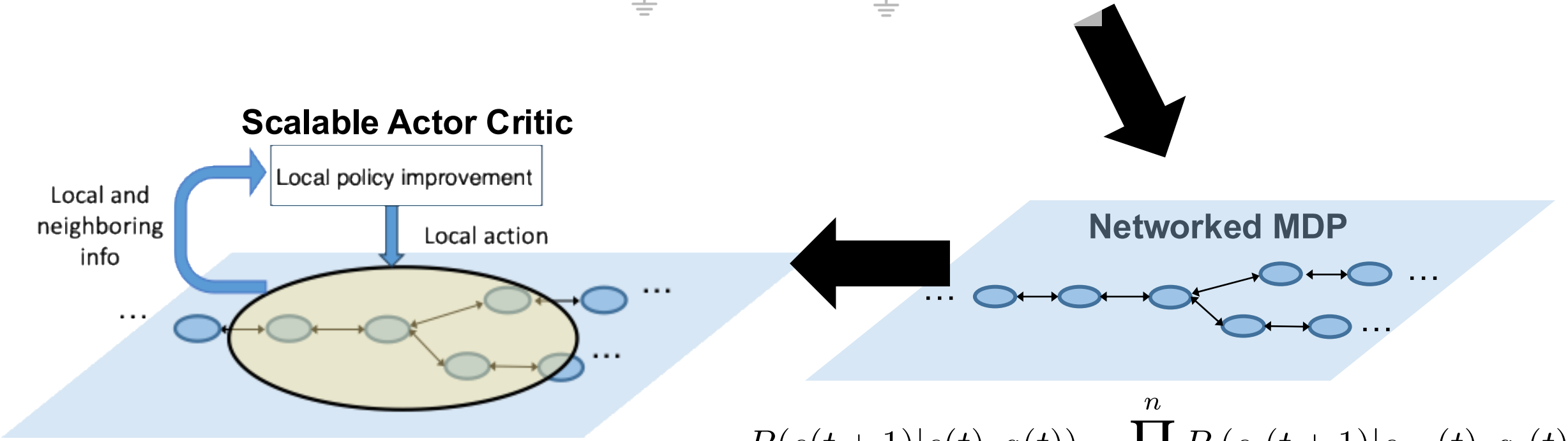






$$P(s(t+1)|s(t),a(t)) = \prod_{i=1}^{n} P_i(s_i(t+1)|s_{N_i}(t),a_i(t)).$$



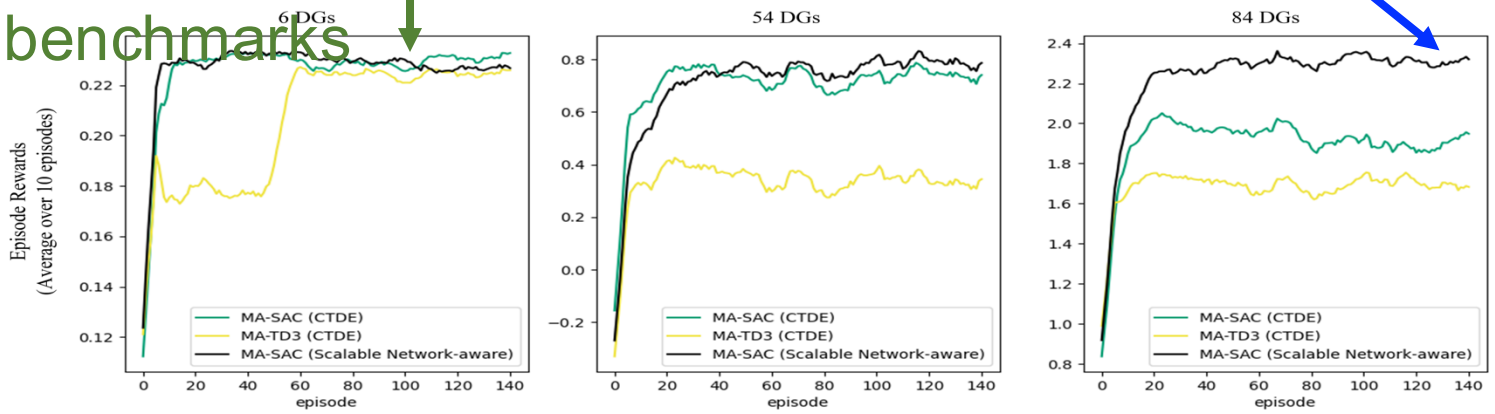


$$P(s(t+1)|s(t),a(t)) = \prod_{i=1}^{n} P_i(s_i(t+1)|s_{N_i}(t),a_i(t)).$$

[**Qu**, Wieman, Li Operations Research 2022] [**Qu**, Lin, Wierman, Li, NeurIPS 2020] [Zhang, **Qu**, Xu, Lin, Chen, Wierman SIGMETRICS 2023]

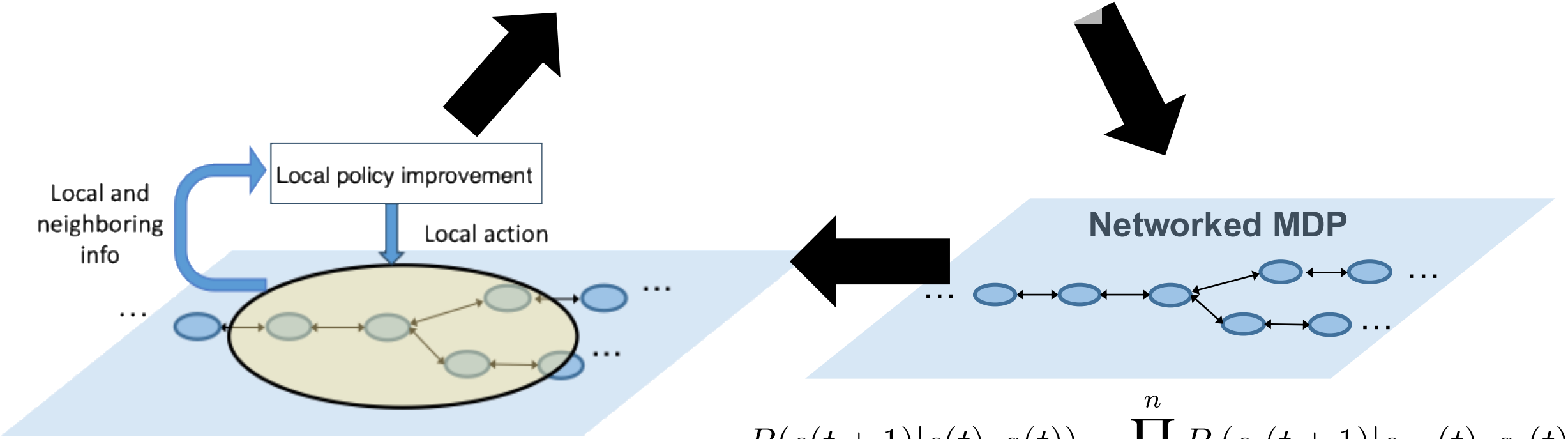
For small systems, our approach is similar to benchmarks

For large systems, our approach is the best



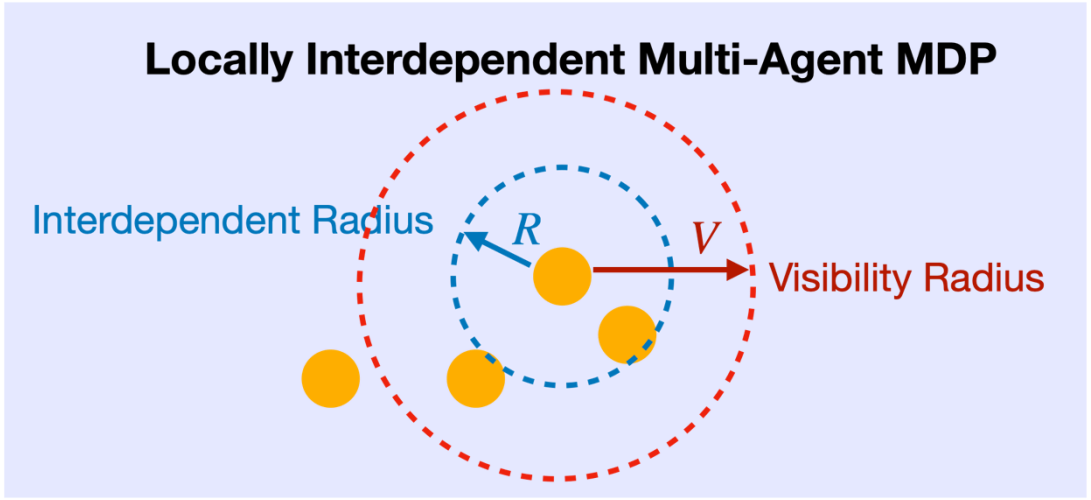
Our approach

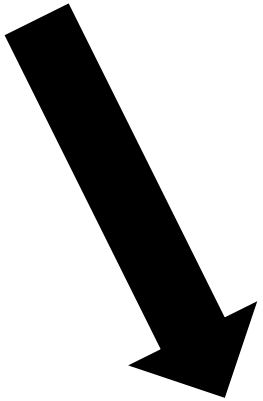
} Benchmarks without using structure

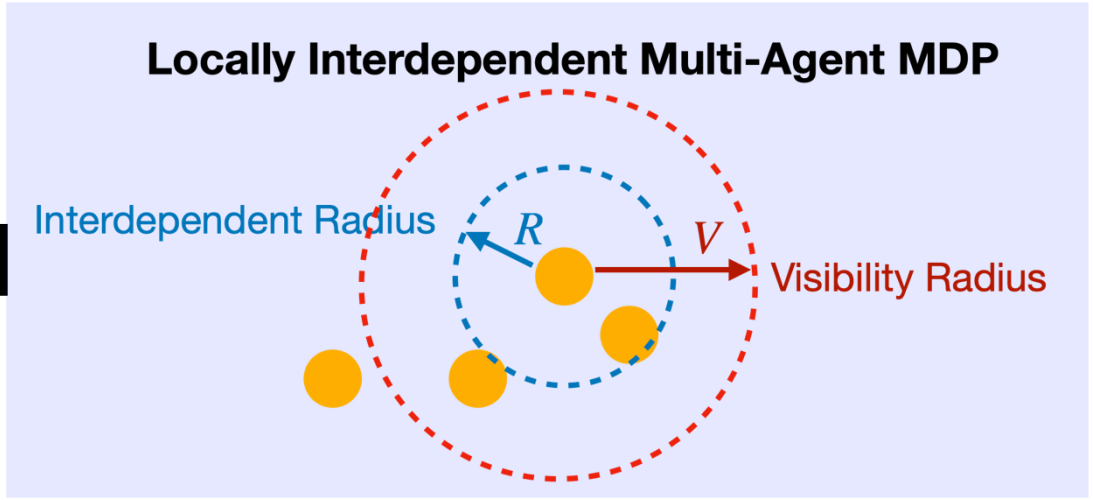


$$P(s(t+1)|s(t),a(t)) = \prod_{i=1}^{n} P_i(s_i(t+1)|s_{N_i}(t),a_i(t)).$$

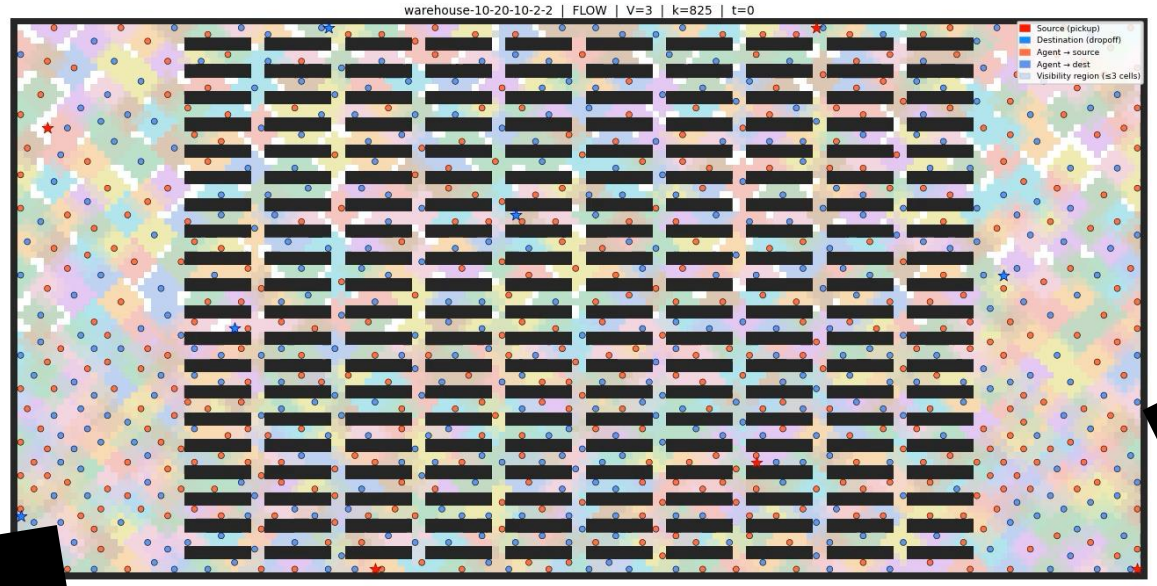
[**Qu**, Wieman, Li Operations Research 2022] [**Qu**, Lin, Wierman, Li, NeurIPS 2020] [Zhang, **Qu**, Xu, Lin, Chen, Wierman SIGMETRICS 2023]


Locally Interdependent Multi-Agent MDP
Interdependent Radius
R
V
Visibility Radius

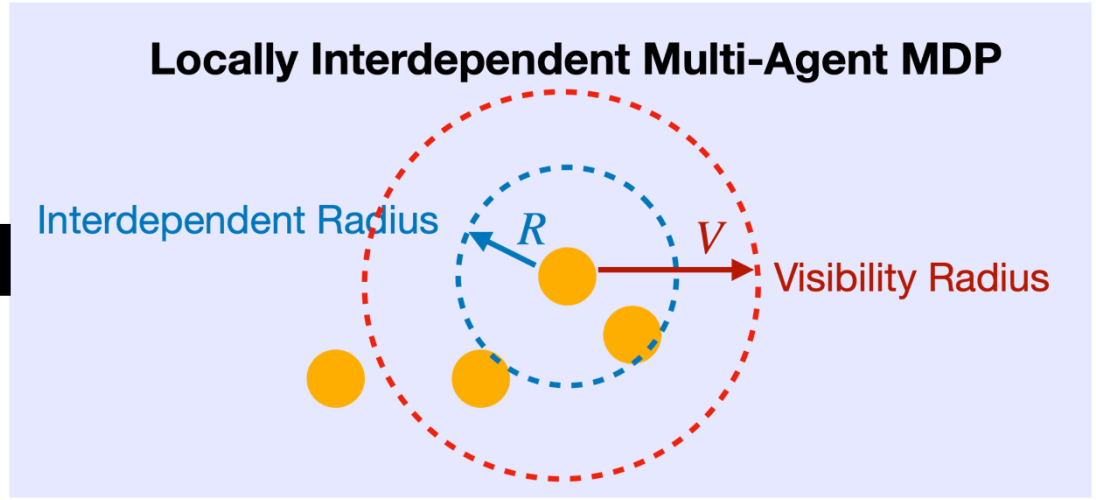




[DeWeese and Qu 2024]
Policy decompositions that are scalable and near optimal

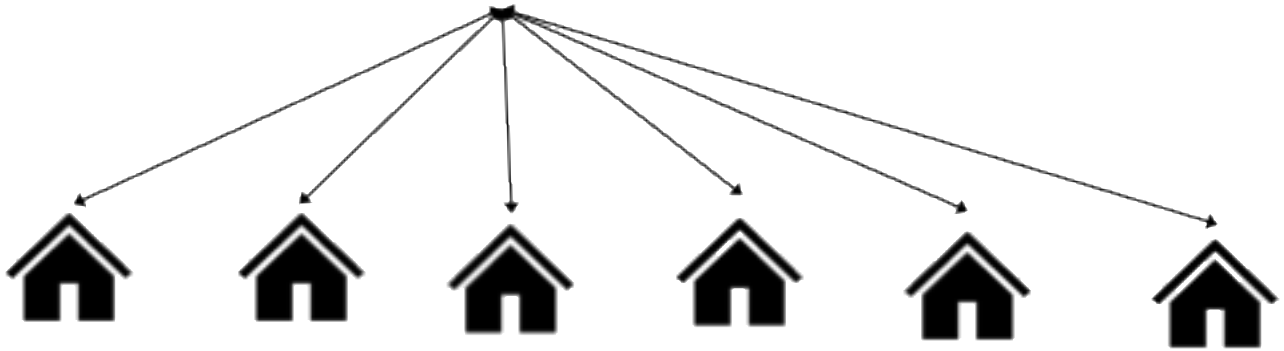


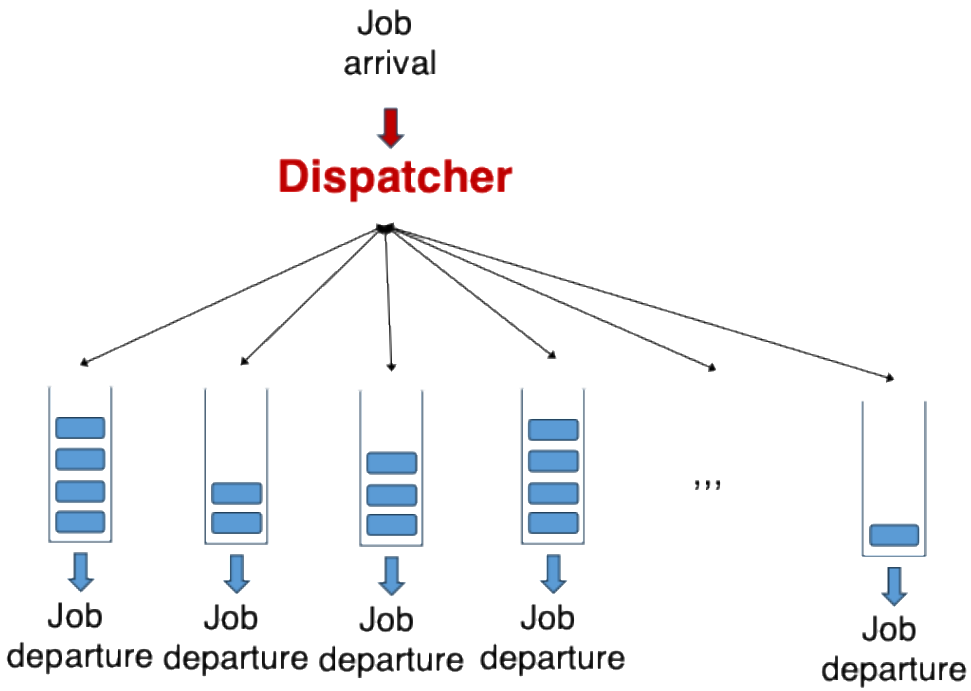
[DeWeese and Qu 2024]
Policy decompositions that are scalable and near optimal

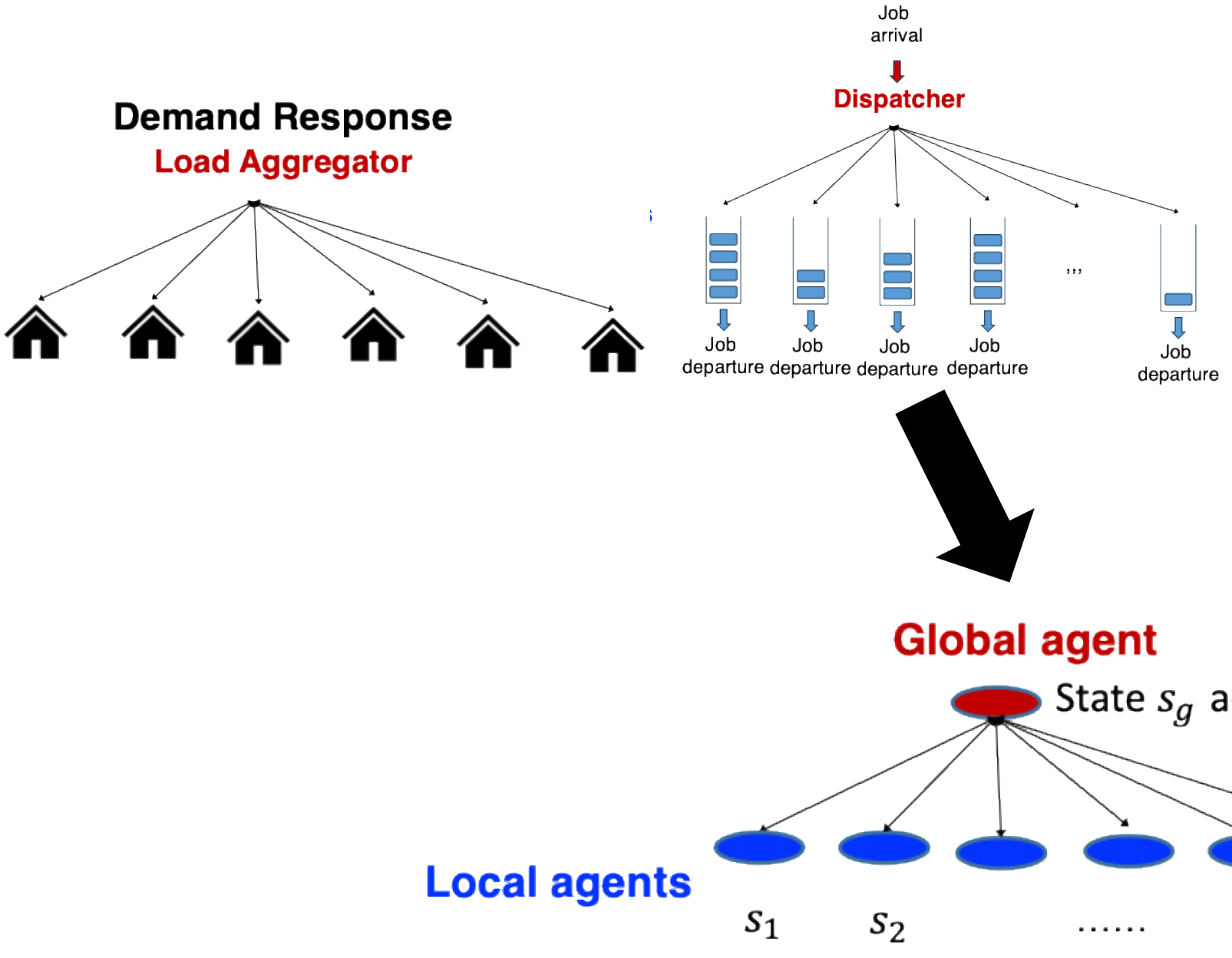


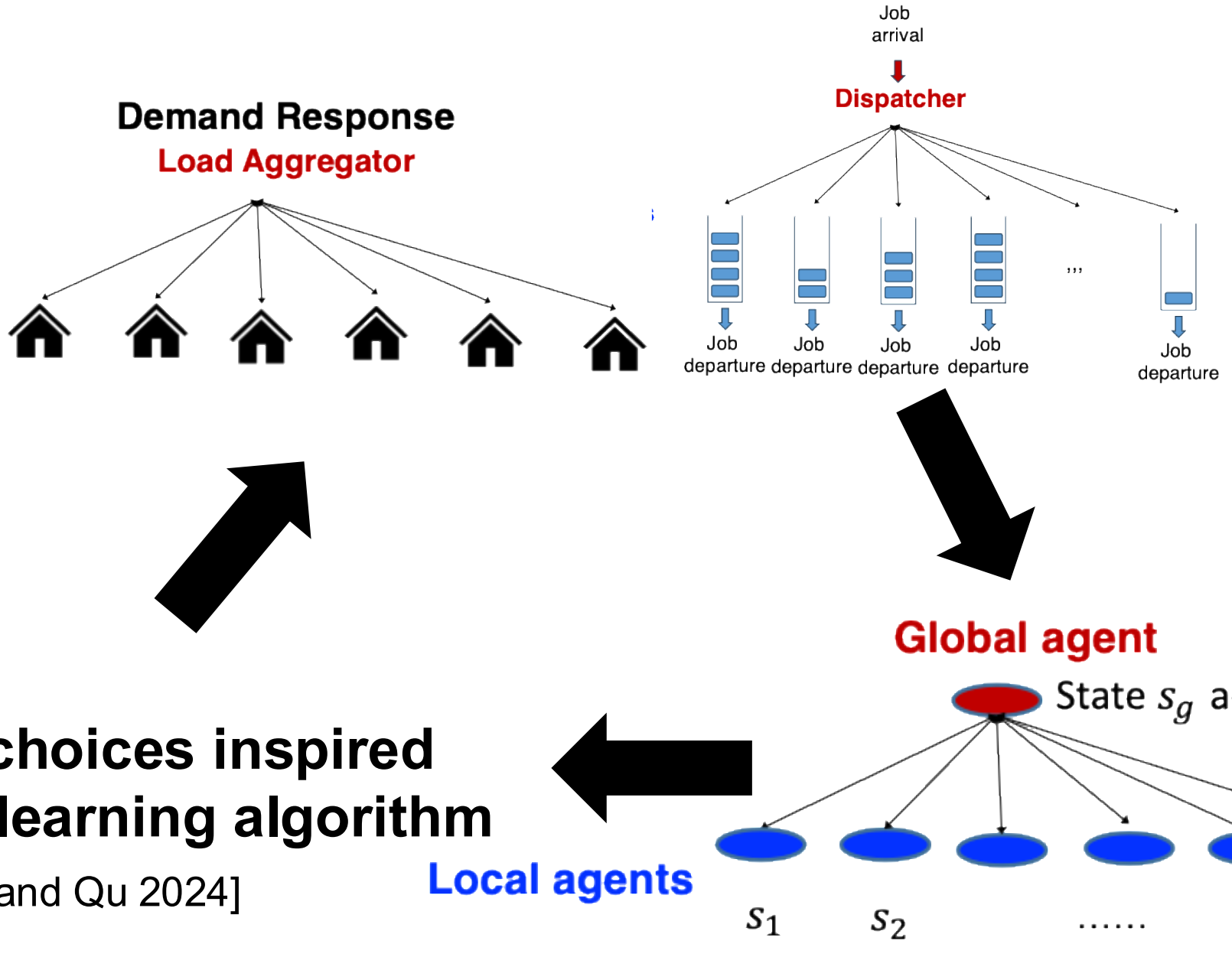
Demand Response

Load Aggregator



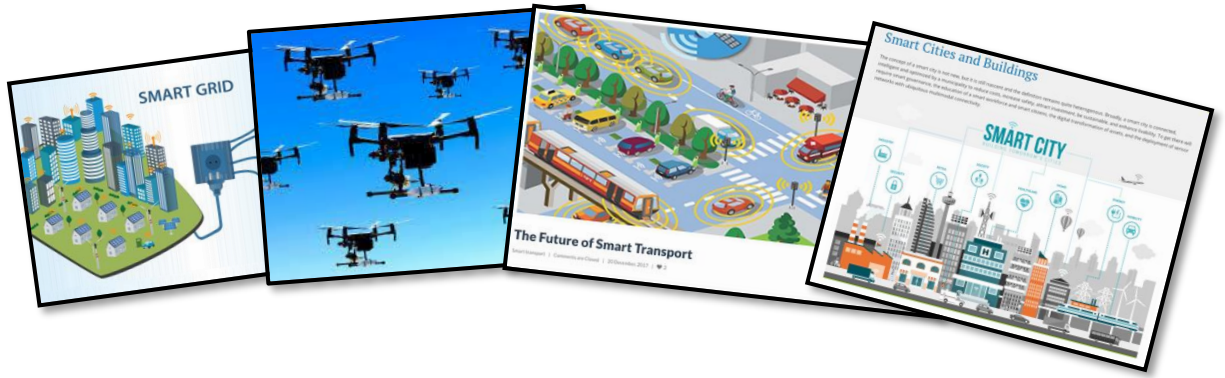



Demand Response
Load Aggregator
Job
arrival
Dispatcher
Job
departure
Job
departure
Job
departure
Job
departure
Job
departure
Global agent
State s_g action a_g
Local agents
s_1
s_2
.....
s_n



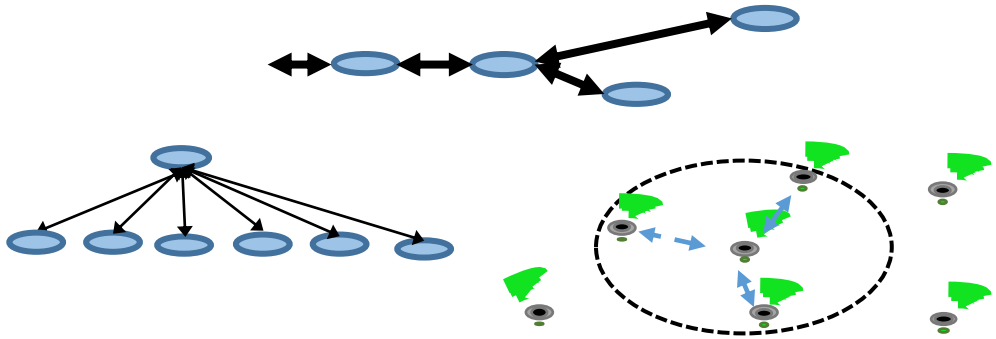
Power of k-choices inspired subsample Q-learning algorithm

[Anand and Qu 2024]



Apply to applications

Extract abstract structural property



$$x_{t+1} = f_{physics}(x_t, u_t)$$

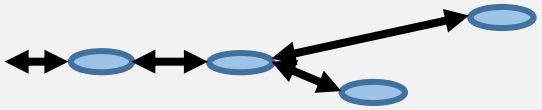
Exploit structure to obtain "positive" MARL theory

MARL Algorithm
with theoretical guarantee
that **avoids intractability**

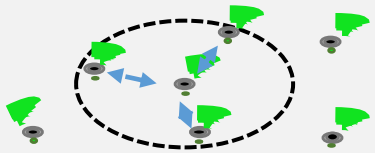
Today's Tutorial

Part 1: Background on reinforcement learning

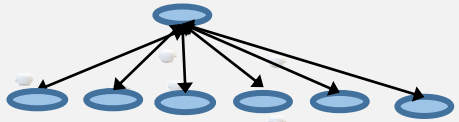
Part 2: Exploiting network topology structure



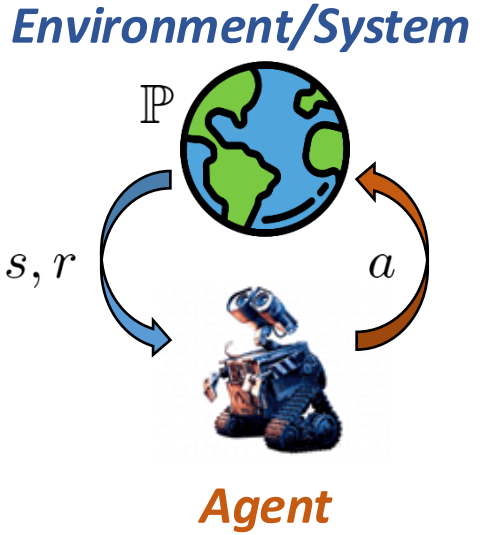
Part 3: Exploiting locality structure

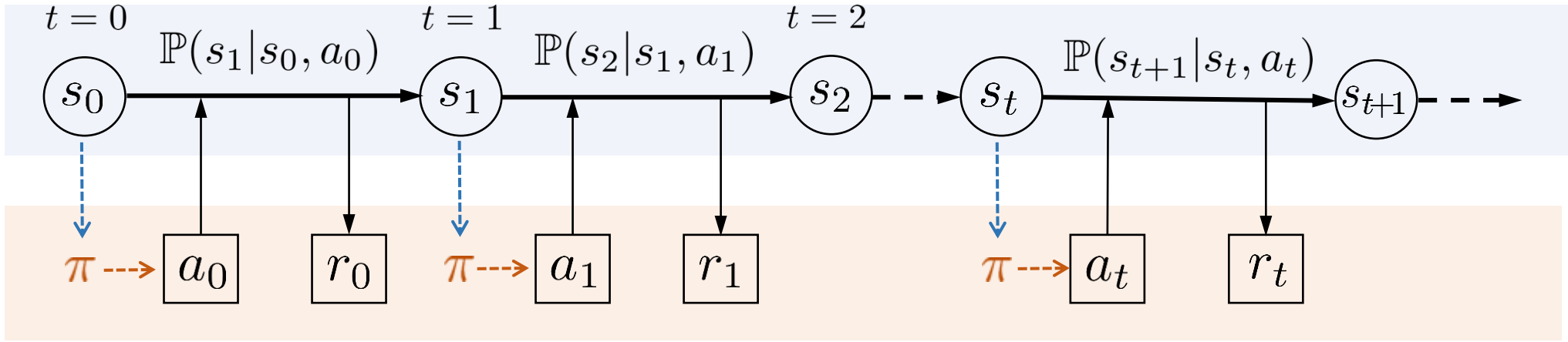


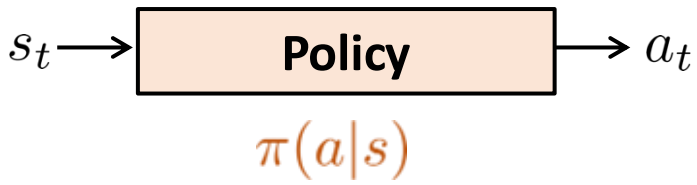
Part 4: Exploiting homogeneity structure



RL Background







Find an optimal policy $\pi^* \in \arg\max_{\pi} J(\pi) = \mathbb{E}_{s_0 \sim \mu_0} \mathbb{E}_{\pi} \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$

without knowing the model a priori

Q Functions

- **Q-Function** $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for a given policy π is defined as $Q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\right]$

- **Q-Function** satisfies the **Bellman Equation**: $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

solve $Q_\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a), a' \sim \pi(\cdot|s')} Q_\pi(s', a')$

Replace with samples

$$\hat{Q}_\pi(s_t, a_t) \leftarrow (1 - \alpha_t)\hat{Q}_\pi(s_t, a_t) + \alpha_t \left[r(s_t, a_t) + \gamma \hat{Q}_\pi(s_{t+1}, a_{t+1})\right]$$

Learning rate

Q Functions

- **Q-Function** $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for a given policy π is defined as $Q_\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\Big]$

Temporal Difference (TD) Learning

(Policy Evaluation)

$$\hat{Q}_\pi(s_t, a_t) \leftarrow (1 - \alpha_t)\hat{Q}_\pi(s_t, a_t) + \alpha_t \left[r(s_t, a_t) + \gamma \hat{Q}_\pi(s_{t+1}, a_{t+1})\right]$$

Policy based RL

- **Q-Function** $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for a given policy π is defined as $Q_\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) | s_0 = s, a_0 = a\Big]$

Policy Gradient Theorem

π_θ parameterized by θ

$$\nabla J(\theta) = \mathbb{E}_{s\sim d_\pi(s), a\sim\pi(\cdot|s)} \nabla_\theta \log \pi_\theta(a|s) Q_\pi(s,a)$$

Replace with samples

Replace with estimated Q from TD-learning

$$\hat{\theta} \leftarrow \hat{\theta} + \eta \nabla_\theta \log \pi_{\hat{\theta}}(a_t|s_t) \hat{Q}_{\pi_{\hat{\theta}}}(s_t,a_t)$$

Policy based RL

- **Q-Function** $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for a given policy π is defined as $Q_\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\Big]$

Actor-Critic Algorithm

Critic (policy evaluation)

$$\hat{Q}_{\pi_{\hat{\theta}}}(s_t, a_t) \leftarrow (1-\alpha_t)\hat{Q}_{\pi_{\hat{\theta}}}(s_t, a_t) + \alpha_t \left[r(s_t, a_t) + \gamma \hat{Q}_{\pi_{\hat{\theta}}}(s_{t+1}, a_{t+1})\right]$$

Actor (policy improvement)

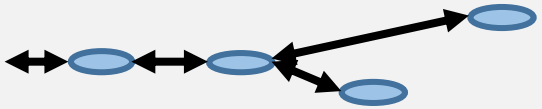
$$\hat{\theta} \leftarrow \hat{\theta} + \eta \nabla_\theta \log \pi_{\hat{\theta}}(a_t|s_t) \hat{Q}_{\pi_{\hat{\theta}}}(s_t, a_t)$$

Today's Tutorial

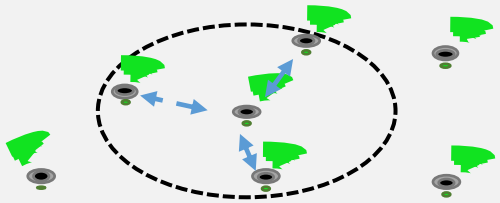
Part 1: Background on reinforcement learning

Up Next

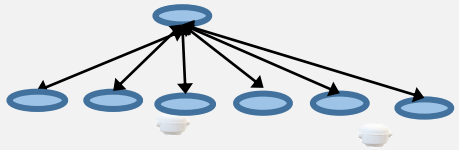
Part 2: Exploiting network topology structure

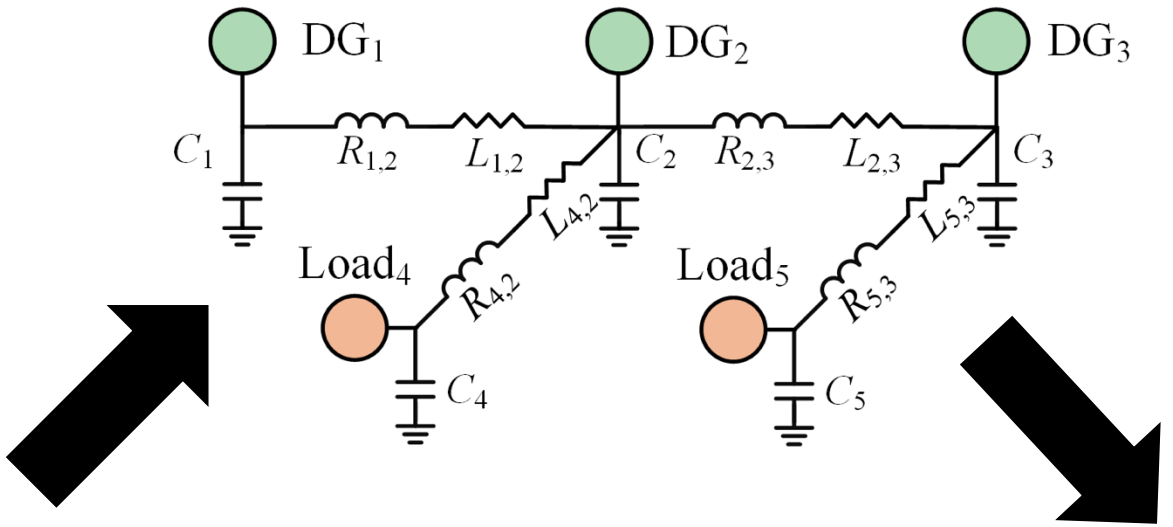


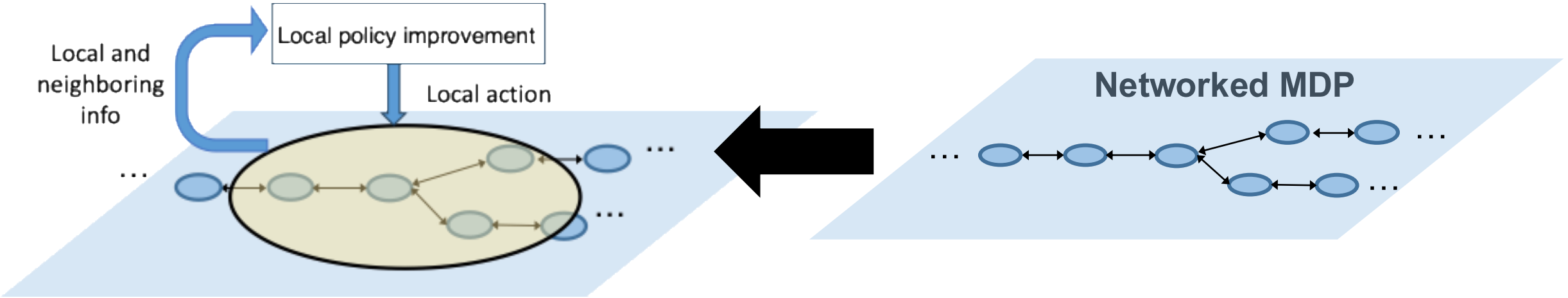
Part 3: Exploiting locality structure

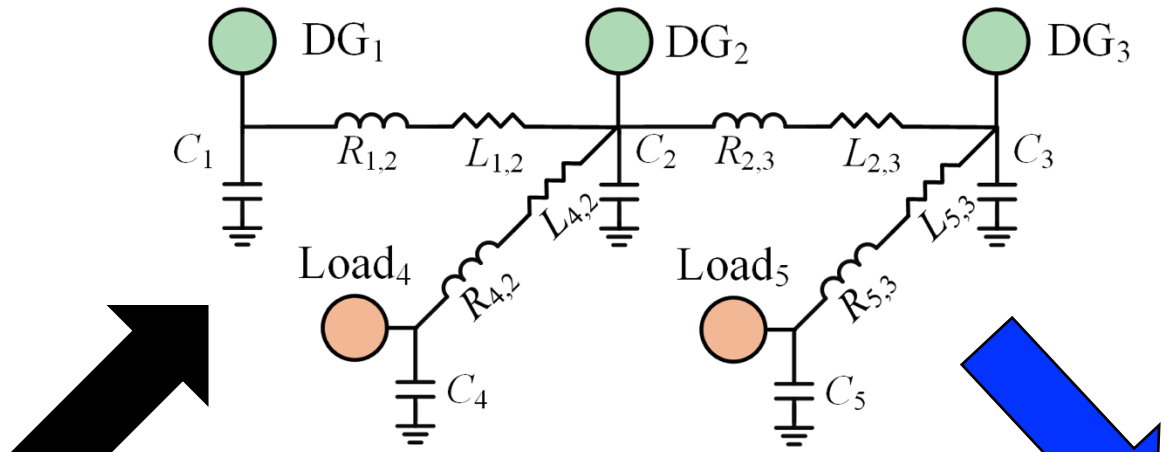


Part 4: Exploiting homogeneity structure

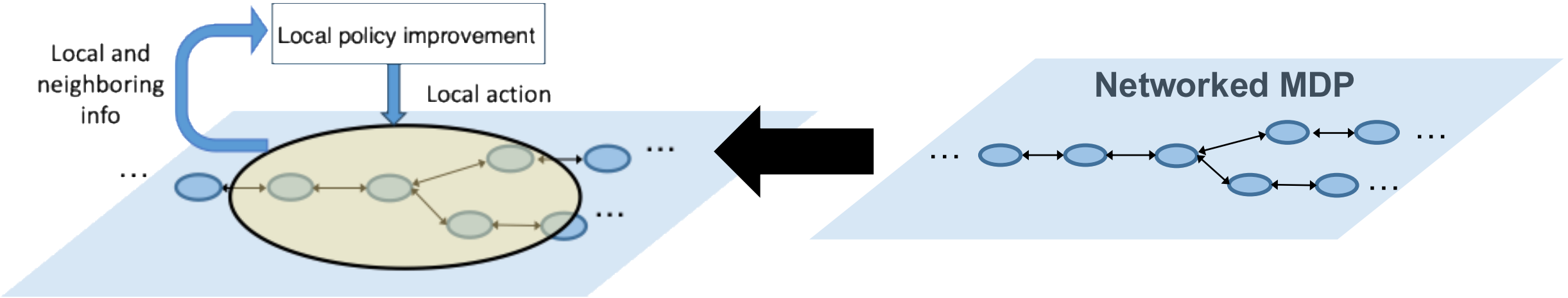



DG1
DG2
DG3
C1
R1,2
L1,2
C2
R2,3
L2,3
C3
L4,2
L5,3
Load4
R4,2
Load5
R5,3
C4
C5


Local and neighboring info
Local policy improvement
Local action
Networked MDP

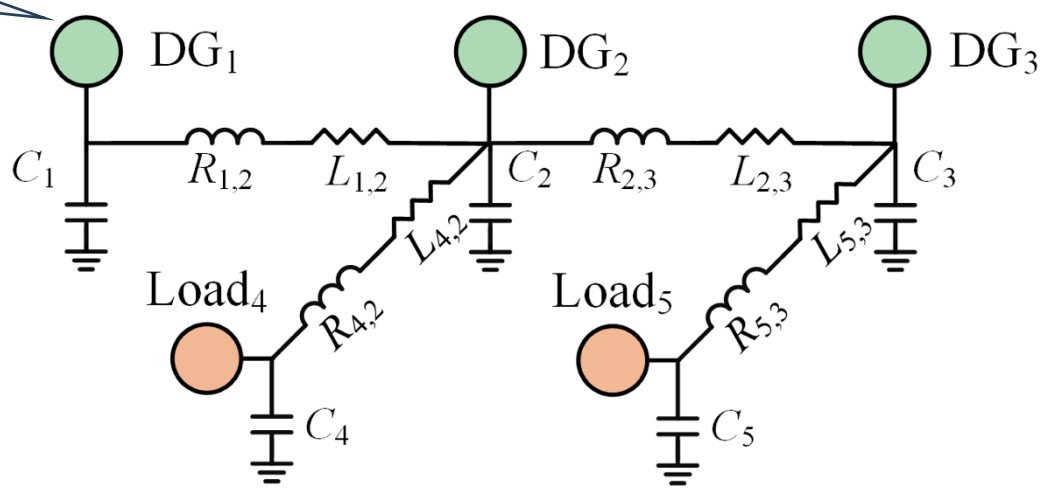

DG1
DG2
DG3
C1
R1,2
L1,2
C2
R2,3
L2,3
C3
L4,2
L5,3
Load4
R4,2
Load5
R5,3
C4
C5

UP NEXT

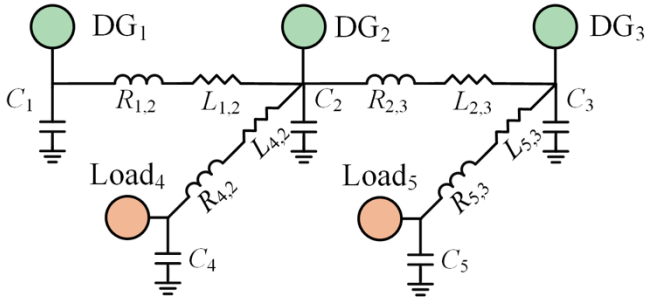

Local and neighboring info
Local policy improvement
Local action
Networked MDP

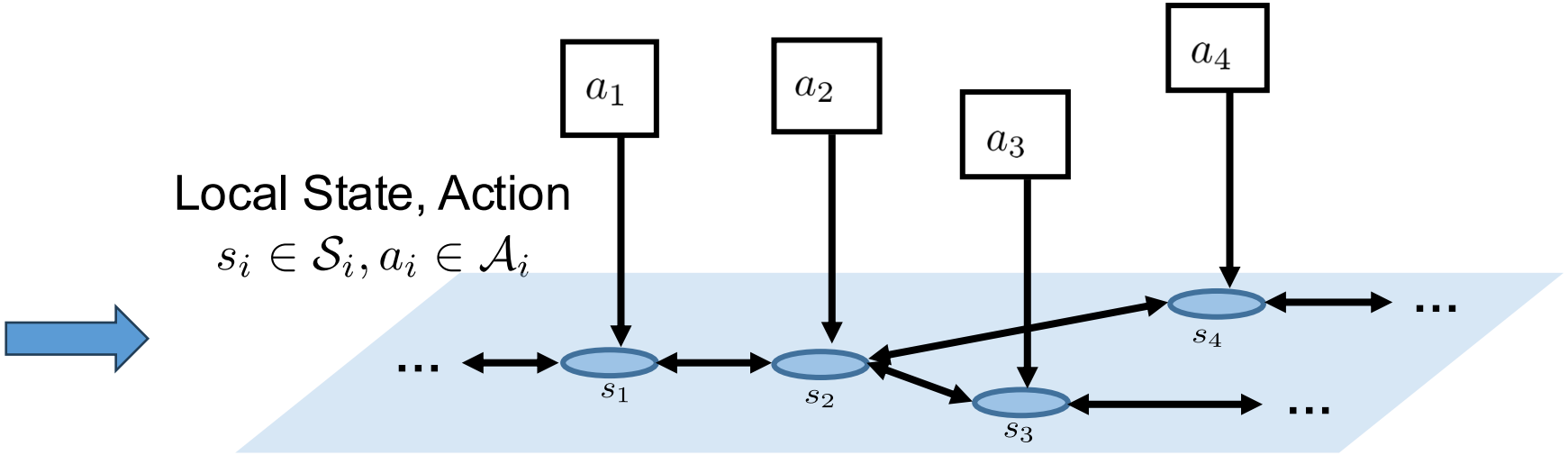
state: voltage, freq, …
action: reference set point…

Power Networks
[Chen et. al. 2022]



Markov Decision Process (MDP) over network

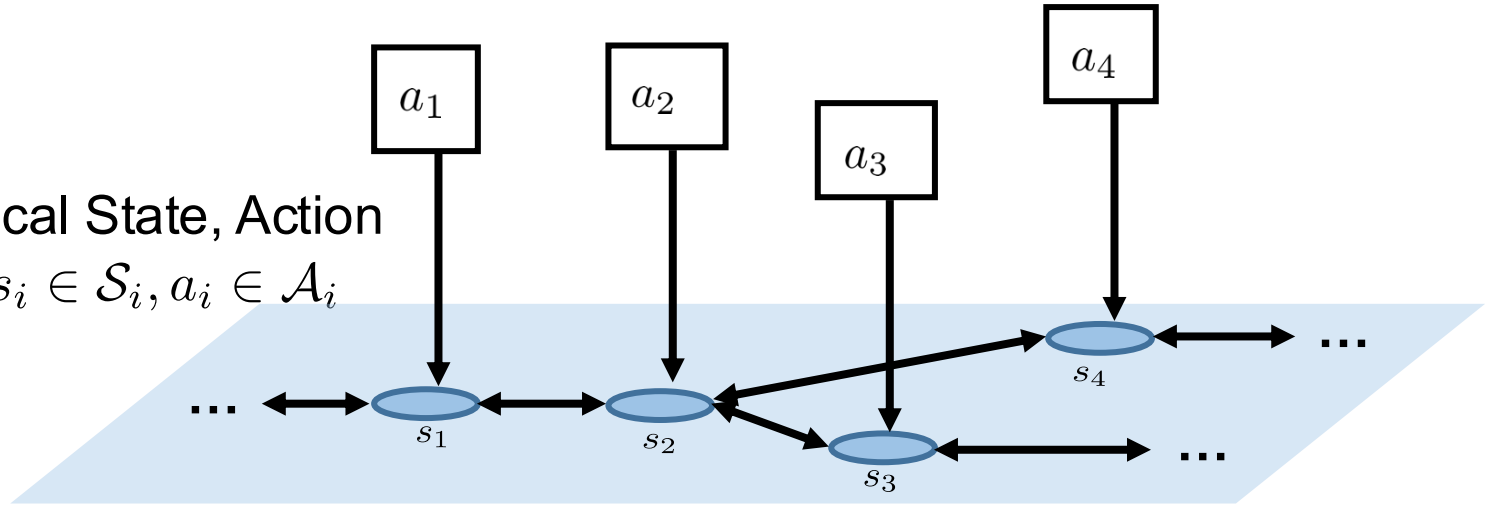




State $s = (s_1, \ldots, s_n) \in S_1 \times S_2 \times \cdots \times S_n := \mathcal{S}$

Action $a = (a_1, \ldots, a_n) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n := \mathcal{A}$

Markov Decision Process (MDP) over network



action $a = (a_1, \ldots, a_n)$

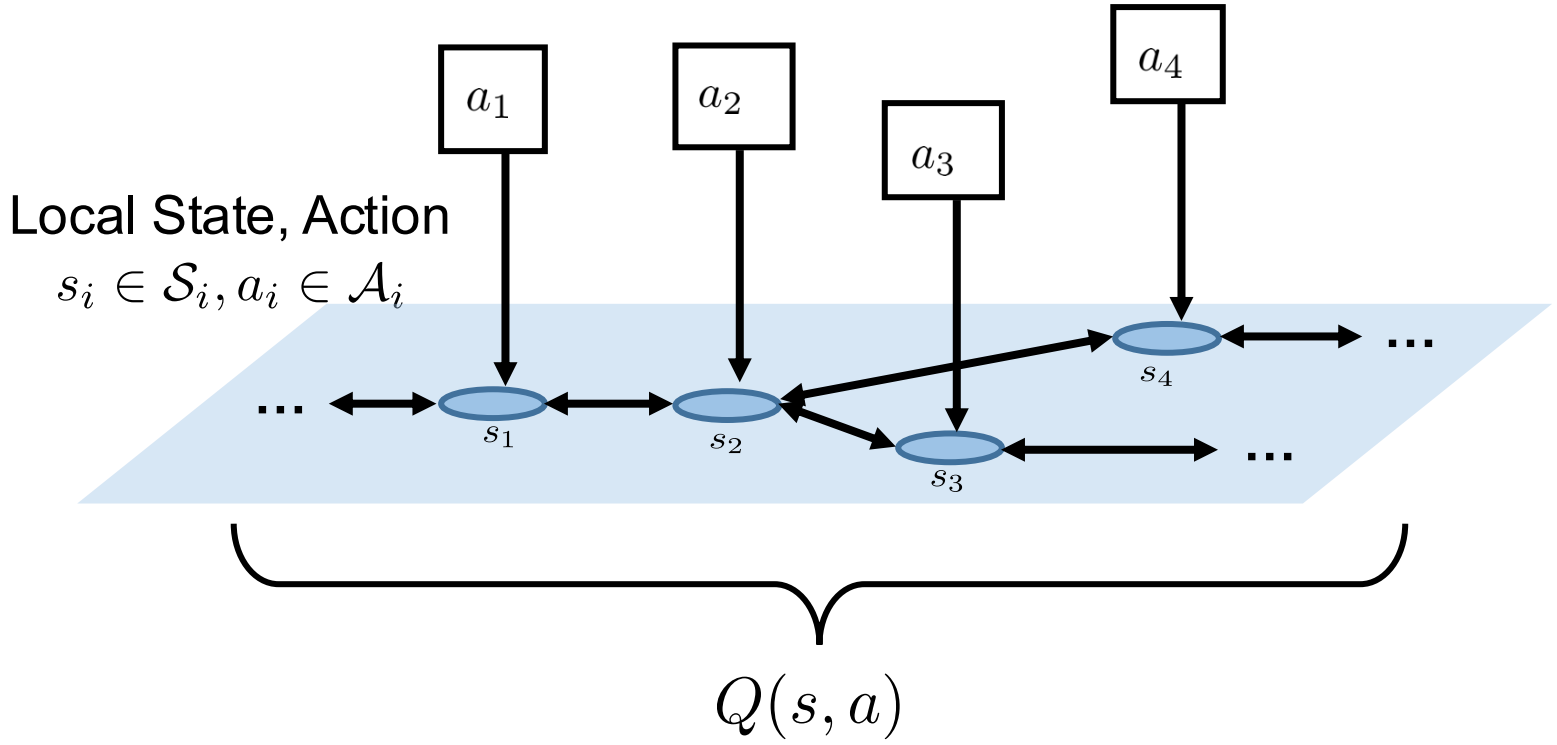
state $s = (s_1, \ldots, s_n)$

$Q(s,a)$	(0,0,….,0)	(0,0,…,1)	…	(1,1,…,1)
(0,0,…,0)				
(0,0,…,1)				
…				
(1,1,….,1)				

. . .

Time/space complexity exponential in n!

Markov Decision Process (MDP) over network



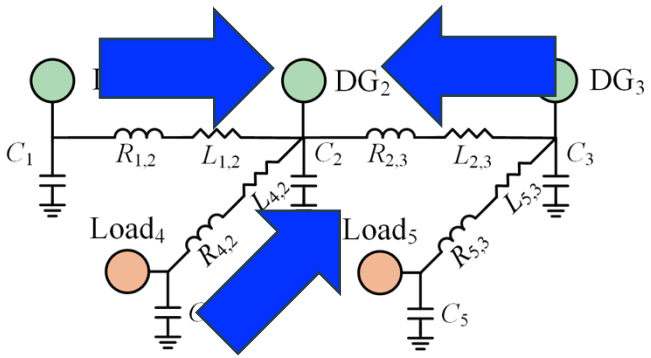
Centralized training is needed

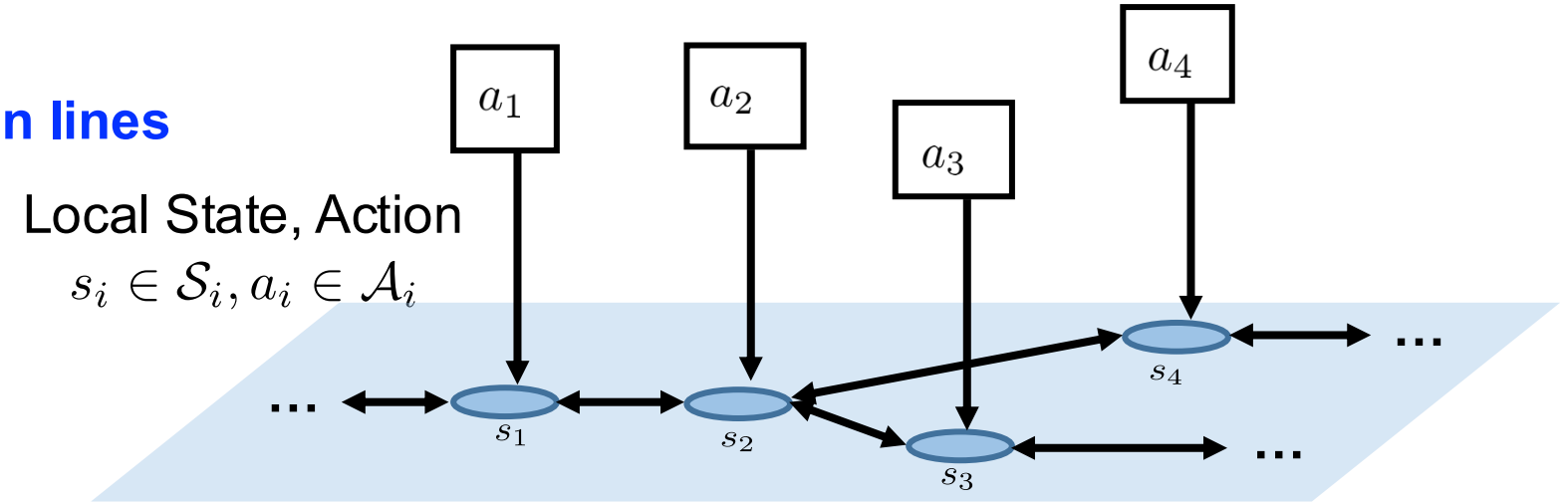
[Lowe 2017]

state/action/reward observation of all agents required to train Q functio

Markov Decision Process (MDP) over network

Power flows along the transmission lines



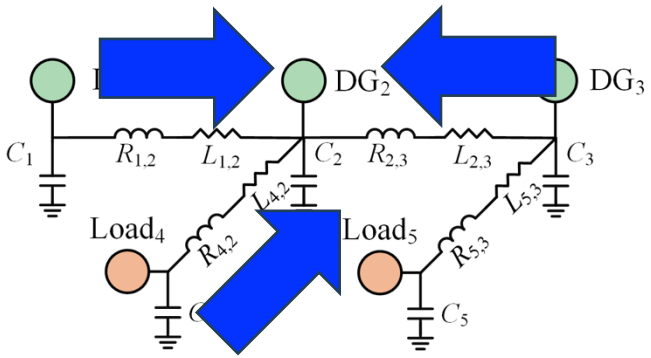


Is there any structural properties we can leverage

to overcome curse of dimensionality and avoid centralized training

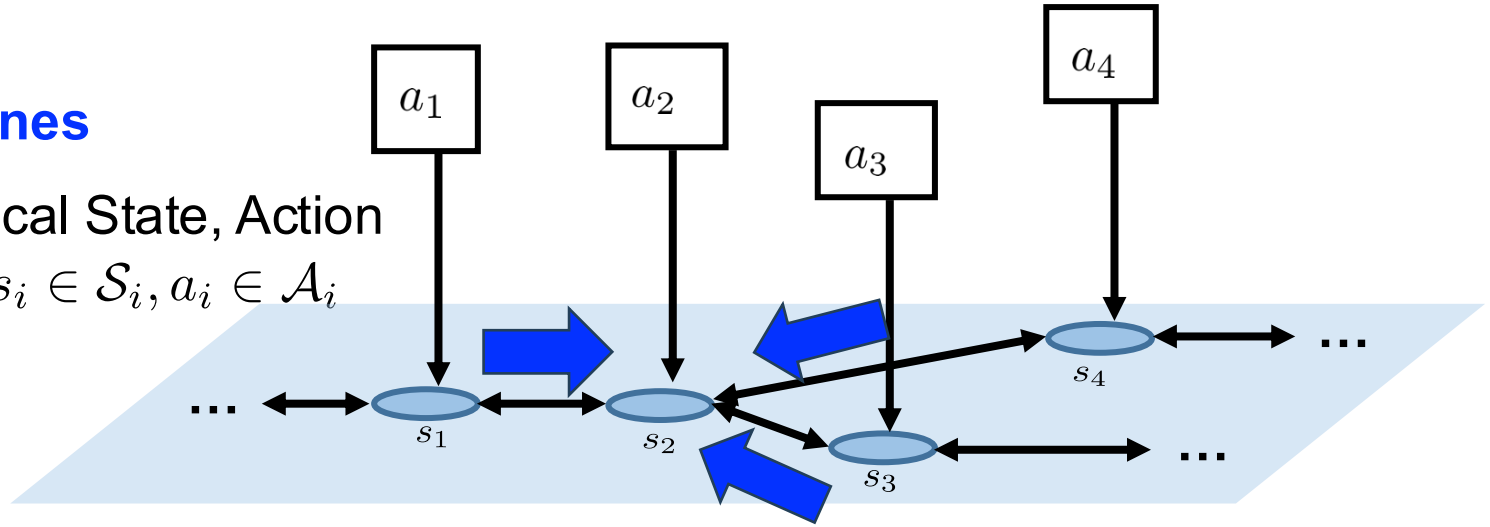
Markov Decision Process (MDP) over network

Power flows along the transmission lines



Local State, Action

$s_i \in \mathcal{S}_i, a_i \in \mathcal{A}_i$



State Transition $P(s(t+1)|s(t),a(t)) = \prod_{i=1}^{n} P_i(s_i(t+1)|s_{N_i}(t),a_i(t)).$

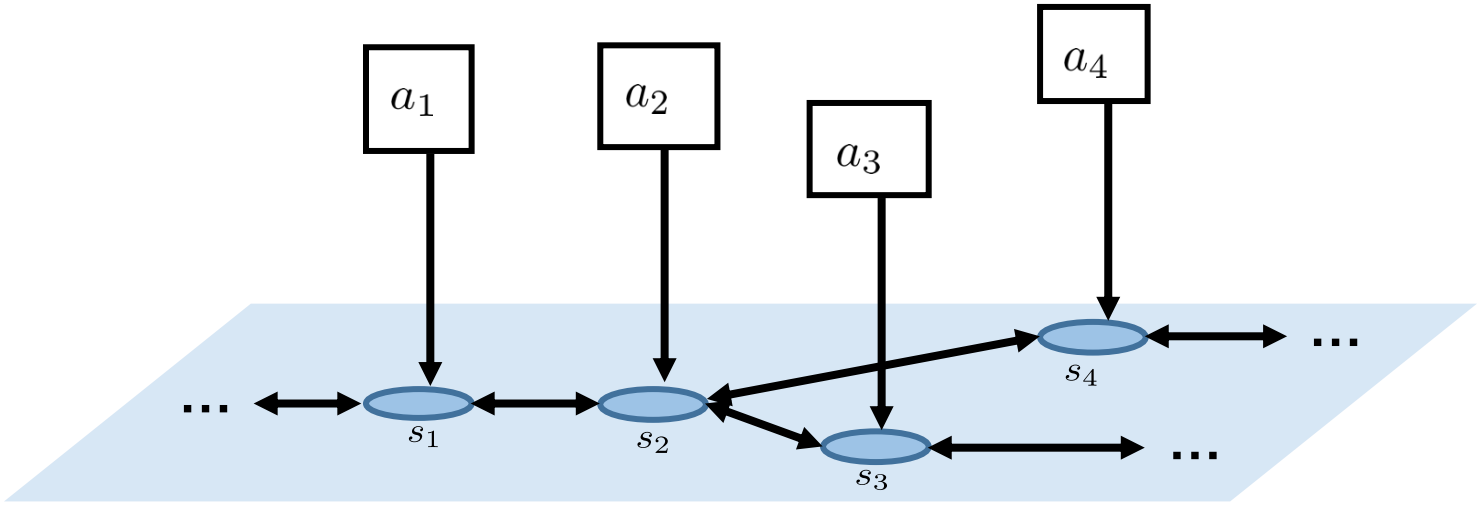
i's next state

Own state and Neighbor's state

Own action

Networked local interaction structure

Markov Decision Process (MDP) over network



State Transition $$P(s(t+1)|s(t),a(t)) = \prod_{i=1}^{n} P_i(s_i(t+1)|s_{N_i}(t),a_i(t)).$$

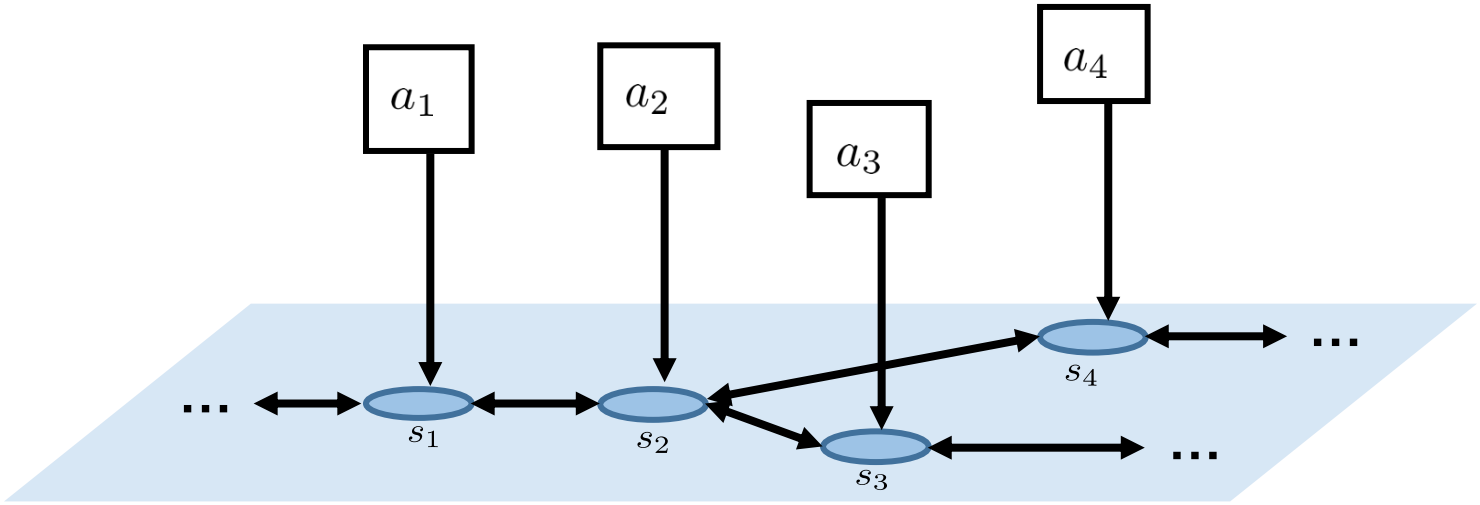
i's next state

Own state and Neighbor's state

Own action

Stage Reward $$r(s,a) = \frac{1}{n}\sum_{i=1}^{n} r_i(s_i,a_i)$$

Markov Decision Process (MDP) over network



Find decentralized policies to maximize objective.

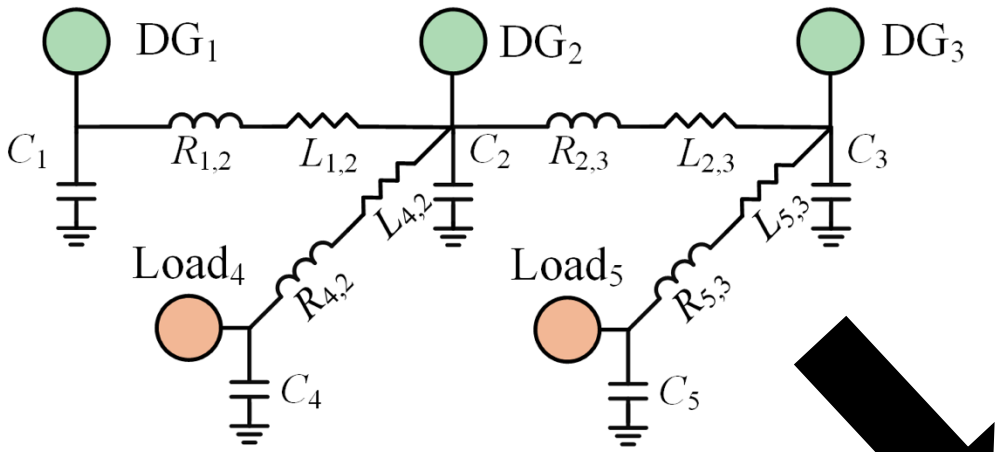
$$a_i(t) \sim \zeta_i(\cdot|s_i(t))$$

local action local state

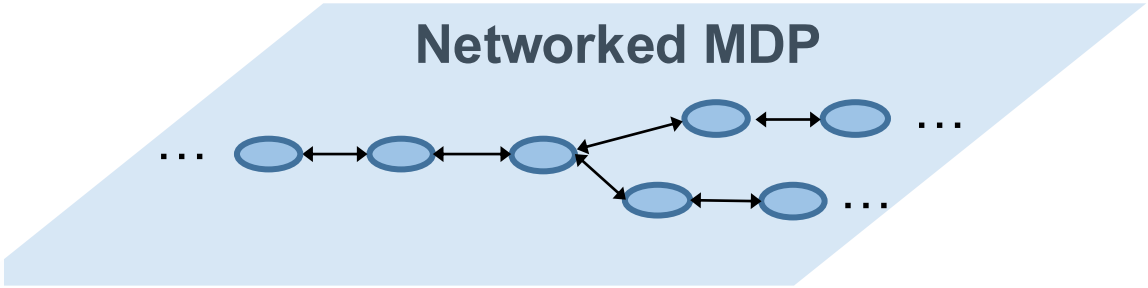
where $\zeta_i : \mathcal{S}_i \to \Delta(\mathcal{A}_i)$

$$\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s(t), a(t))|s(0) = s\Big]$$

Infinite horizon discounted reward

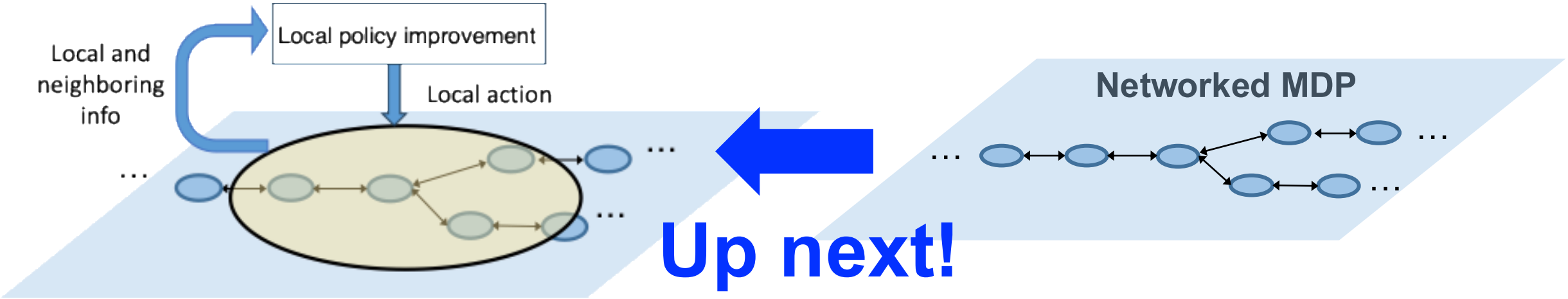

DG1
DG2
DG3
C1
R1,2
L1,2
C2
R2,3
L2,3
C3
L4,2
L5,3
Load4
R4,2
Load5
R5,3
C4
C5

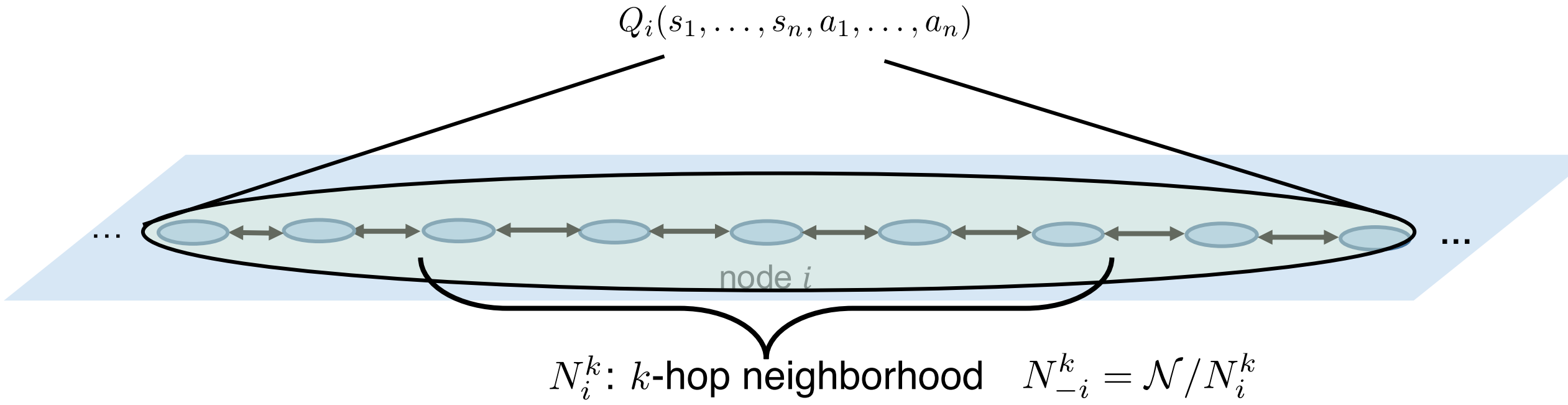
Localized MARL Algo.
Local policy improvement
Local and neighboring info
Local action


Networked MDP

Can we exploit network structure to achieve

- Scalable RL that breaks exponential scaling in n
- Distributed RL that avoids centralized training





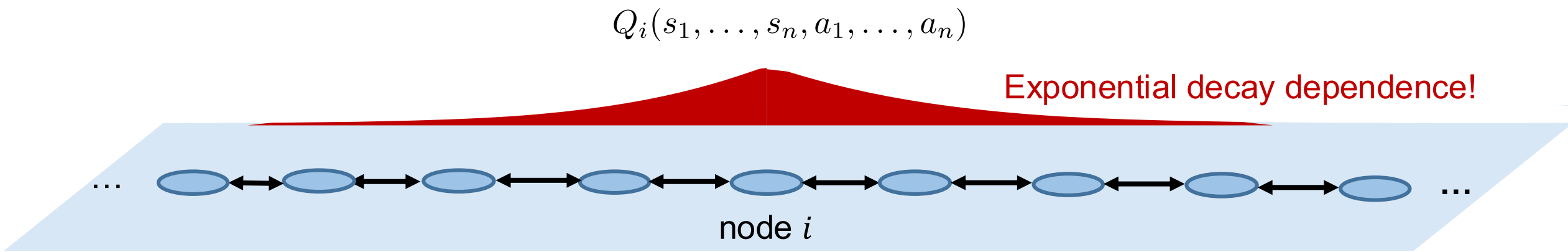
Definition.* The (c, ρ)-exponential decay property holds if for integer k,

$$|Q_i(s_{N_i^k}, s_{N_{-i}^k}, a_{N_i^k}, a_{N_{-i}^k}) - Q_i(s_{N_i^k}, s'_{N_{-i}^k}, a_{N_i^k}, a'_{N_{-i}^k})| \leq c\rho^{k+1}$$

change state/action outside N_i^k

exponential decay

* See also [Gamarnik et al. 2013, 2014, Bamieh et al. 2002, Motee and Jadbabaie 2008]



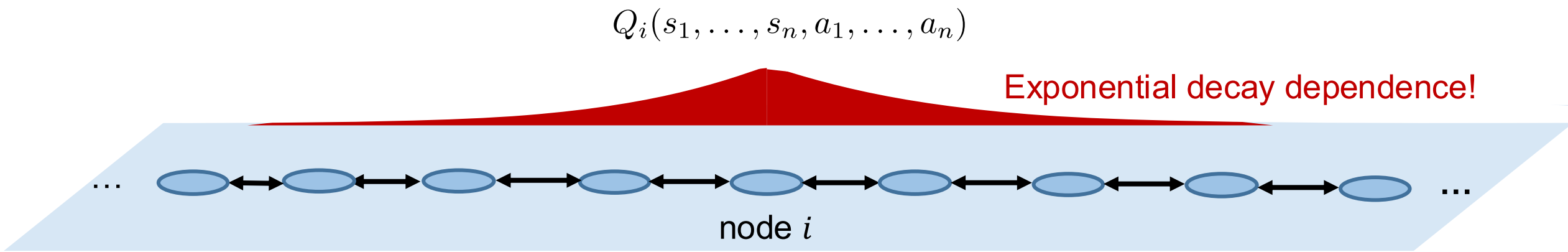
Definition.* The (c, ρ)-exponential decay property holds if for integer k,

$$|Q_i(s_{N_i^k}, s_{N_{-i}^k}, a_{N_i^k}, a_{N_{-i}^k}) - Q_i(s_{N_i^k}, s'_{N_{-i}^k}, a_{N_i^k}, a'_{N_{-i}^k})| \leq c\rho^{k+1}$$

change state/action outside N_i^k

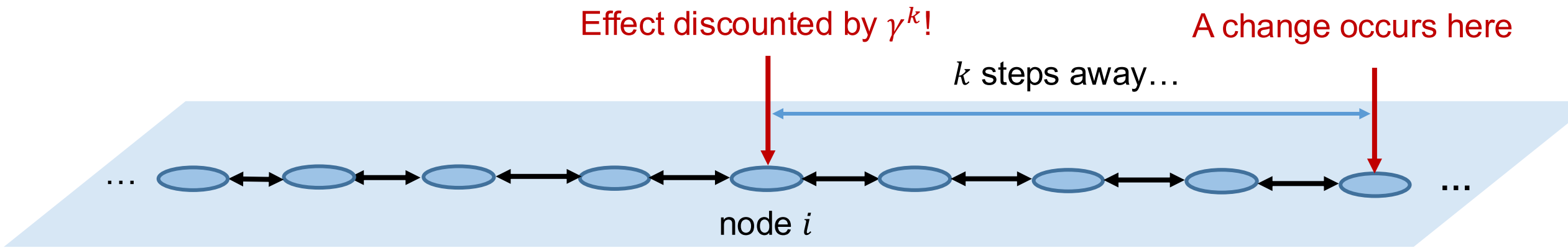
exponential decay

* See also [Gamarnik et al. 2013, 2014, Bamieh et al. 2002, Motee and Jadbabaie 2008]



Lemma [Qu, Wierman, Li 2022]:

If all rewards are bounded, then (c, ρ)-exponential decay property holds with $\rho \leq \gamma$.

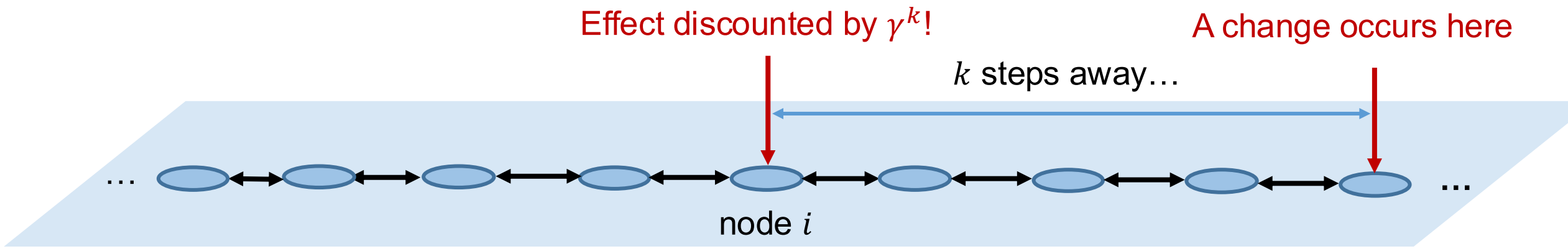


Proof.

$$|Q_i(s,a) - Q_i(s',a')|$$

$$\leq \sum_{t=0}^{\infty} \left| \gamma^t \mathbb{E}_{(s_i,a_i)\sim\pi_{t,i}} r_i(s_i,a_i) - \gamma^t \mathbb{E}_{(s_i,a_i)\sim\pi'_{t,i}} r_i(s_i,a_i) \right|$$

$$= \sum_{t=k+1}^{\infty} \left| \gamma^t \mathbb{E}_{(s_i,a_i)\sim\pi_{t,i}} r_i(s_i,a_i) - \gamma^t \mathbb{E}_{(s_i,a_i)\sim\pi'_{t,i}} r_i(s_i,a_i) \right| \leq \frac{\bar{r}}{1-\gamma}\gamma^{k+1}$$



Lemma [Qu, Wierman, Li 2022]:

If all rewards are bounded, then (c, ρ)-exponential decay property holds with $\rho \leq \gamma$.

When will the decay rate ρ strictly smaller than γ?

Interaction strength
how change of s_j affects transition of s_i

Bounded total interaction strength from neighbors
- Small pairwise interaction strength & small degree
- Consistent with results in combinatorial optimization [Lovasz Local Lemma] [Gamarnik 2014]

Theorem. Define interaction strength between i, j as

$$C_{ij} = \begin{cases} 0, & \text{if } j \notin N_i, \\ \sup_{s_{N_i/j}, a_i} \sup_{s_j, s_j'} \mathrm{TV}(P_i(\cdot|s_j, s_{N_i/j}, a_i), P_i(\cdot|s_j', s_{N_i/j}, a_i)), & \text{if } j \in N_i, j \neq i, \\ \sup_{s_{N_i/i}} \sup_{s_i, s_i', a_i, a_i'} \mathrm{TV}(P_i(\cdot|s_i, s_{N_i/i}, a_i), P_i(\cdot|s_i', s_{N_i/i}, a_i')), & \text{if } j = i, \end{cases}$$

where TV denotes total variation distance. If the rewards are bounded, and further

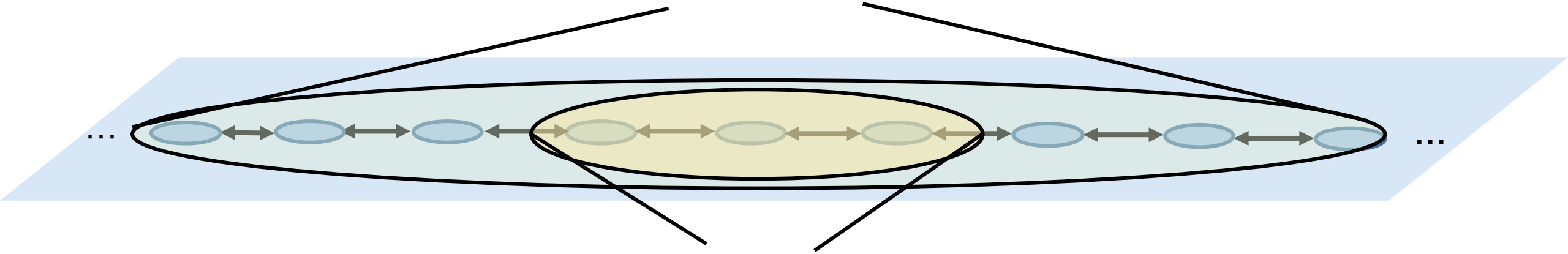
$$\forall i, \sum_{j \in N_i} C_{ij} \leq \mu < 1,$$

then, the exponential decay property holds **with decaying rate $\rho = \mu\gamma < \gamma$.**

When will the decay rate ρ strictly smaller than γ?

Full Q function:

$$Q_i(s_1, \ldots, s_n, a_1, \ldots, a_n)$$



$\hat{Q}_i(s_{N_i^k}, a_{N_i^k})$ only depends on k-hop neighborhood

$$\hat{Q}_i(s_{N_i^k}, a_{N_i^k}) = \sum_{s_{N_{-i}^k}, a_{N_{-i}^k}} w_i(s_{N_{-i}^k}, a_{N_{-i}^k}; s_{N_i^k}, a_{N_i^k}) Q_i(s_{N_i^k}, s_{N_{-i}^k}, a_{N_i^k}, a_{N_{-i}^k})$$

Arbitrary weights **Average out outside neighborhood**

Lemma (informal) Under the (c, ρ)-exponential decay property, then

$$\sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |Q_i(s, a) - \hat{Q}_i(s_{N_i^k}, a_{N_i^k})| \leq c\rho^{k+1}$$

Parameterized Policy: $a_i(t) \sim \zeta_i^{\theta_i}(\cdot|s_i(t))$ with $\theta = (\theta_1, \ldots, \theta_n)$

Objective Function: $J(\theta) = \mathbb{E}_{s\sim\pi_0}\mathbb{E}_\theta\left[\sum_{t=0}^{\infty}\gamma^t r(s(t), a(t))\middle| s(0) = s\right]$

Global summation

Exponentially large table

Full Policy Gradient $\nabla_{\theta_i} J(\theta) = \sum_{t=0}^{\infty}\gamma^t \mathbb{E}_{s\sim\pi_t^\theta, a\sim\zeta^\theta(a|s)} \nabla_{\theta_i} \log \zeta_i^{\theta_i}(a_i|s_i) \frac{1}{n}\sum_{j=1}^{n} Q_j^\theta(s,a)$

Truncated Policy Gradient $h_i(\theta) = \sum_{t=0}^{\infty}\gamma^t \mathbb{E}_{s\sim\pi_t^\theta, a\sim\zeta^\theta(a|s)} \nabla_{\theta_i} \log \zeta_i^{\theta_i}(a_i|s_i) \frac{1}{n}\sum_{j\in N_i^k} \hat{Q}_j^\theta(s_{N_j^k}, a_{N_j^k})$

Local summation

Much smaller table

Actor-Critic Method

Actor Outer Loop: $m = 0,1,\ldots,M$

Critic Inner Loop: $t = 0,1,\ldots,T$

Sample state, action reward

Temporal Difference (TD) update for **full Q function**

Estimate **full policy gradient**

Gradient Ascent

~~Actor-Critic Method~~ Scalable Actor-Critic Method

Actor Outer Loop: $m = 0,1,\ldots,M$

Critic Inner Loop: $t = 0,1,\ldots,T$

Sample state, action reward

Temporal Difference (TD) update for ~~full Q function~~ **truncated Q function**

Estimate ~~full policy gradient~~ **truncated policy gradient**

Gradient Ascent

Scalable Actor-Critic Method

Actor Outer Loop: $m = 0,1,\ldots,M$

Critic Inner Loop: $t = 0,1,\ldots,T$

Get state $s_i(t)$, take action $a_i(t) \sim \zeta_i^{\theta_i(m)}(\cdot|s_i(t))$, get reward $r_i(t)$.

$\hat{Q}_i(s_{N_i^k}(t-1), a_{N_i^k}(t-1)) \leftarrow$ //TD-update for truncated Q function

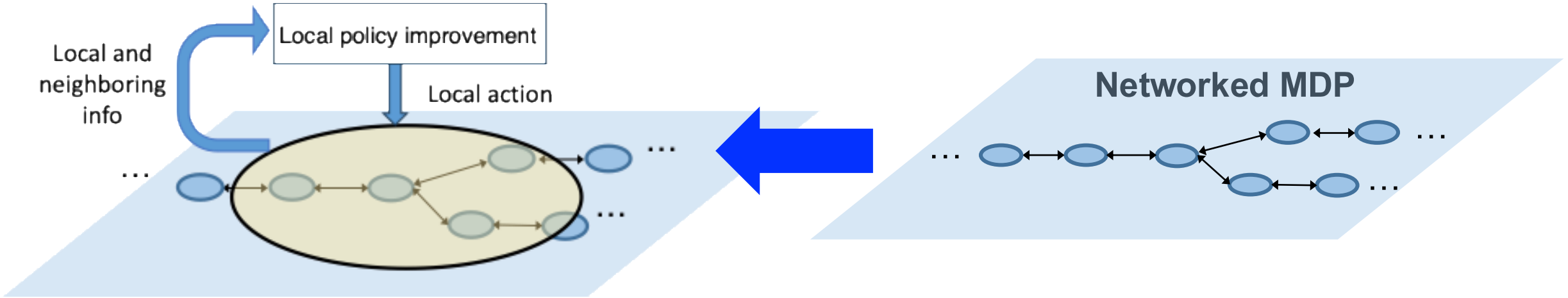
$$(1-\alpha_{t-1})\hat{Q}_i(s_{N_i^k}(t-1), a_{N_i^k}(t-1)) + \alpha_{t-1}(r_i(t-1) + \gamma \hat{Q}_i(s_{N_i^k}(t), a_{N_i^k}(t)))$$

$$\hat{g}_i(m) = \sum_{t=0}^{T} \gamma^t \nabla_{\theta_i} \log \zeta_i^{\theta_i(m)}(a_i(t)|s_i(t)) \frac{1}{n} \sum_{j \in N_i^k} \hat{Q}_j(s_{N_j^k}(t), a_{N_j^k}(t))$$ // truncated policy gradient

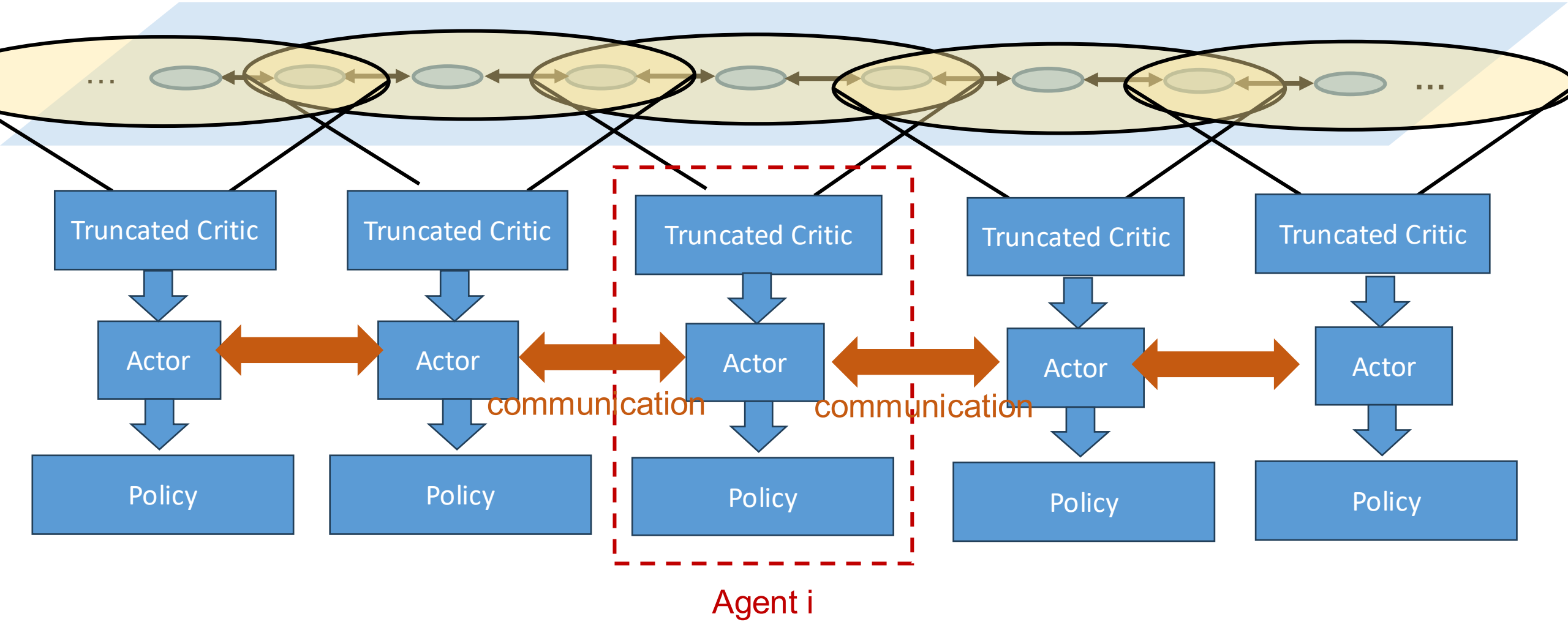
$$\theta_i(m+1) = \theta_i(m) + \eta_m \hat{g}_i(m) \text{ with } \eta_m = \frac{\eta}{\sqrt{m+1}}$$ // gradient ascent

Can we exploit network structure to achieve

- Scalable RL that breaks the curse of dimensionality
- Distributed RL that avoids centralized communication **Yes**

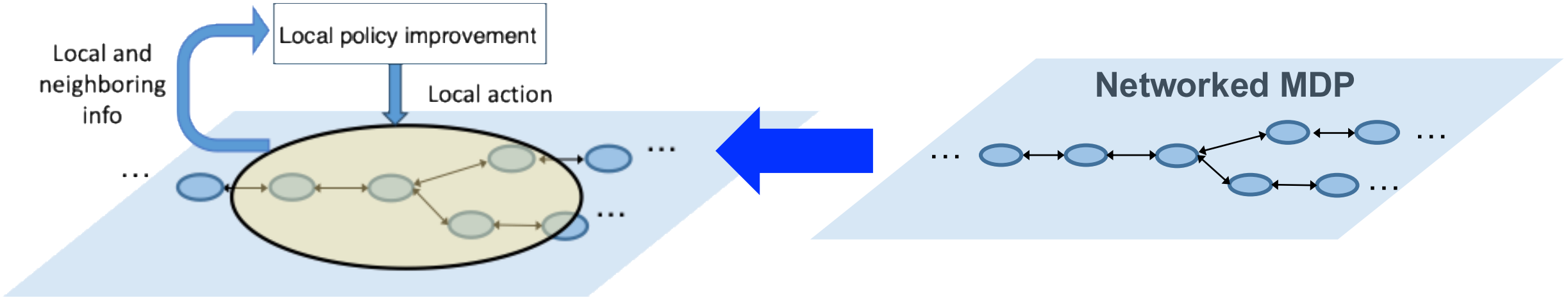


Our approach: Distributed Training, Decentralized Execution



Can we exploit network structure to achieve

- Scalable RL that breaks the curse of dimensionality ?
- Distributed RL that avoids centralized communication Yes



Optimality Guarantee

Theorem [Qu, Wierman, Li 2022]. Under some assumptions, with high probability,

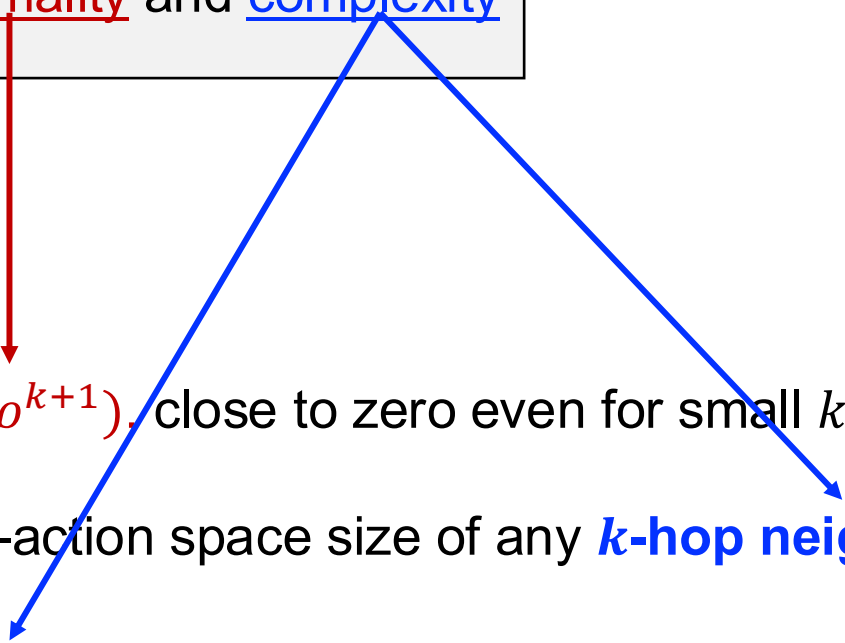
$$\frac{\sum_{m=0}^{M} \eta_m \|\nabla J(\theta(m))\|^2}{\sum_{m=0}^{M} \eta_m} \leq \underbrace{\tilde{O}\left(\frac{\text{poly}_1}{\sqrt{M+1}}\right)}_{\text{Converges to zero}} + \underbrace{O(\rho^{k+1})}_{:=\varepsilon_k \text{ steady-state error due to truncation}}$$

when the inner-loop length $T \geq \tilde{\Omega}\left(\frac{1}{\epsilon_k^2}\text{poly}_2\right)$

- Reaches **steady state error of** $\varepsilon_k = O(\rho^{k+1})$, close to zero even for small k
- Complexity scales with the largest state-action space size of any ***k*-hop neighborhood**
- Communication with ***k*-hop neighborhoods** required during training

Optimality Guarantee

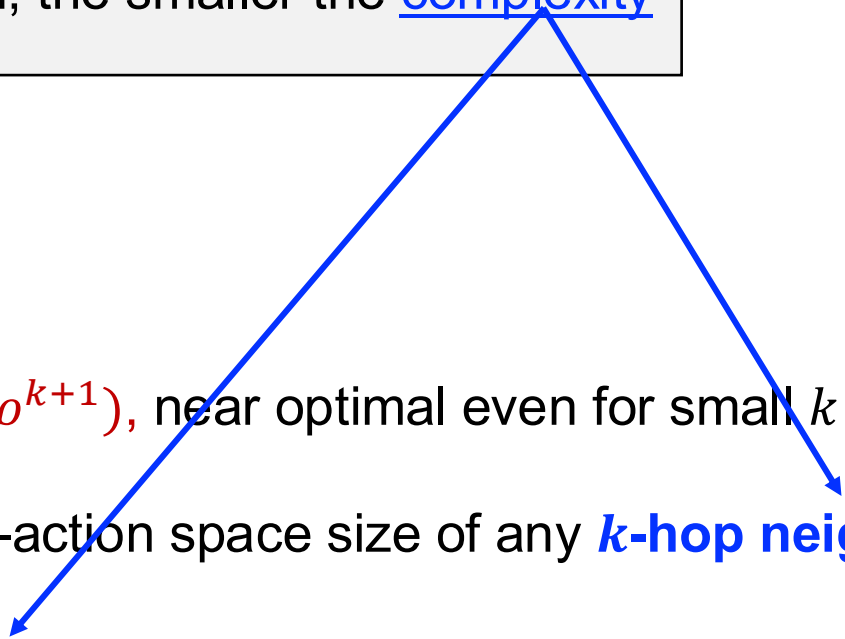
Role of k?
Trades off between optimality and complexity



- Reaches **steady state error of $\varepsilon_k = O(\rho^{k+1})$**, close to zero even for small k
- Complexity scales with the largest state-action space size of any **k-hop neighborhood**
- Communication with **k-hop neighborhoods** required during training

Optimality Guarantee

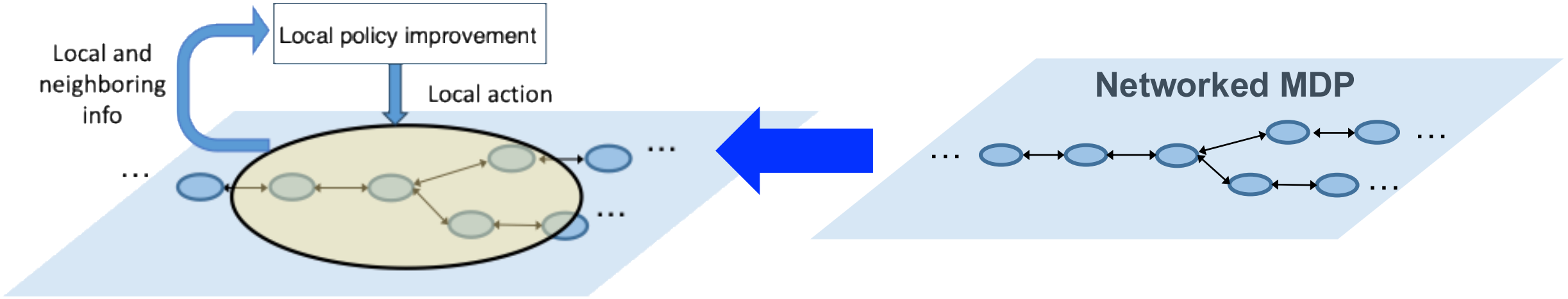
Role of graph?
Fixing k, the sparser the graph, the smaller the complexity



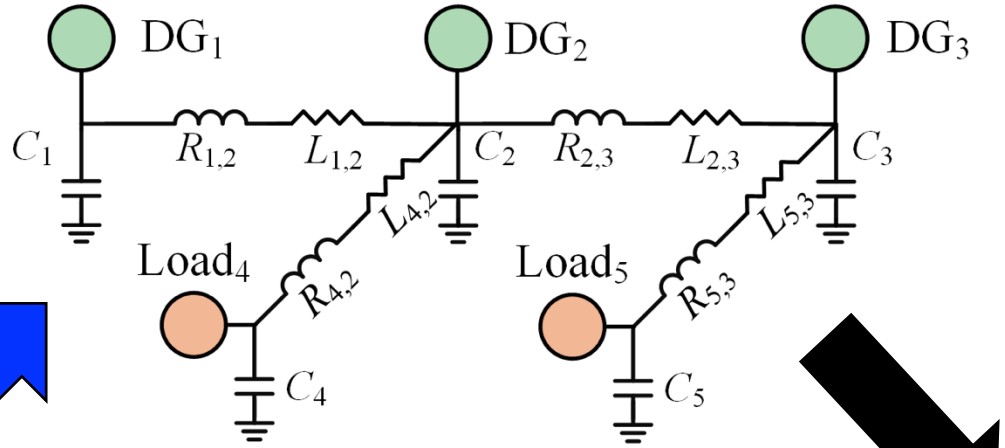
- Reaches **steady state error of** $\varepsilon_k = O(\rho^{k+1})$, near optimal even for small k
- Complexity scales with the largest state-action space size of any **k-hop neighborhood**
- Communication with **k-hop neighborhoods** required during training

Can we exploit network structure to achieve

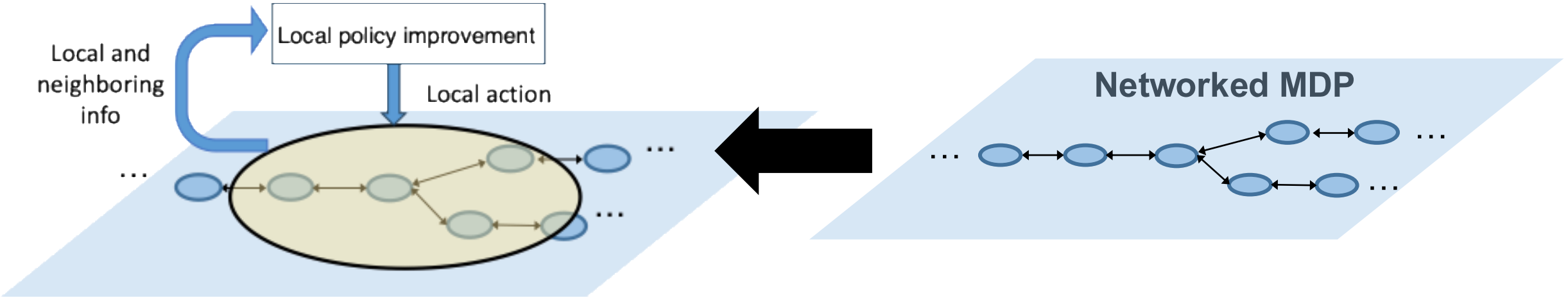
- Scalable RL that breaks the curse of dimensionality
- Distributed RL that avoids centralized communication



Up next!

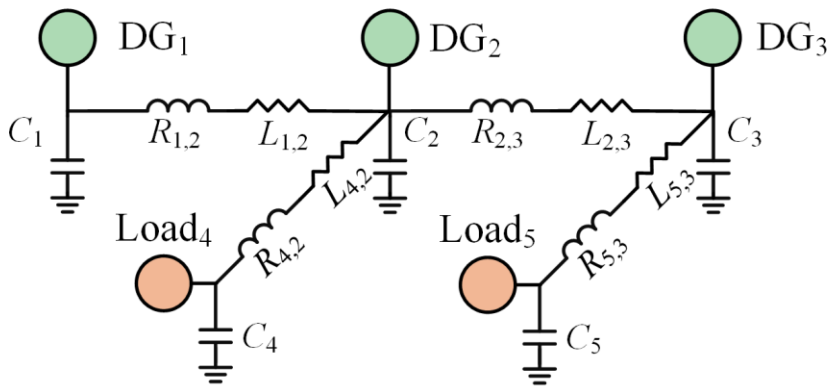


Localized MARL Algo.

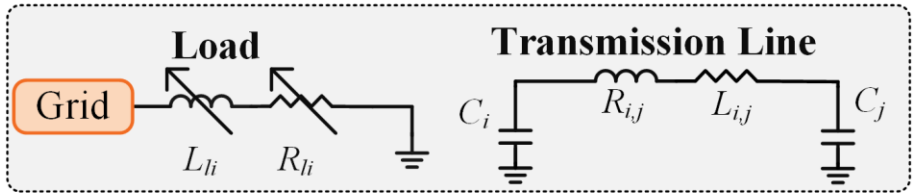


Microgrid Voltage Control

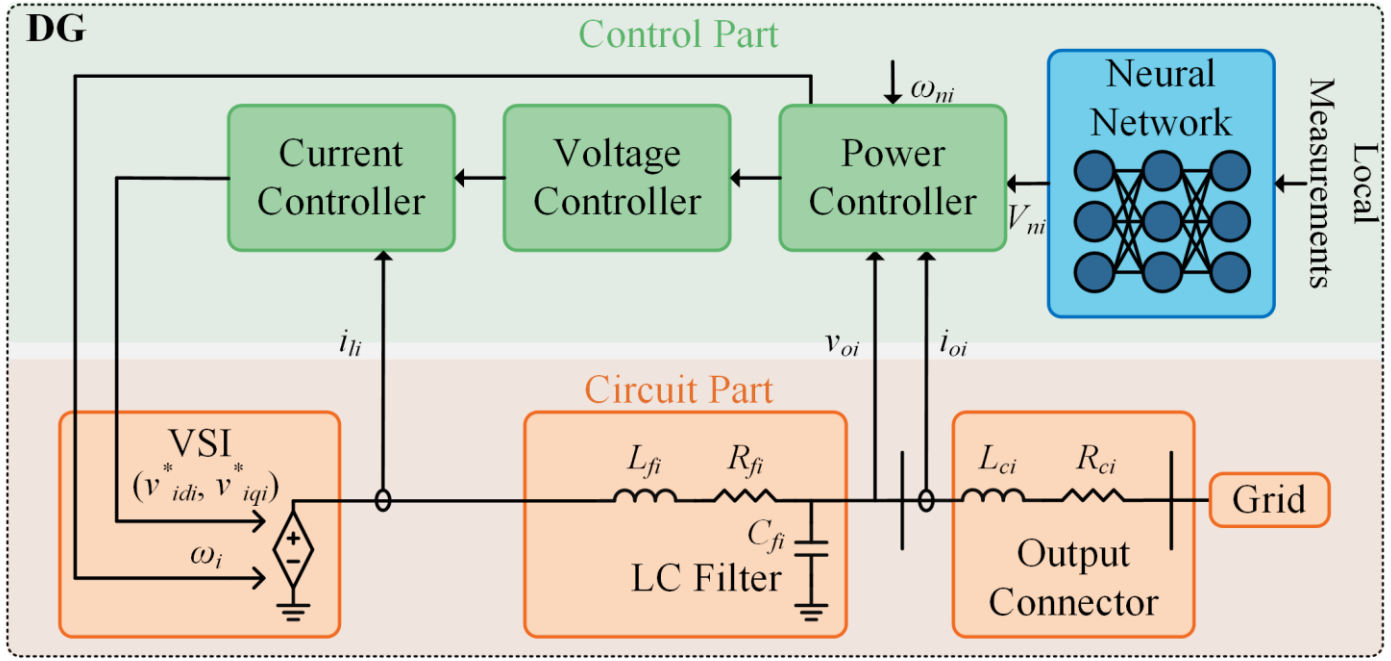
[Xu and Qu 2023]



(a) Diagram of a Microgrid



(b) Load and transmission line models



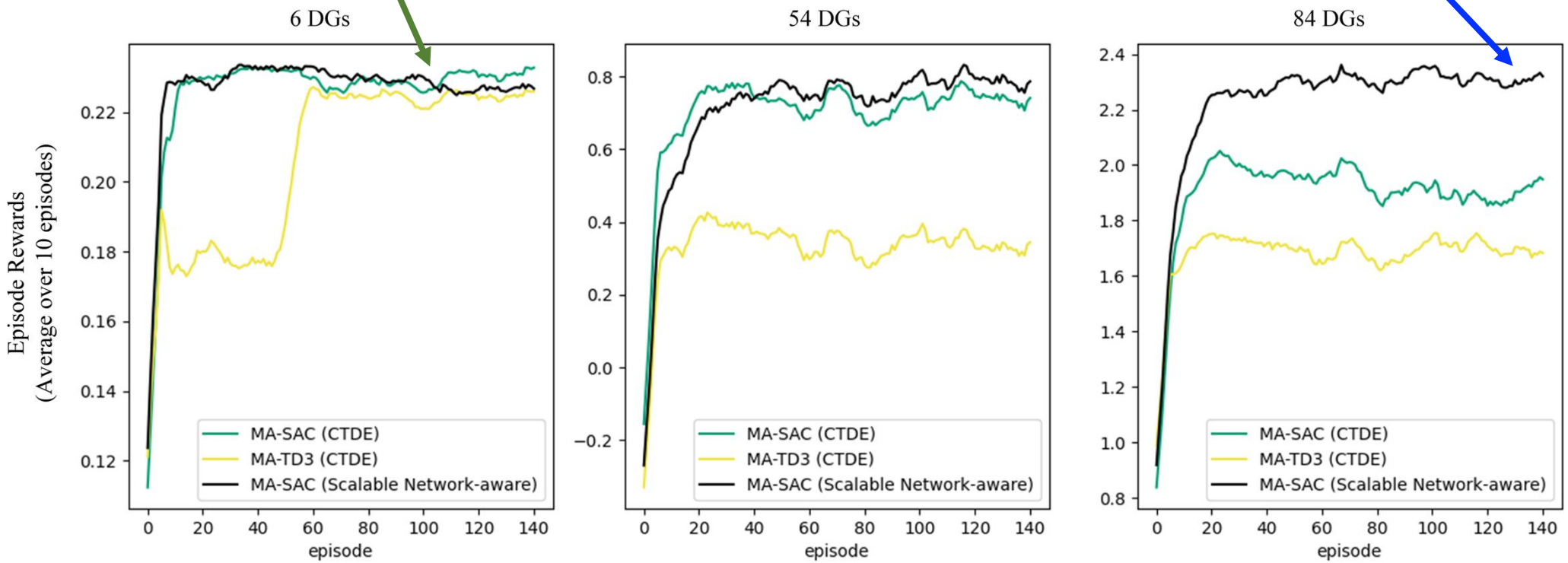
(c) DG model

How many agents (DGs) can we scale up our approach to?

How does our approach compare with Centralized Training Decentralized Execution (CTDE)?

For small systems, our approach is similar to benchmarks

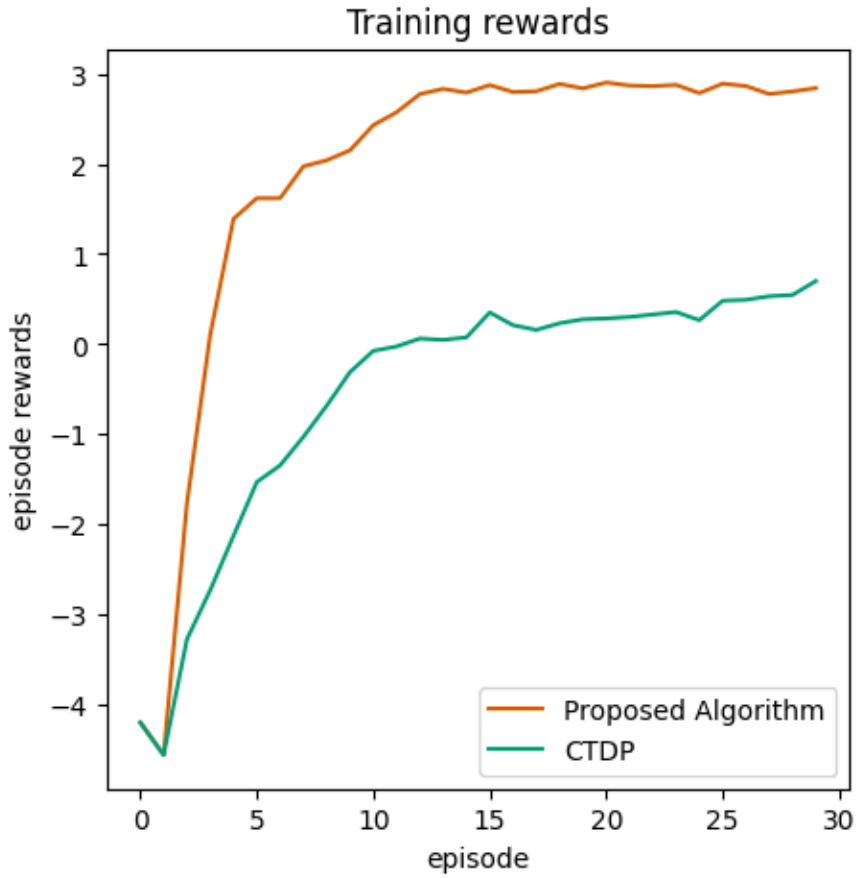
For large systems, our approach is the best



Our approach

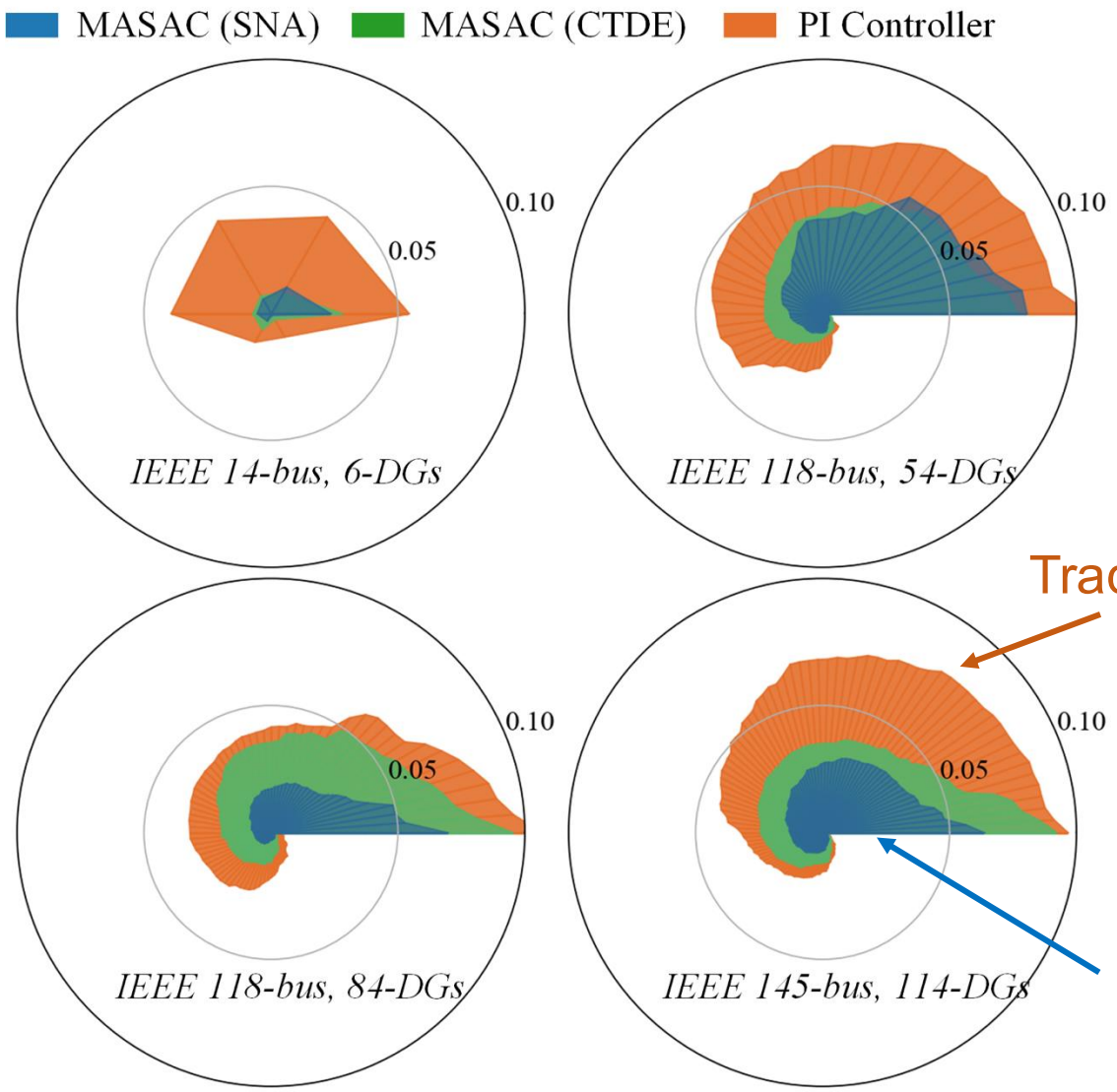
Benchmarks using CTDE

120 DGs



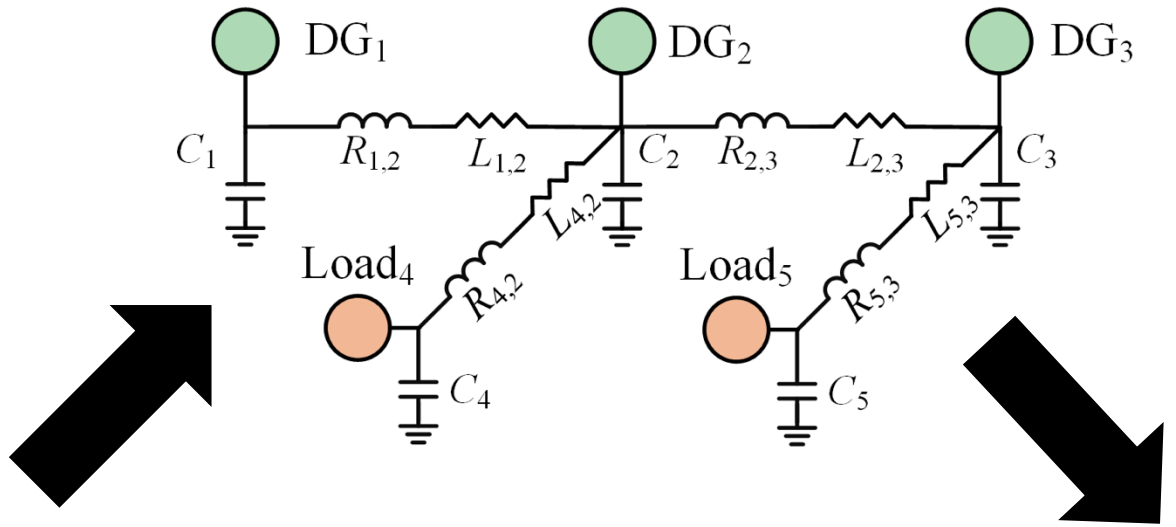
- We can efficiently train on 120 DGs
- For similar microgrid control problems, ***the SOTA is 40 DGs****
- Ongoing: run on even larger systems, currently the main bottleneck is simulator

* Dong Chen, Kaian Chen, Zhaojian Li, Tianshu Chu, Rui Yao, Feng Qiu, and Kaixiang Lin. Powernet: Multi-agent deep reinforcement learning for scalable powergrid control. IEEE Transactions on Power Systems, 37(2):1007–1017, 2021

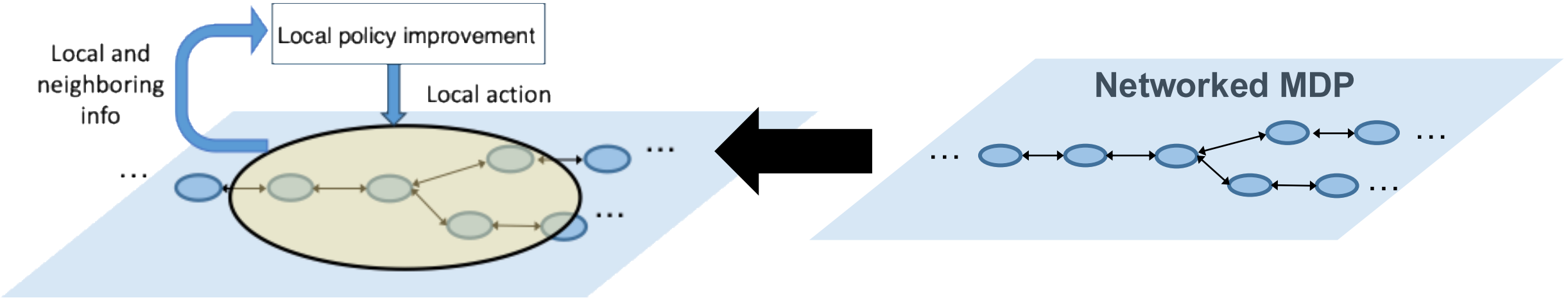


Traditional PI control has large voltage violations

Our approach has the least voltage violation



Scalable Actor Critic



Extensions

What if the objective is average reward (reward in stationarity)?

Guannan Qu, Yiheng Lin, Adam Wierman, Na Li, Scalable Multi-Agent Reinforcement Learning for Networked Systems with Average Reward, NeurIPS 2020.

What if the interaction graph is stochastic?

Yiheng Lin, Guannan Qu, Longbo Huang, Adam Wierman, Distributed Reinforcement Learning in Multi-Agent Networked Systems, arXiv:2006.06555, June 2020.

Theorem [Qu, Wierman, Li 2022]. Under some assumptions, with high probability,

$$\frac{\sum_{m=0}^{M} \eta_m \|\nabla J(\theta(m))\|^2}{\sum_{m=0}^{M} \eta_m} \leq \tilde{O}\left(\frac{\text{poly}_1}{\sqrt{M+1}}\right) + O(\rho^{k+1})$$

Our guarantee is on the norm of gradient, so this is local optimality guarantee!

Also, we have been restricted to decentralized policies, i.e. a_i only depends on s_i

What about global optimality guarantee?

i.e. compare with the best policy among all possible policies (not just decentralized)?

Global Convergence Guarantee

Theorem [Informal, Zhang, Qu et al 2023].

Under conditions on γ, τ and weak interaction, we have

$$\underbrace{J(\zeta^*) - J(\hat{\zeta}^M)}_{\text{Global optimality gap}} \leq \underbrace{\gamma^M \|V^{\hat{\zeta}^0} - V^*\|_\infty}_{\text{Converges to zero}} + \underbrace{O\left(\frac{1}{(k+1)^\mu}\right)}_{\varepsilon_k\text{: error due to using } k\text{-hop policy}}$$

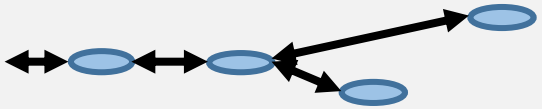
when the rollout length per iteration $T \geq \tilde{\Theta}(\frac{1}{\varepsilon_k^2}\text{poly})$

- This is the **global optimal guarantee**.
- Sub-optimality gap is **decaying polynomially in k**
- Similar tradeoff of k between optimality and complexity as before.

Today's Tutorial

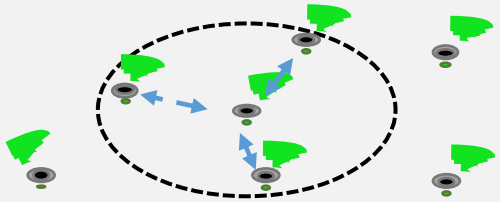
Part 1: Background on reinforcement learning

Part 2: Exploiting network topology structure

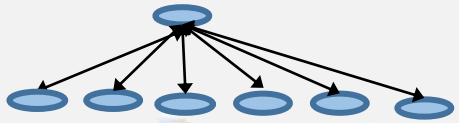


Up Next

Part 3: Exploiting locality structure



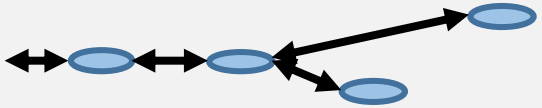
Part 4: Exploiting homogeneity structure



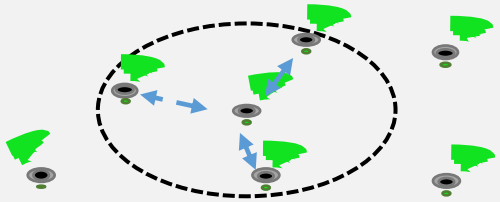
Today's Tutorial

Part 1: Background on reinforcement learning

Part 2: Exploiting network topology structure

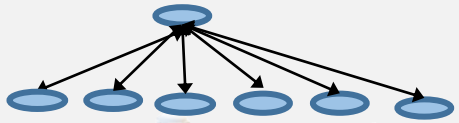


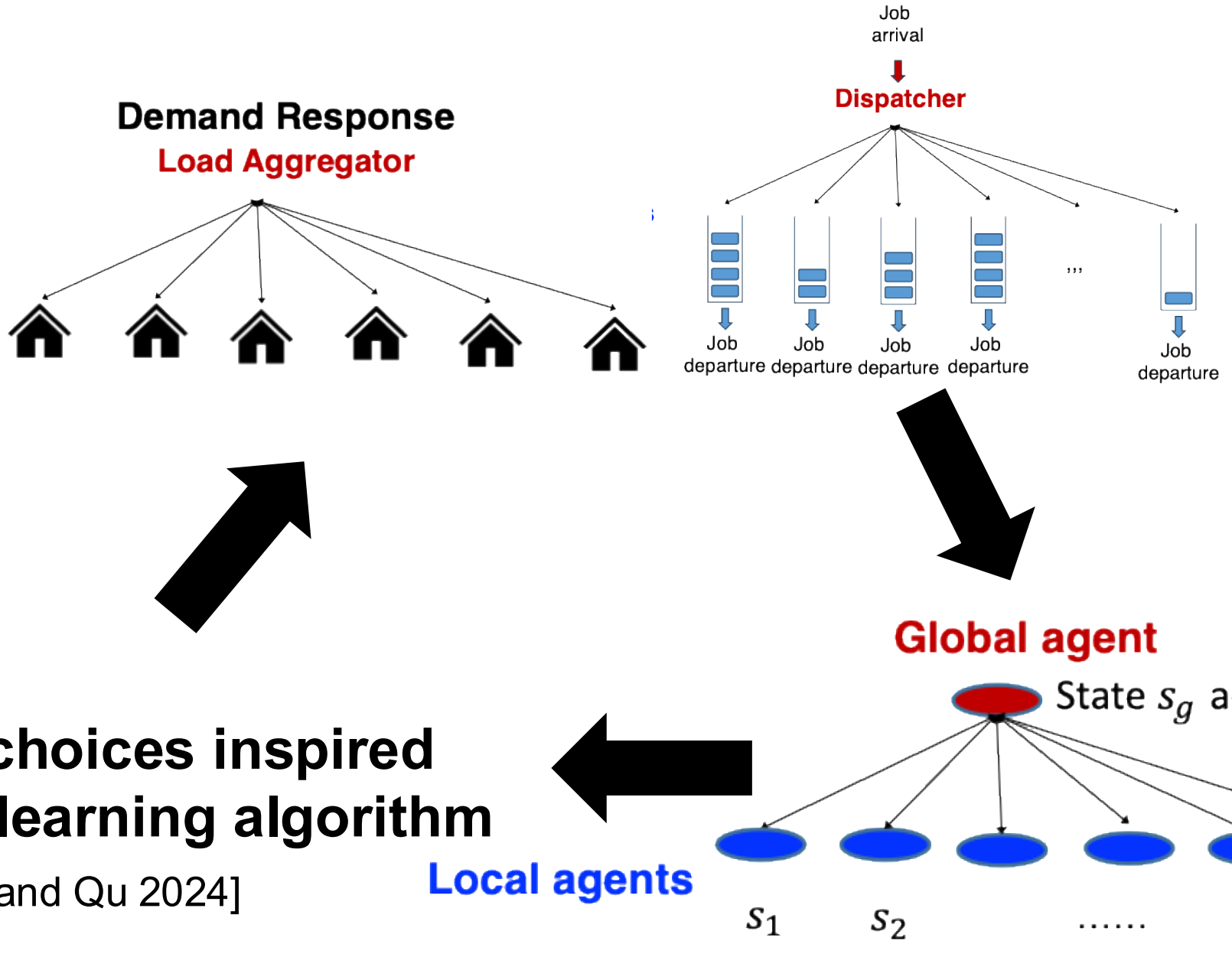
Part 3: Exploiting locality structure



Up Next

Part 4: Exploiting homogeneity structure

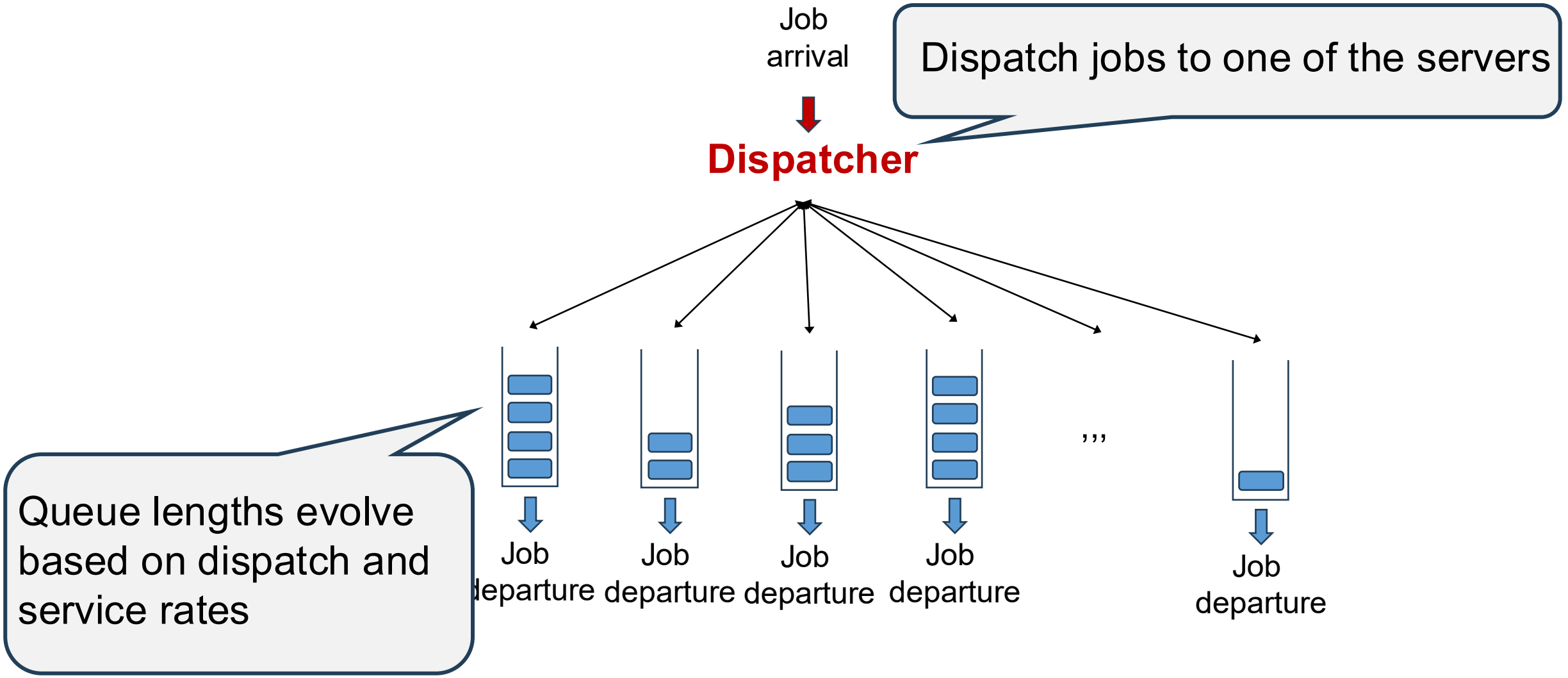




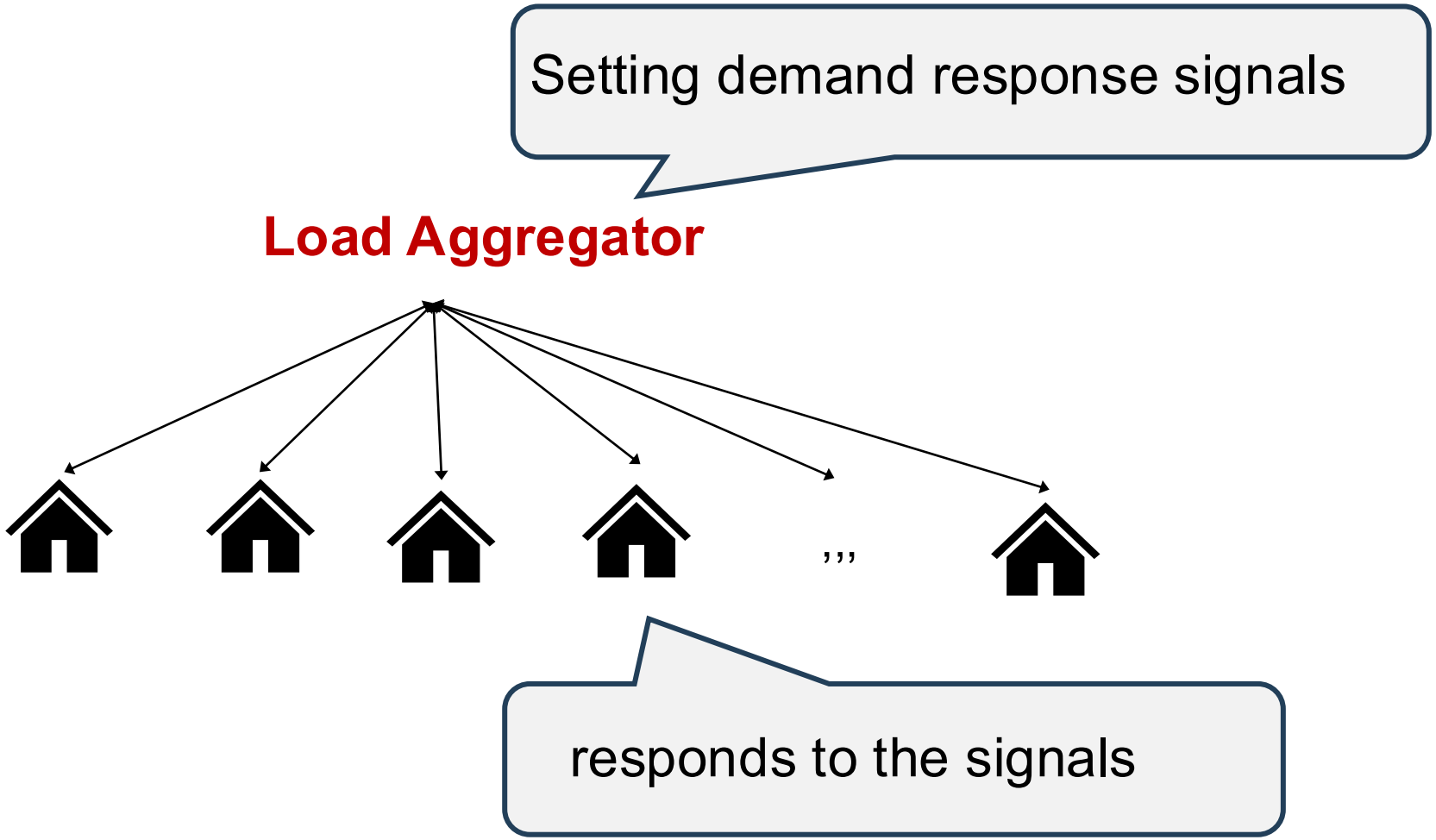
Power of k-choices inspired subsample Q-learning algorithm

[Anand and Qu 2024]

Example: Load balancing



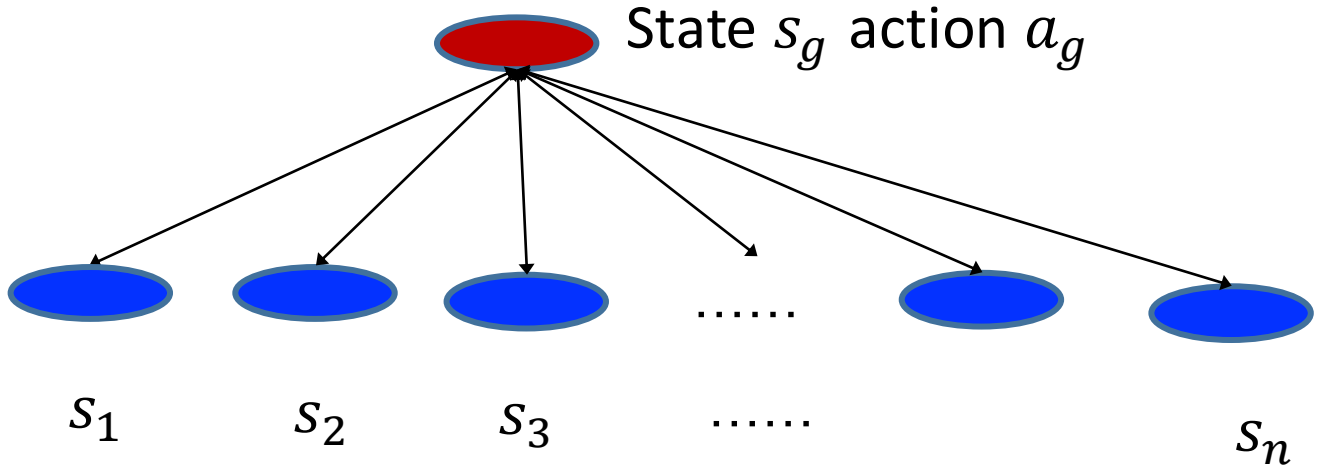
Example: demand response



Global state transitions

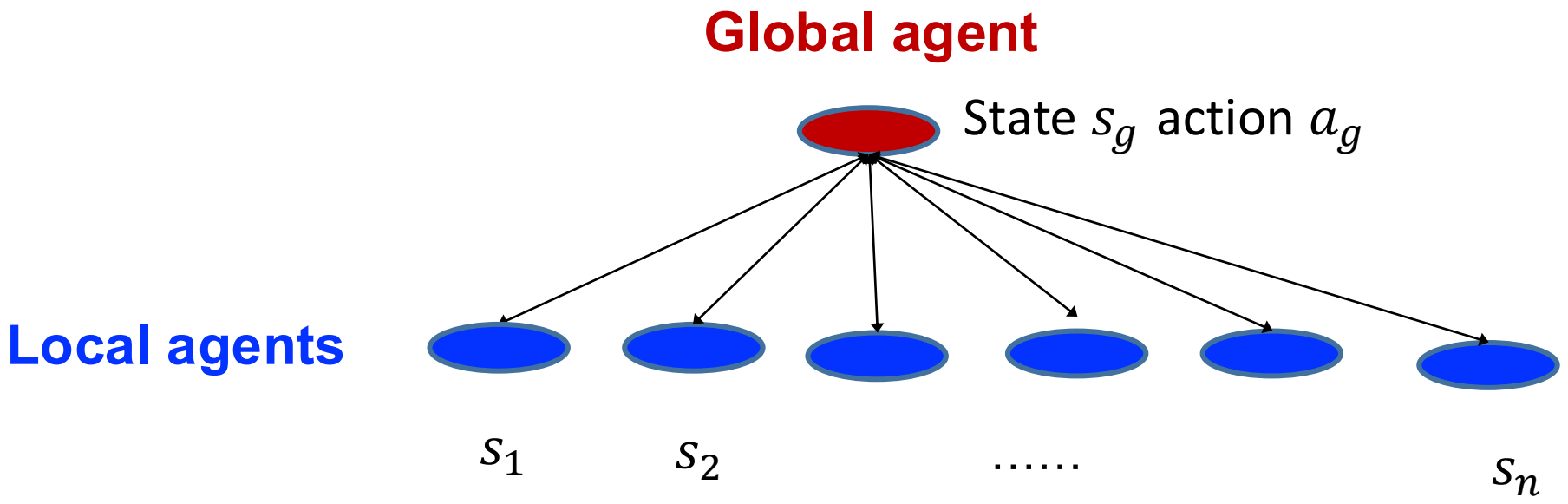
$$s_g(t+1) \sim P_g(\cdot \,|s_g(t), a_g(t))$$

Global agent

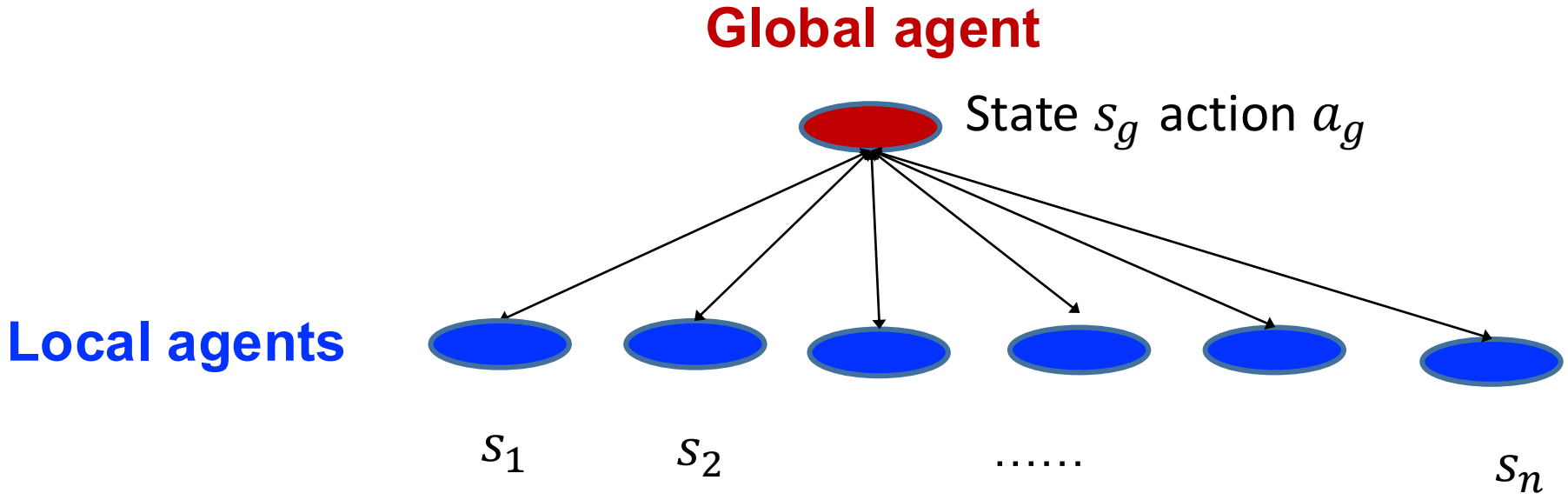


$$s_i(t+1) \sim P_{local}(\cdot \,|s_g(t), s_i(t))$$

Local agent states influenced by global state



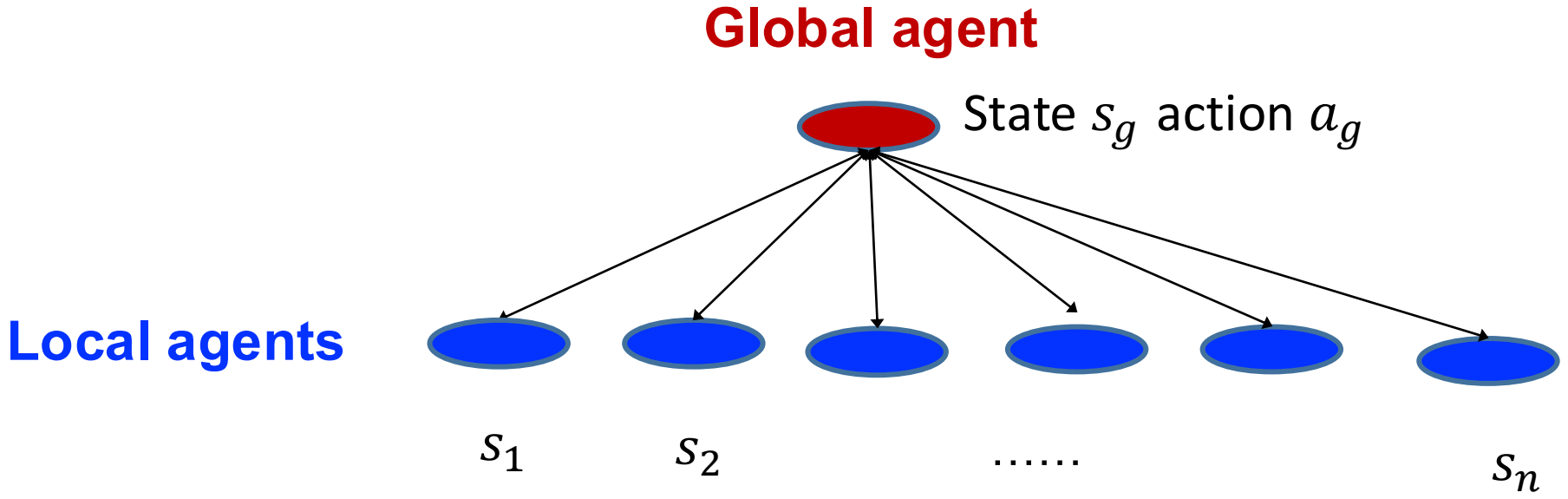
How to design policies to achieve optimal reward?



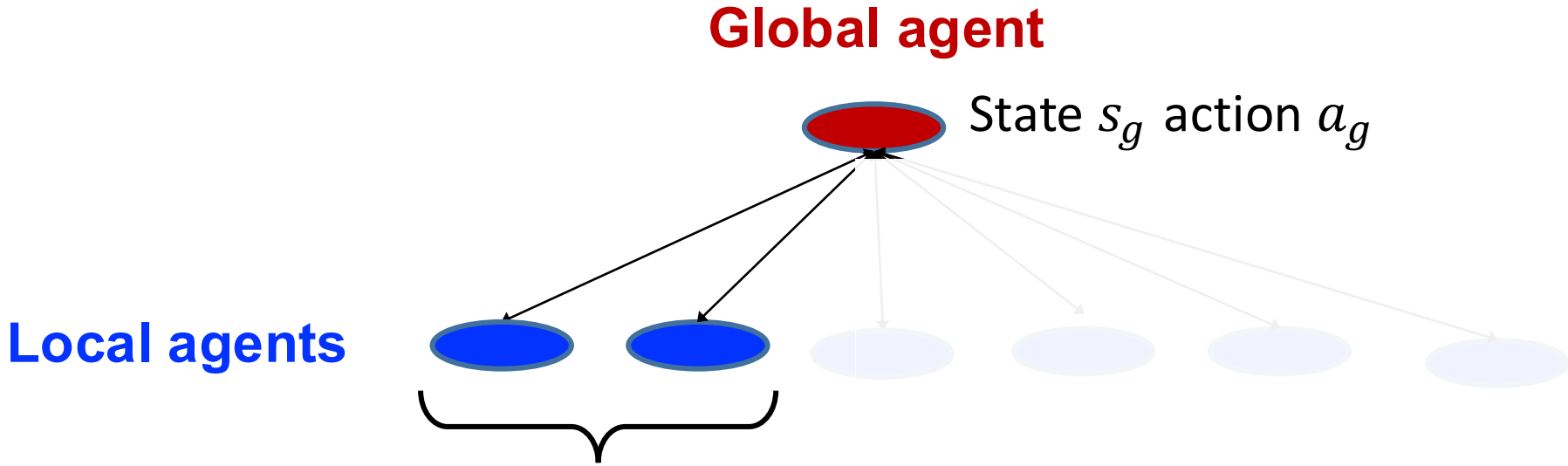
Directly solving this via RL is intractable for large n

$$s = \underbrace{(s_g, s_1, \ldots, s_n)}$$

State space size exponentially large in n!



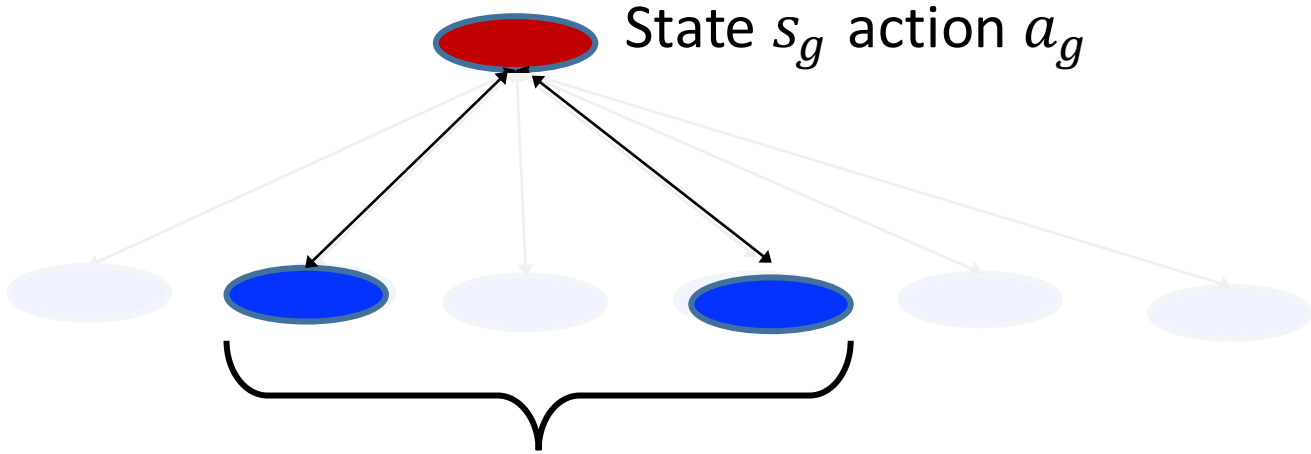
Our approach: structure of MDP enable "**power of two choices**" trick!



Training: assuming only k local agents (ignoring the rest)
Solve for the optimal policy

$k \ll n$ (the pic. shows $k = 2$)

Global agent



Local agents

In implementation, at each time, randomly sample k agents
Apply the policy learned for the k-agent-system

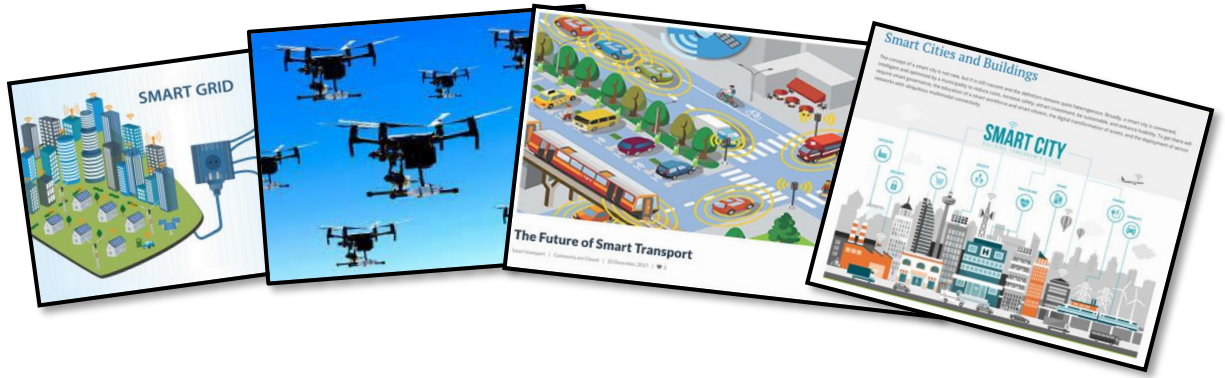
Complexity $2^{\Theta(k)} \ll 2^{\Theta(n)}$

which is exponential speed up as $k \ll n$

Theorem. [Anand and Qu 2024]
The learned ”power-of-k-choices” policy has the following optimality gap

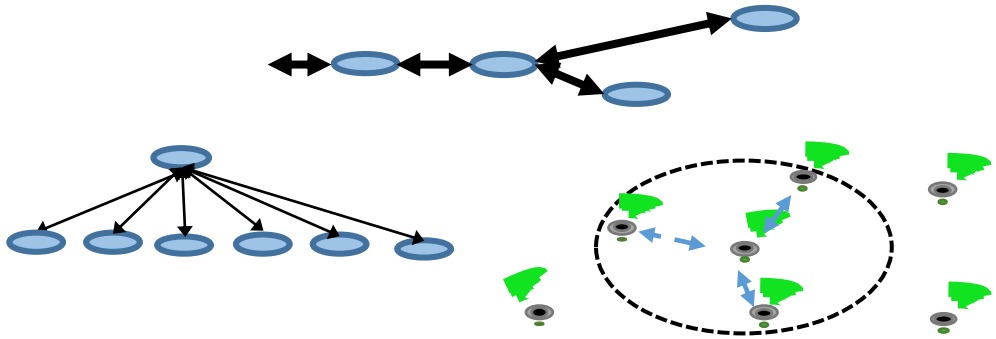
$$V^* - V^\pi \leq \underbrace{\tilde{O}\left(\frac{r_{max}}{1-\gamma}\frac{1}{\sqrt{k}}\right)}_{\text{Inherent loss due to randomly sample k agents}} + \underbrace{\text{statistical error}}_{\text{Converges to 0 as samples increase}}$$

- Even for a small k, we get close to optimality
- Avoids exponential complexity in n!



Apply to applications

Extract abstract structural property



$$x_{t+1} = f_{physics}(x_t, u_t)$$

Exploit structure to obtain "positive" MARL theory

MARL Algorithm
with theoretical guarantee
that **avoids intractability**

[Network structure]

- Guannan Qu, Adam Wierman, Na Li, Scalable Reinforcement Learning for Multi-Agent Networked Systems, Operations Research 2022
- Guannan Qu, Yiheng Lin, Adam Wierman, Na Li, Scalable Multi-Agent Reinforcement Learning for Networked Systems with Average Reward, NeurIPS 2020
- Yiheng Lin, Guannan Qu, Longbo Huang, Adam Wierman, Multi-Agent Reinforcement Learning in Stochastic Networked Systems, NeurIPS 2021.
- Yizhou Zhang*, Guannan Qu*, Pan Xu*, Yiheng Lin, Zaiwei Chen, Adam Wierman, Global Convergence of Localized Policy Iteration in Networked Multi-Agent Reinforcement Learning, ACM SIGMETRICS 2023.
- Han Xu, Guannan QU, A Scalable Network-Aware Multi-Agent Reinforcement Learning Framework for Distributed Converter-based Microgrid Voltage Control, Tackling Climate Change with Machine Learning workshop at NeurIPS 2023

[Locality-based time-varying networks]

- Alex Deweese, Guannan Qu, “Interdependent Multi-Agent MDP: Theoretical Framework for Decentralized Agents with Dynamic Local Dependencies”, ICML 2024
- Alex Deweese, Guannan Qu, “Thinking Beyond Visibility: A Near-Optimal Policy Framework for Locally Interdependent Multi-Agent MDPs”, preprint 2025.

[Homogeneous structure]

- Emile Anand, Guannan Qu, “Efficient Reinforcement Learning for Global Decision Making in the Presence of Local Agents at Scale”, NeurIPS 2025 (Spotlight).

[Physics model]

- Zeji Yi, Chaoyi Pan, Guanqi He, Guannan Qu*, Guanya Shi*, CoVO-MPC: Theoretical Analysis of Sampling-based MPC and Optimal Covariance Design, 6th Learning for Dynamics and Control 2024.
- Chaoyi Pan, Zeji Yi, Guanya Shi*, Guannan Qu*, Model-based diffusion for trajectory optimization, NeurIPS 2024.