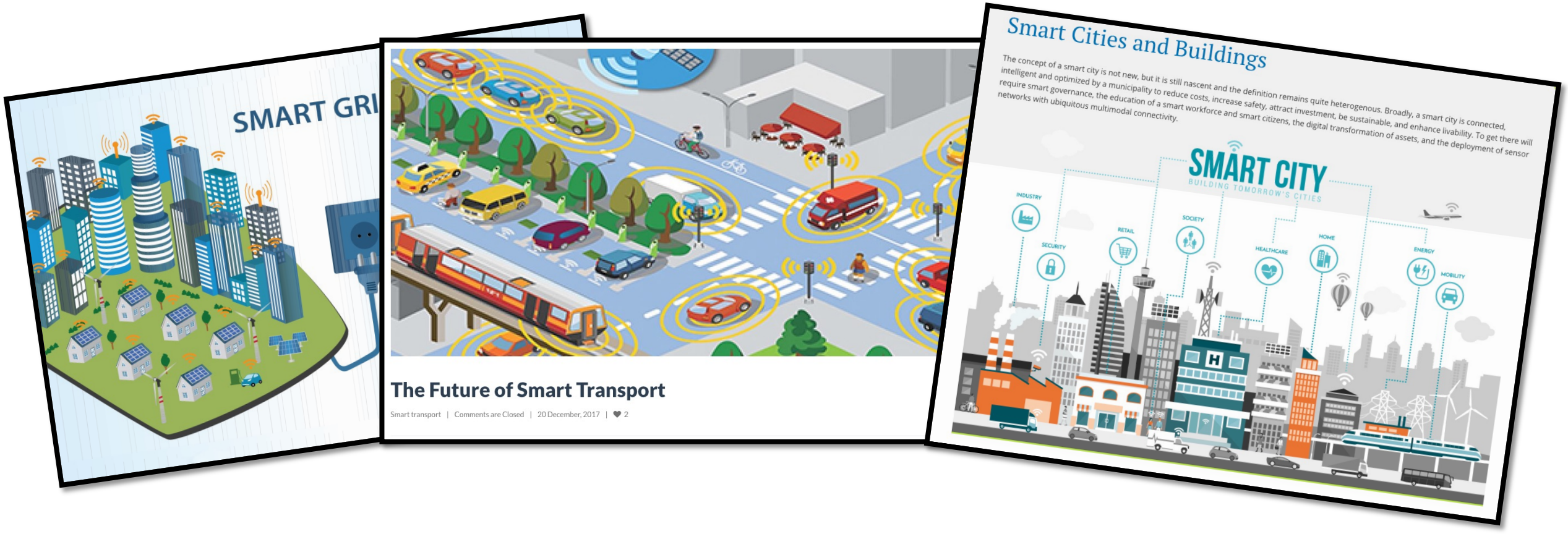
Scalable Reinforcement Learning for Multi-Agent Networked Systems

Guannan Qu
Assistant Professor of ECE
Carnegie Mellon University

Networked systems are becoming smarter



Picture credit: IEEE, qrowd, ARC advisory group

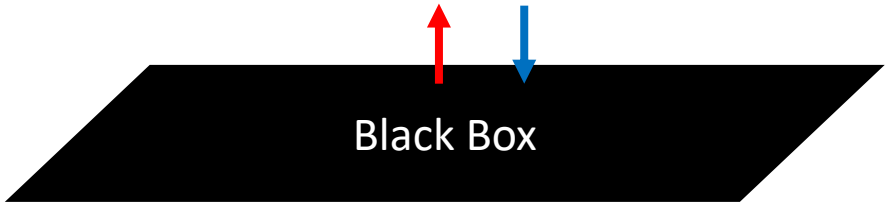
Opportunity: harnessing the data revolution

Increasing number of sensors

Increasing on-device computation power

Increasing communication capabilities

Reinforcement Learning (RL)



Adaptive to complex/hard-to-model dynamics

Great potential for smart networked systems

Artificial intelligence can make the U.S. electric grid smarter

BY CHRISTINA NUNEZ | JUNE 14, 2019

Argonne researchers apply machine learning to inform more reliable grid planning and operations.

The following article is part of a series on Argonne National Laboratory's efforts to use the predictive power of artificial intelligence, specifically machine learning, to advance discoveries in a broad range of scientific disciplines.

How much electricity will you need tomorrow? Answering

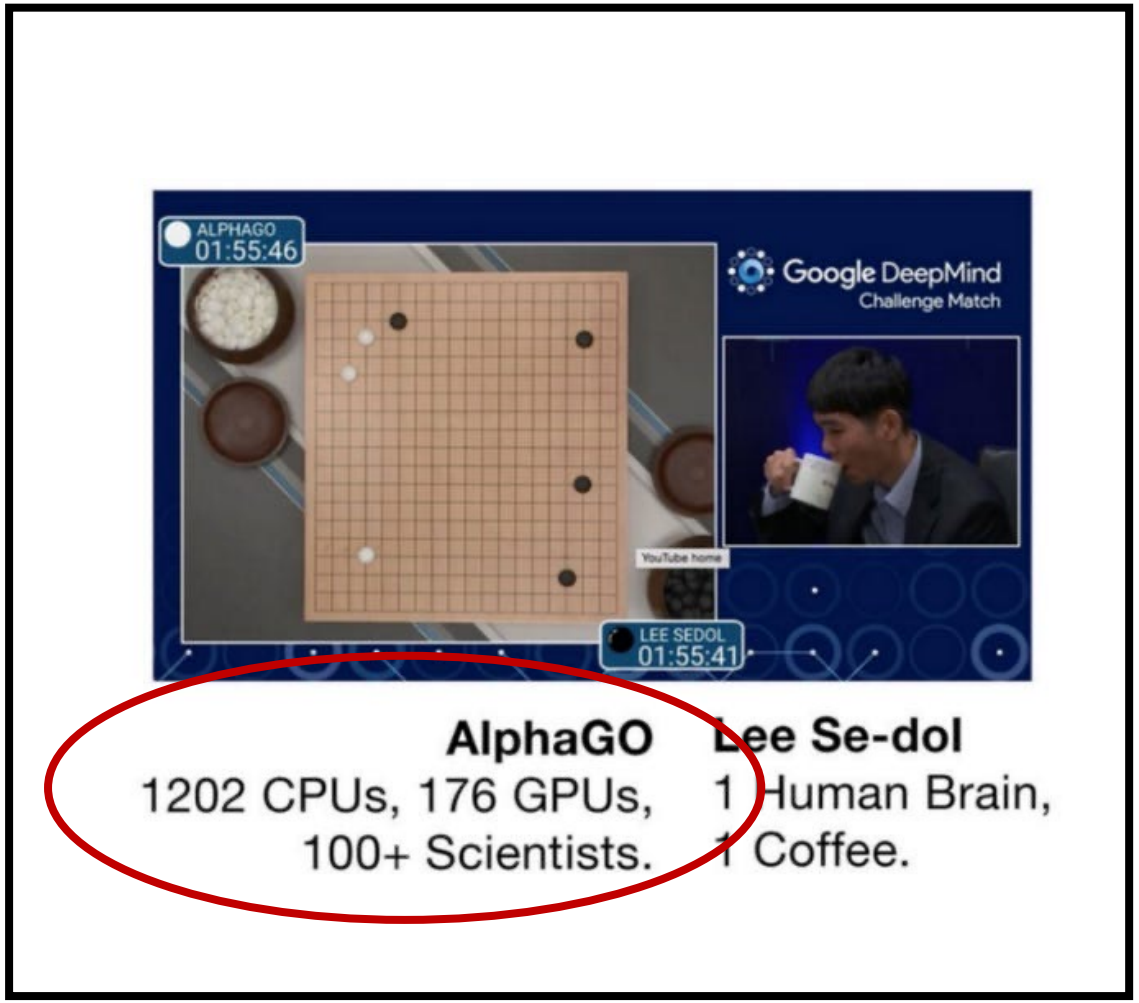
Researchers at Argonne National Laboratory are working on optimization models that use machine learning, a form of artificial intelligence, to simulate the electric system and the severity of various problems. In a region with 1,000 electric power assets, an outage of just three assets can produce nearly a billion scenarios of potential failure. Image by urbans/Shutterstock.com.

Jul 26, 2019, 09:47pm EDT | 23,188 views

How AI Can Transform The Transportation Industry

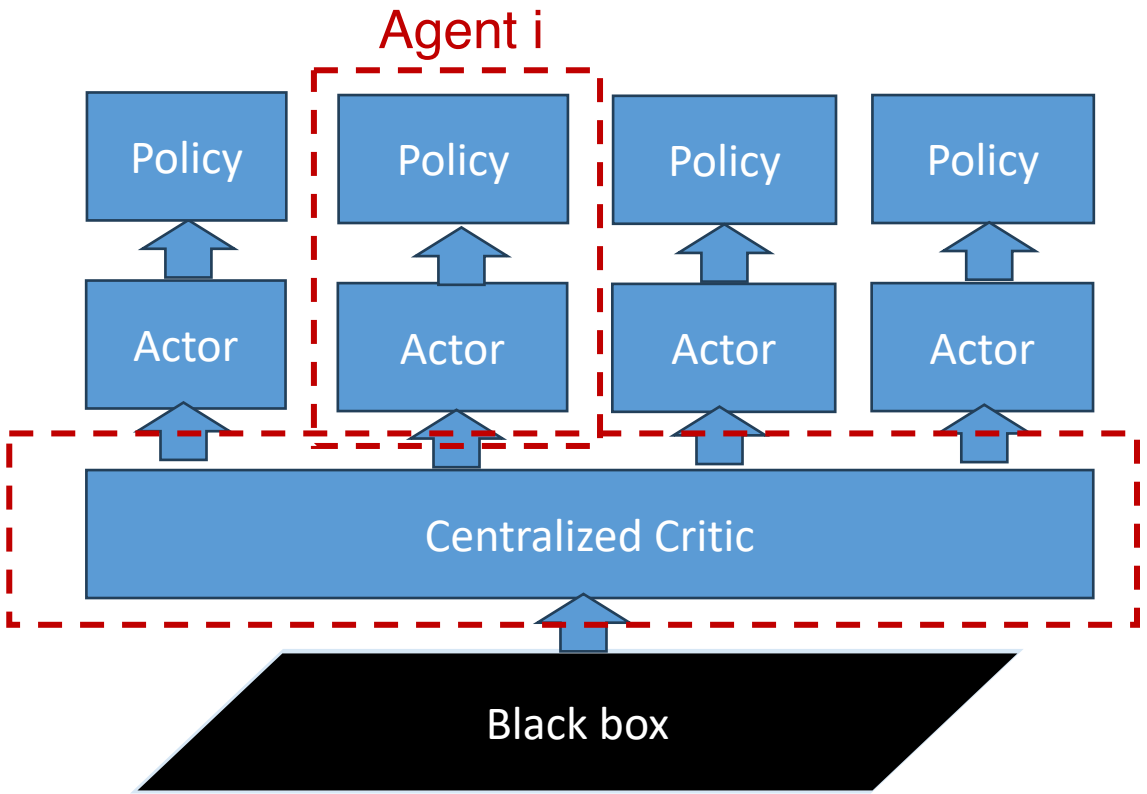
Naveen Joshi Contributor
COGNITIVE WORLD Contributor Group
AI

Realizing the potential turns out to be challenging



Realizing the potential turns out to be challenging

SOTA: Centralized Training, Decentralized Execution/Play



Centralized Training becomes harder and harder when network size increase

Realizing the potential turns out to be challenging

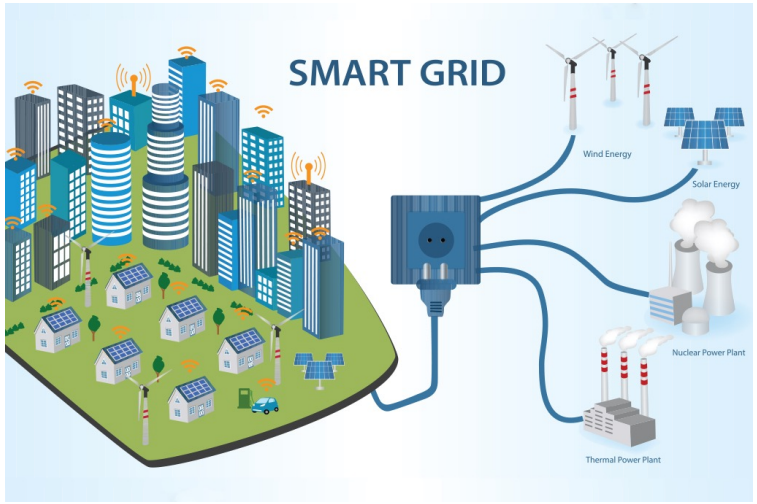
The StarCraft Multi-Agent Challenge

(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_5z	3 Stalkers	5 Stalkers
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



The Future of Smart Transport

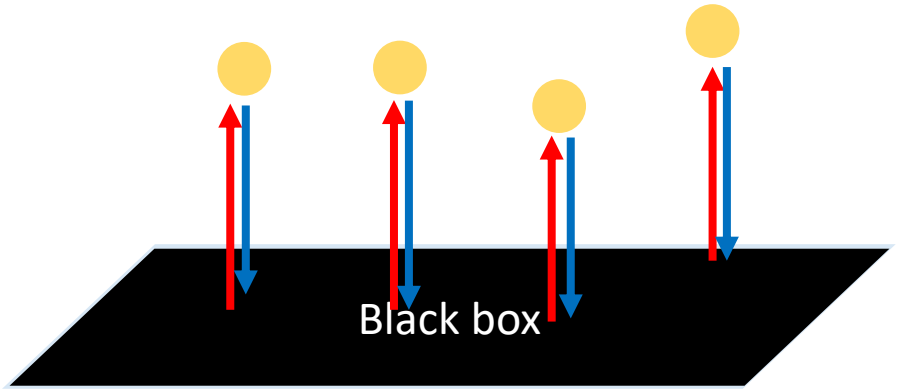
Smart transport | Comments are Closed | 20 December, 2017 | ♥ 2

Today's Talk

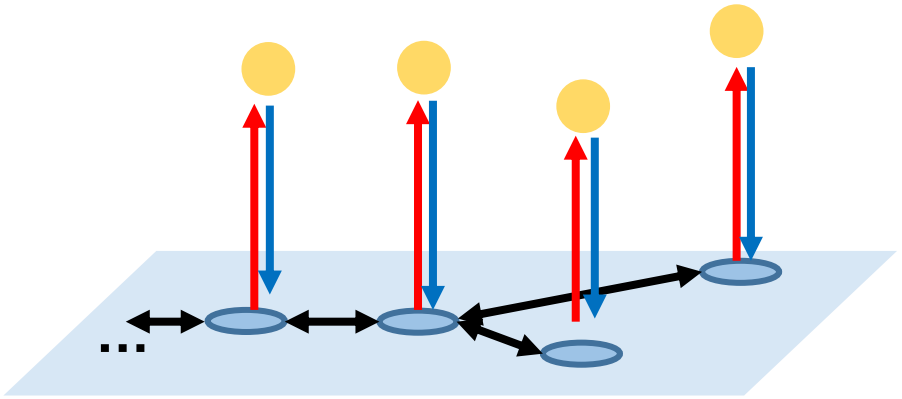
How to do Scalable RL for Large Scale Networks

Picture credit: IEEE, qrowd, ARC advisory group

Off-the-shell RL treats the system as blackbox…

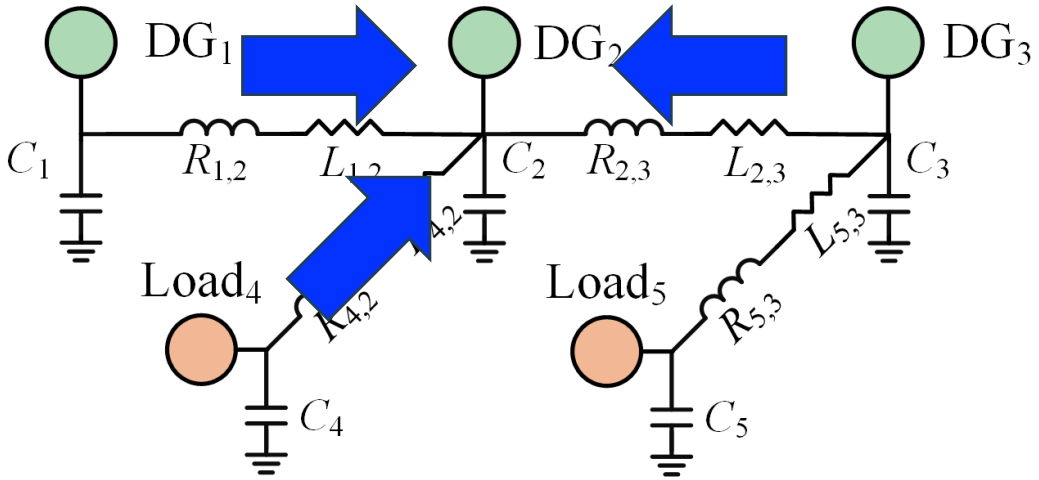


But we know there is a "network structure" underlying the system



Power Networks
[Chen et. al. 2022]

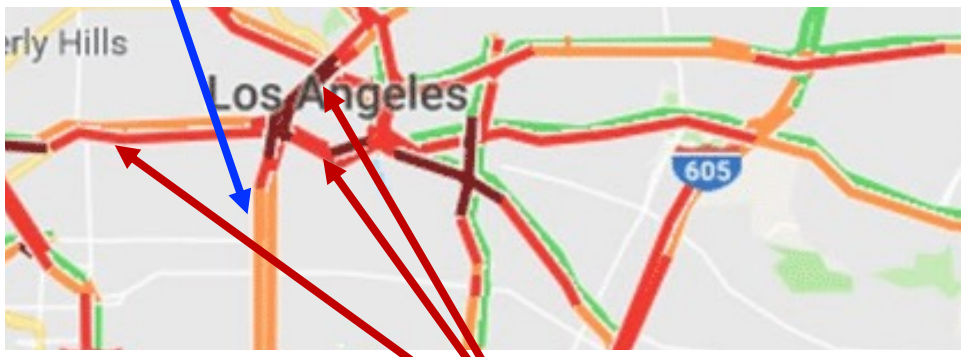
Power flows along the transmission lines



(a) Diagram of a Microgrid

Traffic Networks
[Varaiya 2013]

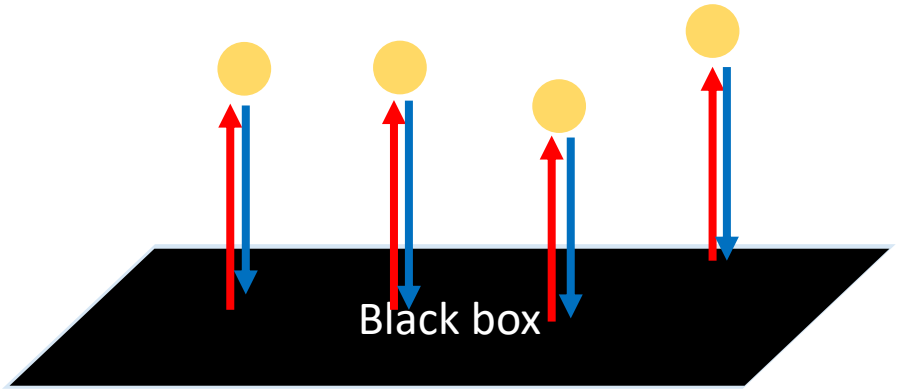
Congestion of a road…



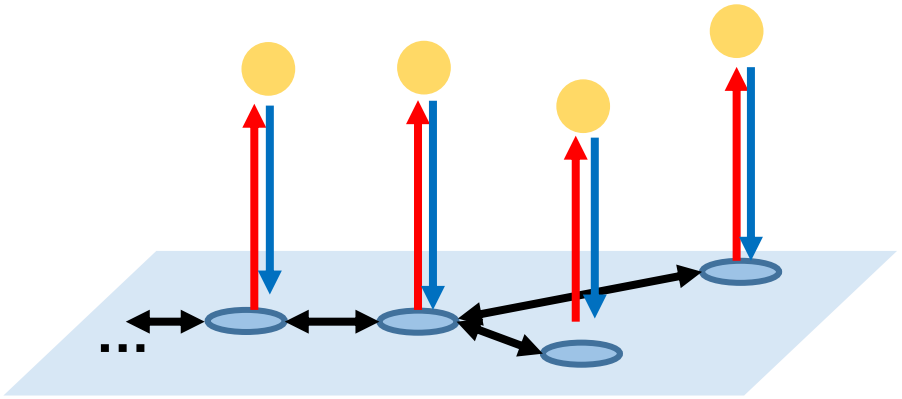
… depends on that of nearby roads

* Picture credits: keepomahamoving.com

Off-the-shell RL treats the system as blackbox…

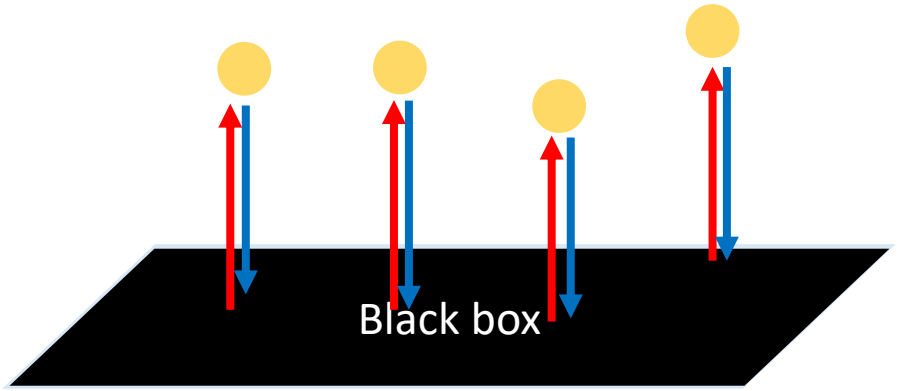


But we know there is a "network structure" underlying the system

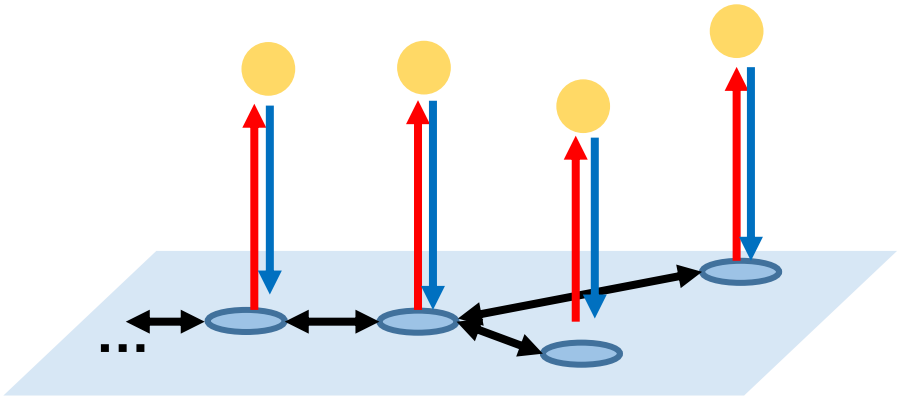


Can we exploit network structure to do RL in a **scalable** manner?

Off-the-shell RL treats the system as blackbox…



But we know there is a "network structure" underlying the system



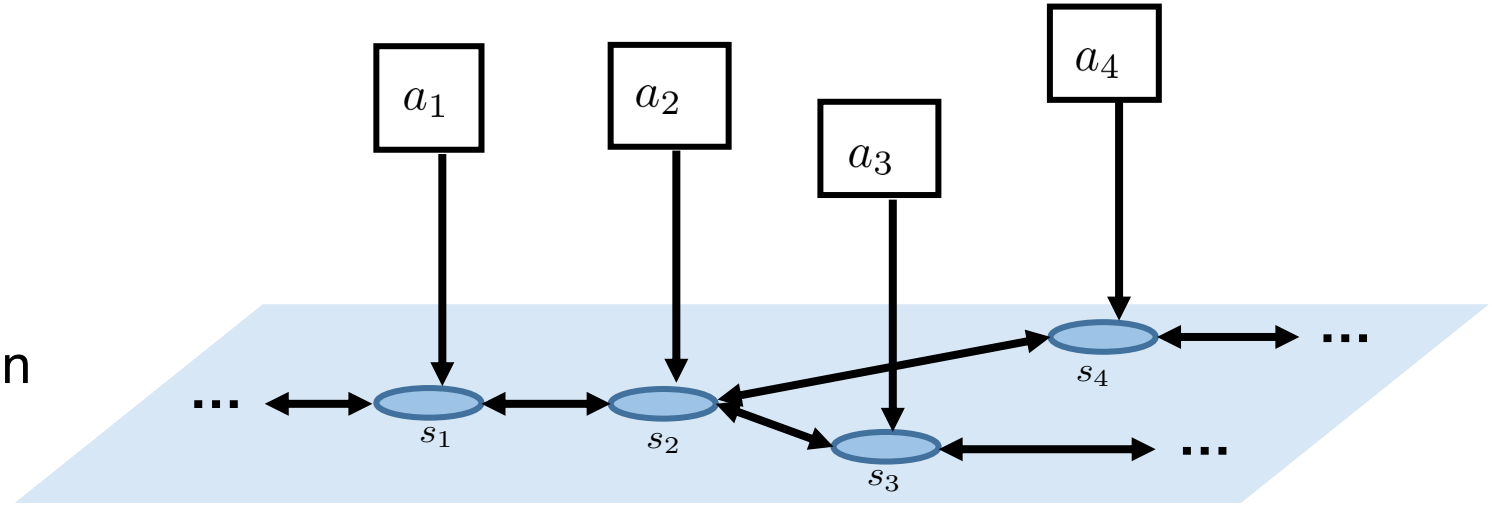
Road Map

Formally define network structure in RL → **Exponential Decay Property** → **Algorithm: Scalable Actor Critic** → **Experimental Validation**

Markov Decision Process (MDP) over network

Local State, Action

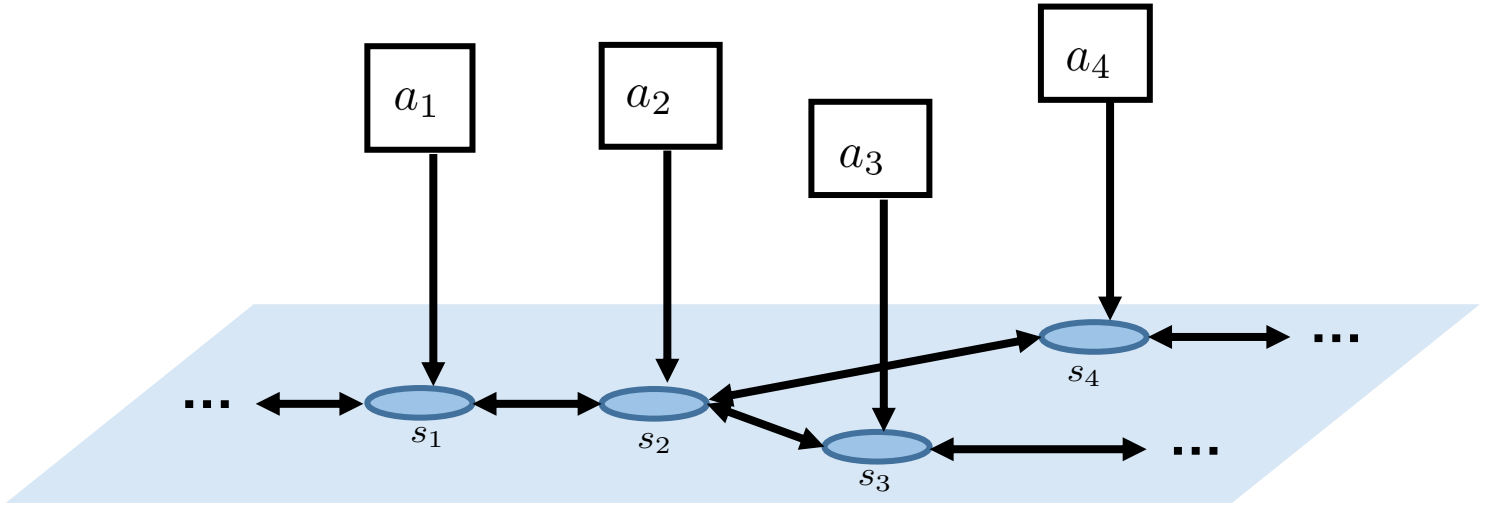
$s_i \in \mathcal{S}_i, a_i \in \mathcal{A}_i$



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



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

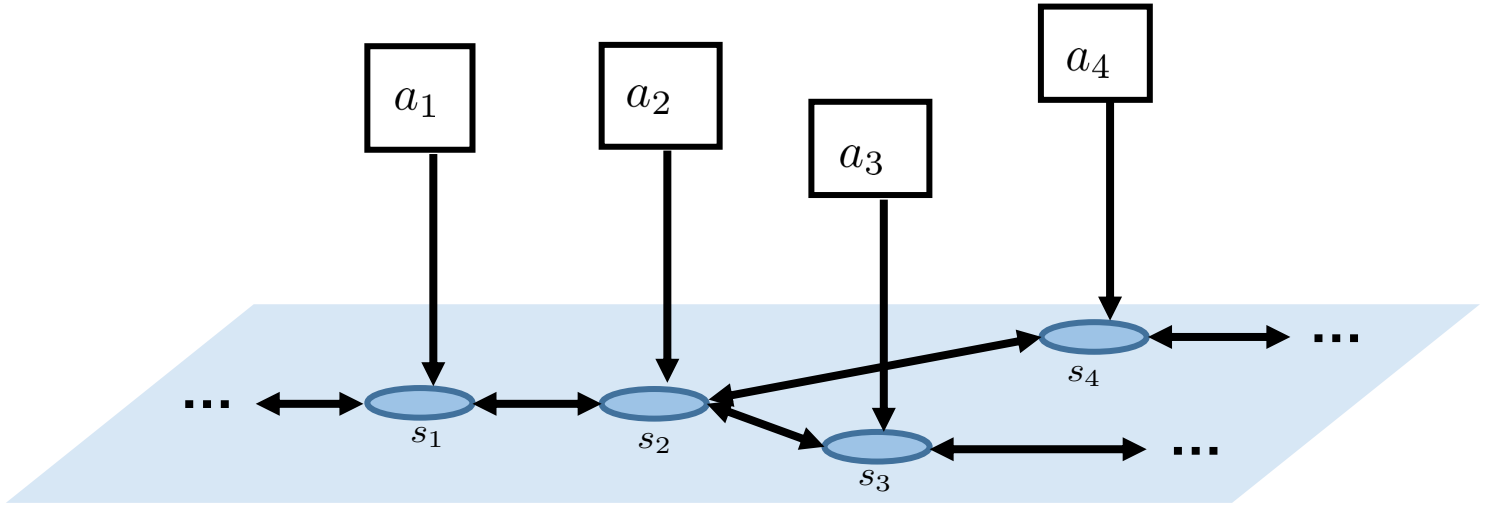
i's next state

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.

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

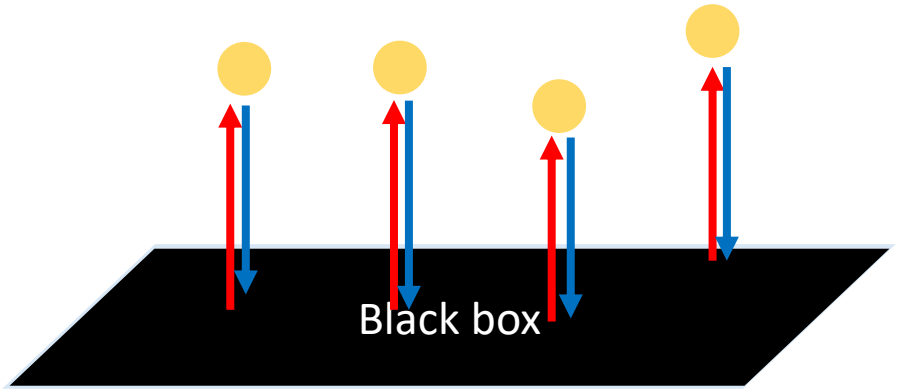
local action local state

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

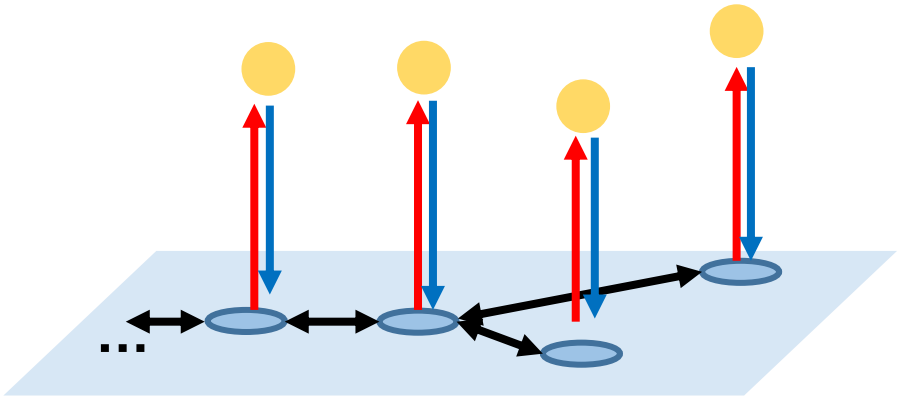
$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s(t), a(t)) | s(0) = s\right]$$

Infinite horizon discounted reward

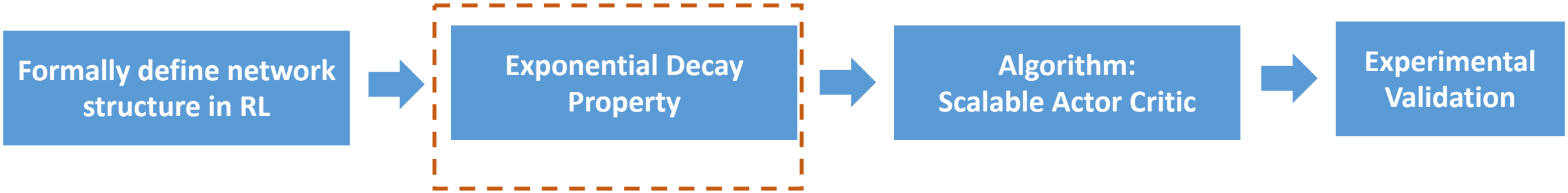
Off-the-shell RL treats the system as blackbox…



But we know there is a "network structure" underlying the system



Road Map

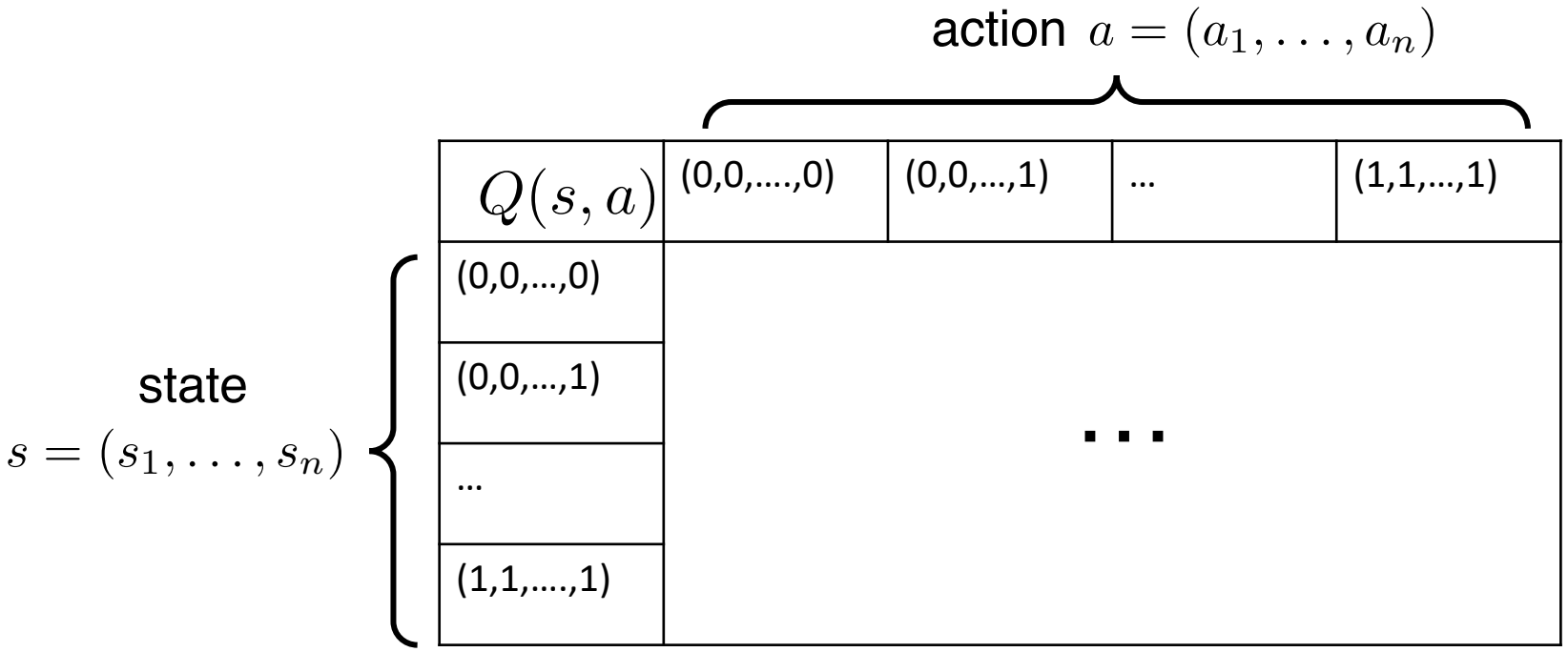


Standard RL doesn't scale to multi-agent systems

Temporal Difference-learning [Sutton 1988], Q-learning [Watkins1989], Actor-Critic methods [Konda and Tsitsiklis 2000], …

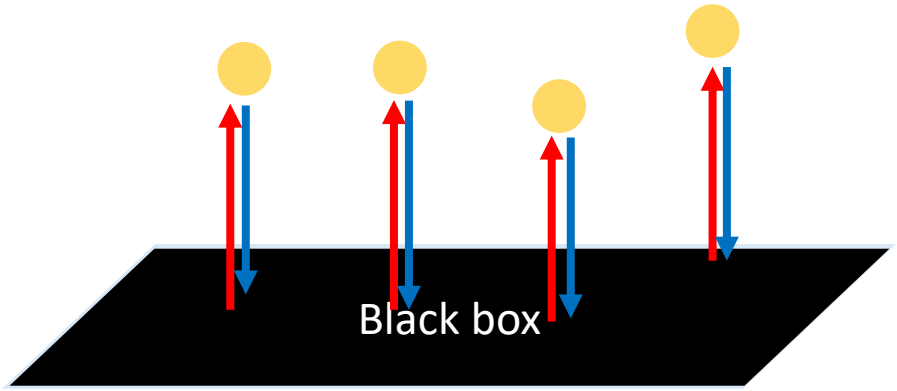
$$Q(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s(t),a(t)) \middle| s(0)=s, a(0)=a\right]$$

Standard RL doesn't scale to multi-agent systems

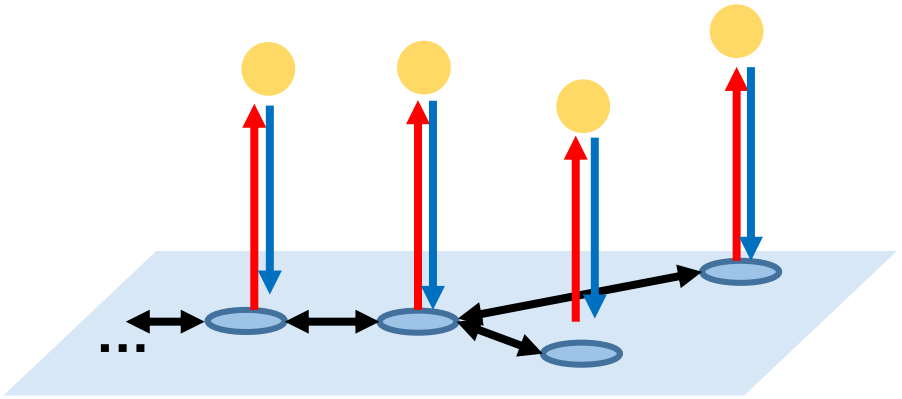


Time/space complexity exponential in n!

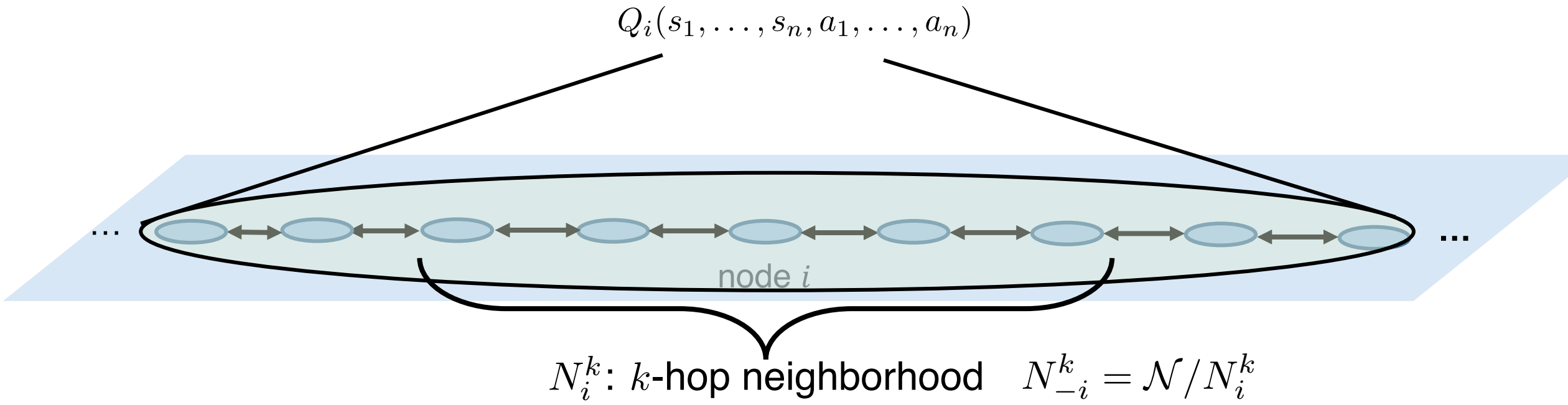
No structure used so far



Exploit network structure!



Is it possible to exploit model structure to do **scalable** RL with **provable** guarantee?



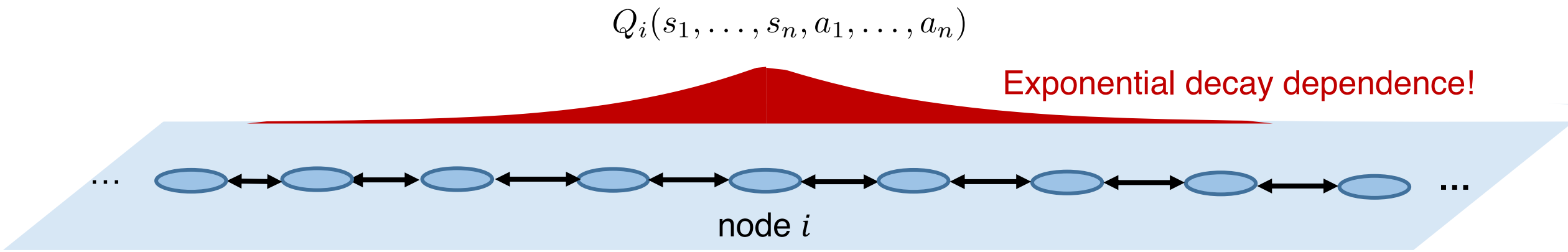
Definition.* The (c, ρ)-exponential decay property holds if for integer k,

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

change state/action outside N_i^k

exponential decay

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



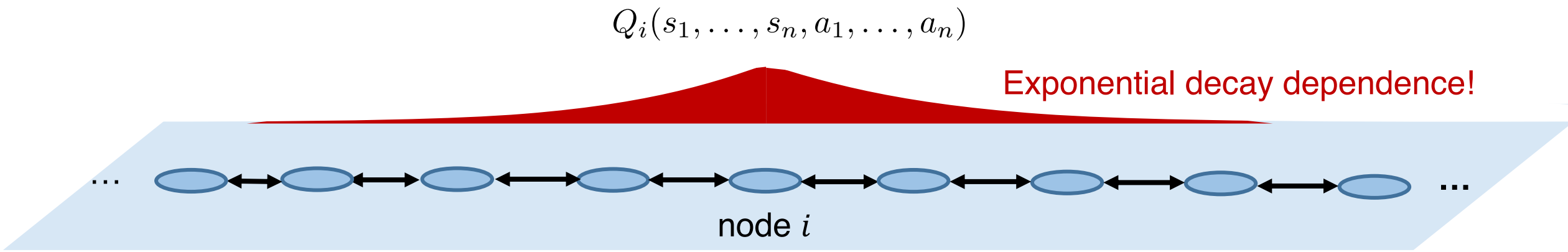
Definition.* The (c, ρ)-exponential decay property holds if for integer k,

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

change state/action outside N_i^k

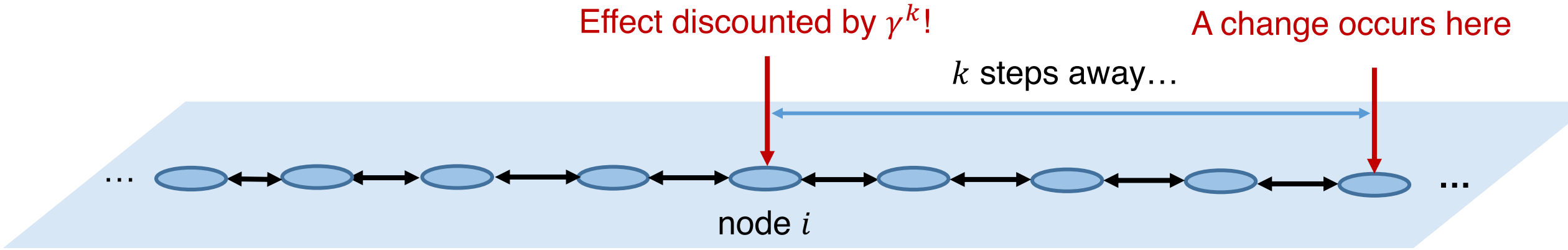
exponential decay

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



Lemma [Qu, Wierman, Li 2019]:

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

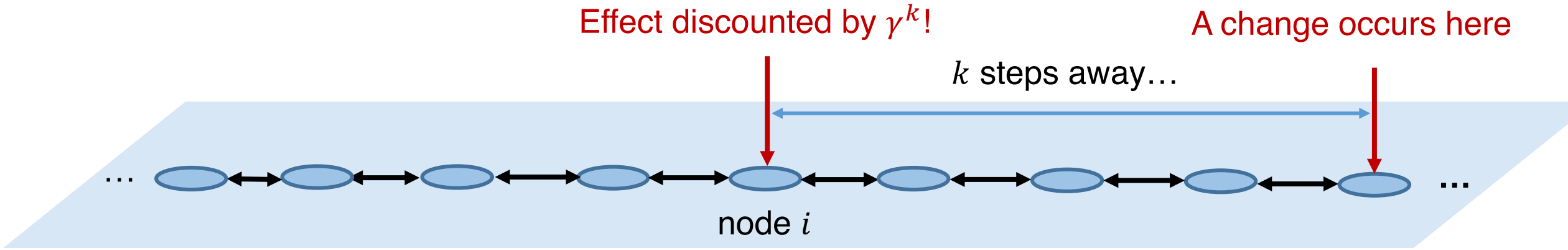


Proof.

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

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

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



Lemma [Qu, Wierman, Li 2019]:

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

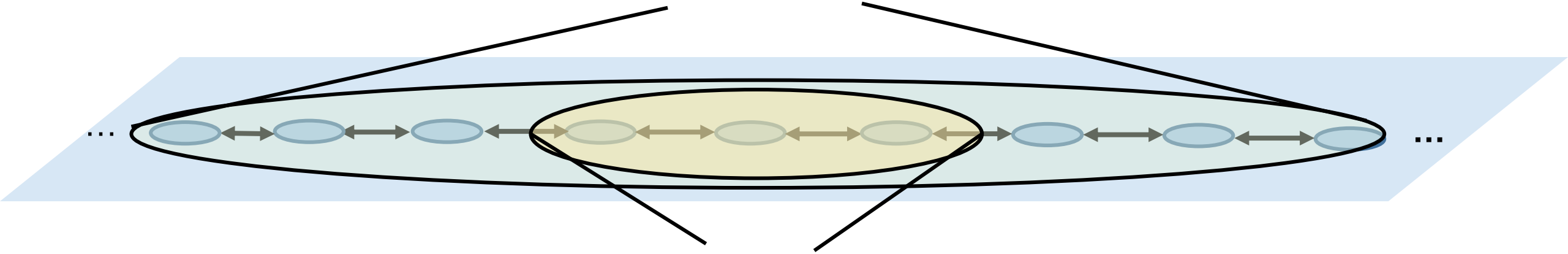
$$\forall i, \sum_{j \in N_i} C_{ij} \leq \mu < 1,$$

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

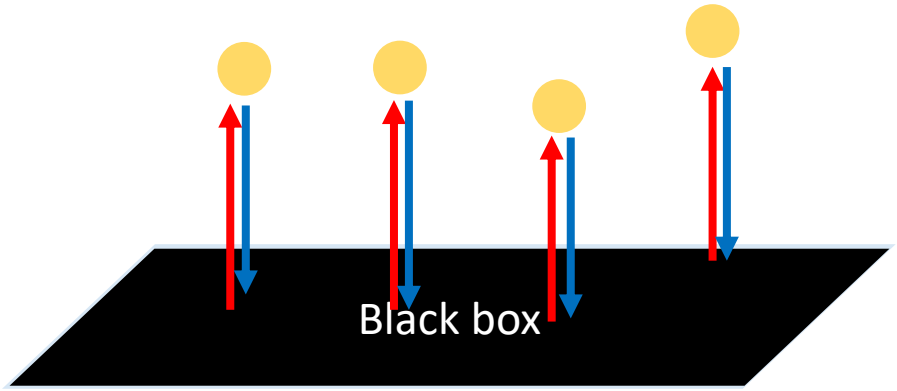
$$\hat{Q}_i(s_{N_i^k}, a_{N_i^k}) = \sum_{s_{N_{-i}^k}, a_{N_{-i}^k}} w_i(s_{N_{-i}^k}, a_{N_{-i}^k}; s_{N_i^k}, a_{N_i^k}) Q_i(s_{N_i^k}, s_{N_{-i}^k}, a_{N_i^k}, a_{N_{-i}^k})$$

Arbitrary weights **Average out outside neighborhood**

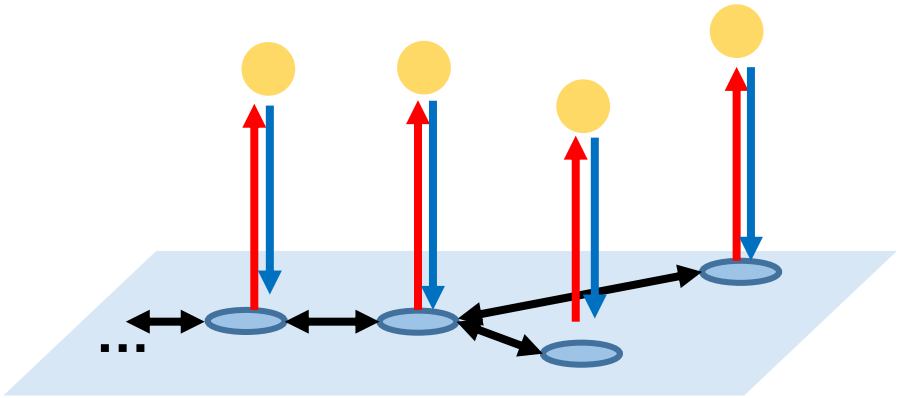
Lemma (informal) Under the (c, ρ)-exponential decay property, then

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

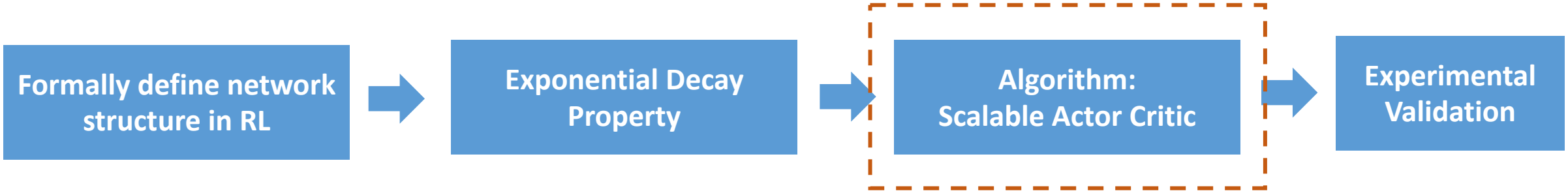
Off-the-shell RL treats the system as blackbox…



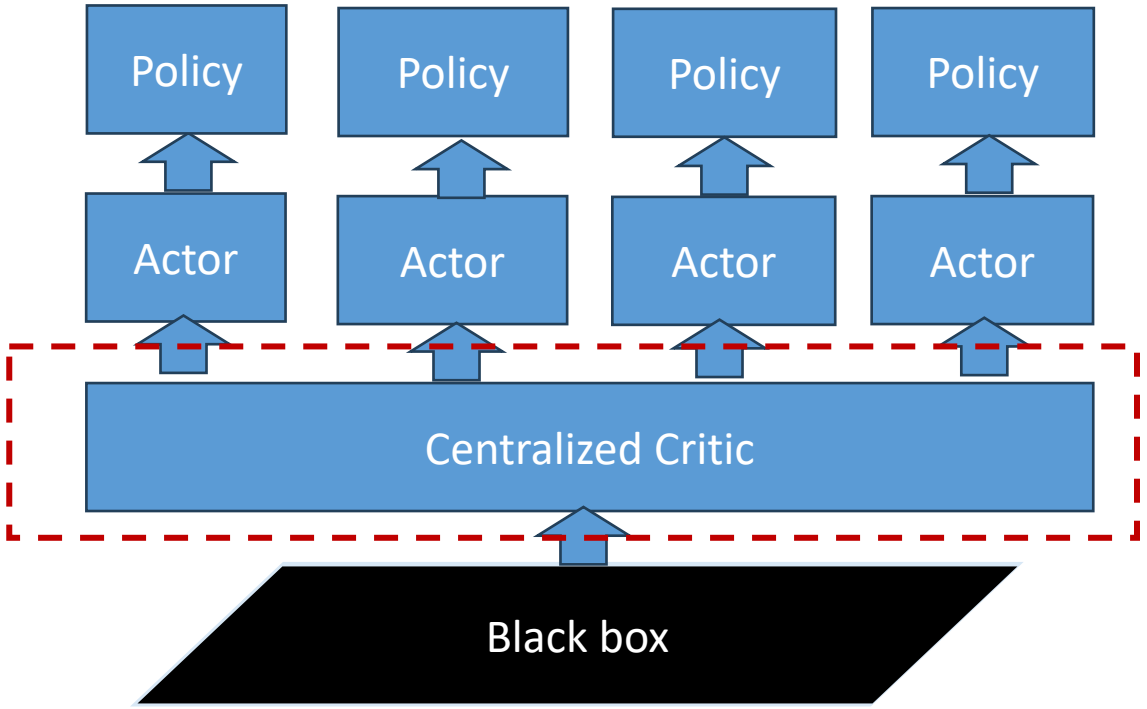
But we know there is a "network structure" underlying the system



Road Map



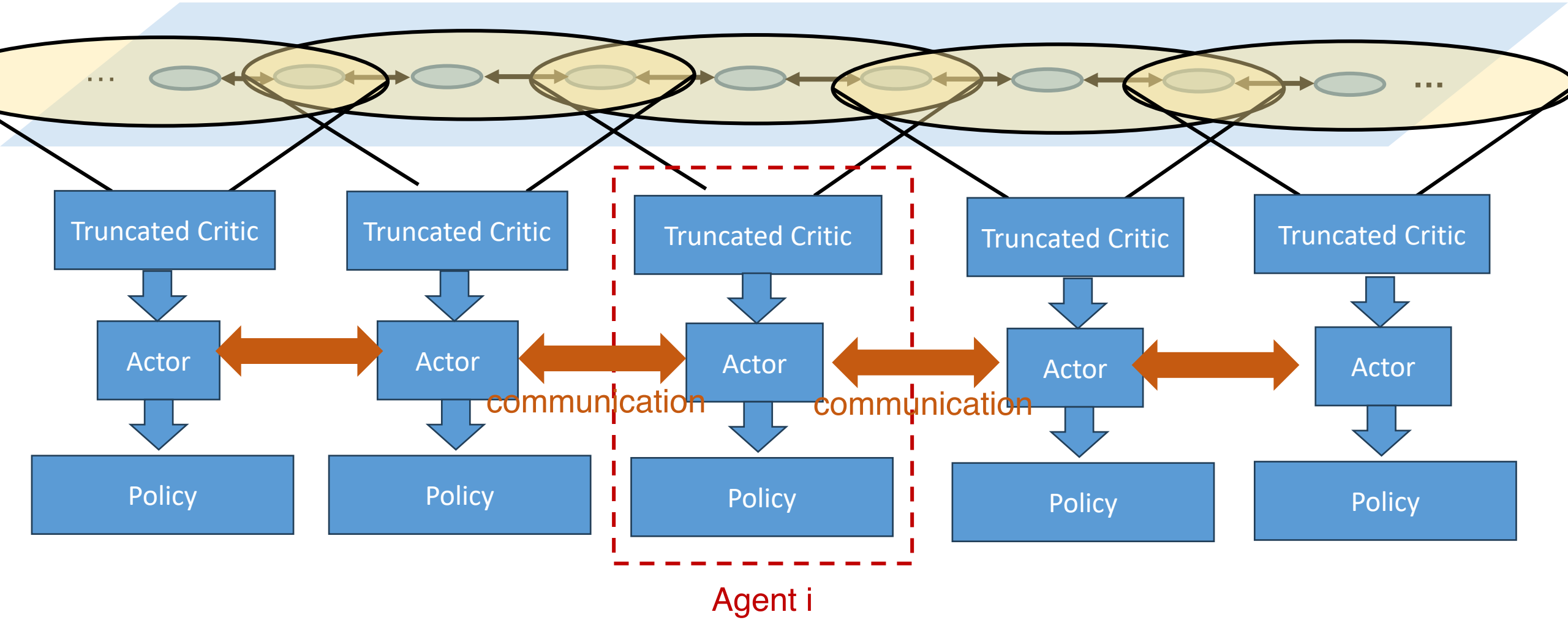
SOTA: Centralized Training, Decentralized Execution/Play



Key idea

Leveraging **truncated Q-function** to make the centralized critic distributed!

only depends on k-hop neighborhood



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 $r_i(t)$.

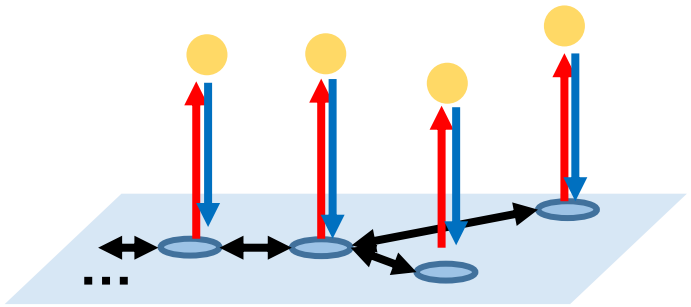
$\hat{Q}_i(s_{N_i^k}(t-1), a_{N_i^k}(t-1)) \leftarrow$ //TD-update for truncated Q function

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

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

Main ideas so far



Exponential Decay Property

Truncated Q-function

Scalable Actor Critic Alg.

Optimality guarantee?

Optimality Guarantee

Theorem [Qu, Wierman, Li 2019]. 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

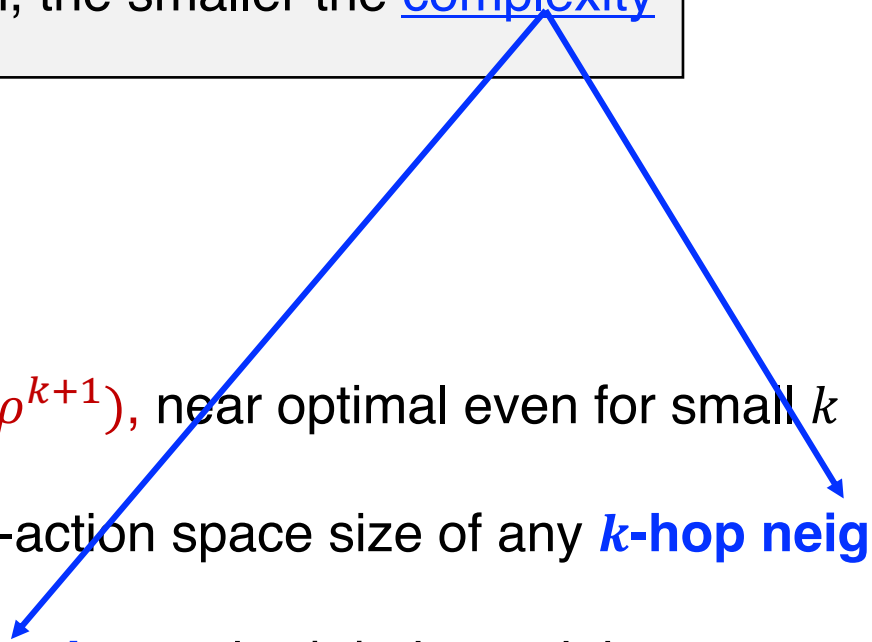
Optimality Guarantee

Role of k?
Trades off between optimality and complexity

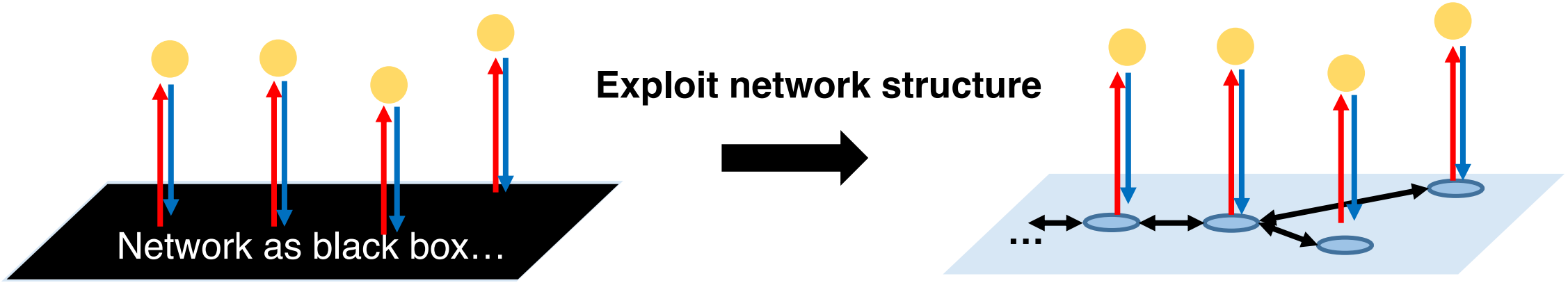
- Reaches **steady state error of $\varepsilon_k = O(\rho^{k+1})$**, close to zero even for small k
- Complexity scales with the largest state-action space size of any **k-hop neighborhood**
- Communication with **k-hop neighborhoods** required during training

Optimality Guarantee

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

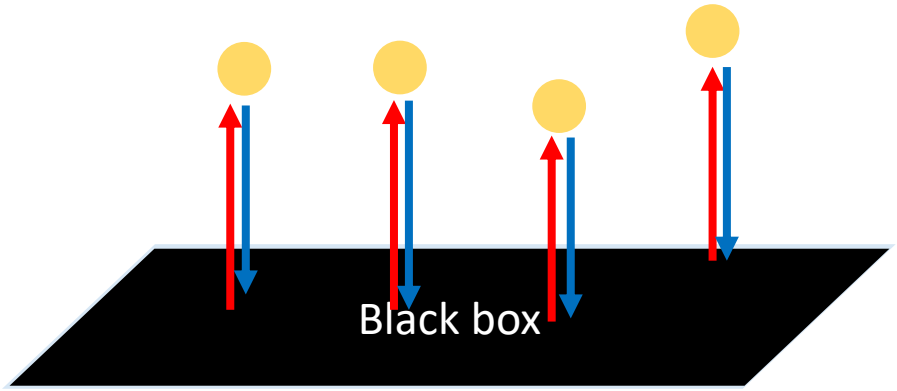


- Reaches **steady state error of** $\varepsilon_k = O(\rho^{k+1})$, near optimal even for small k
- Complexity scales with the largest state-action space size of any **k-hop neighborhood**
- Communication with **k-hop neighborhoods** required during training

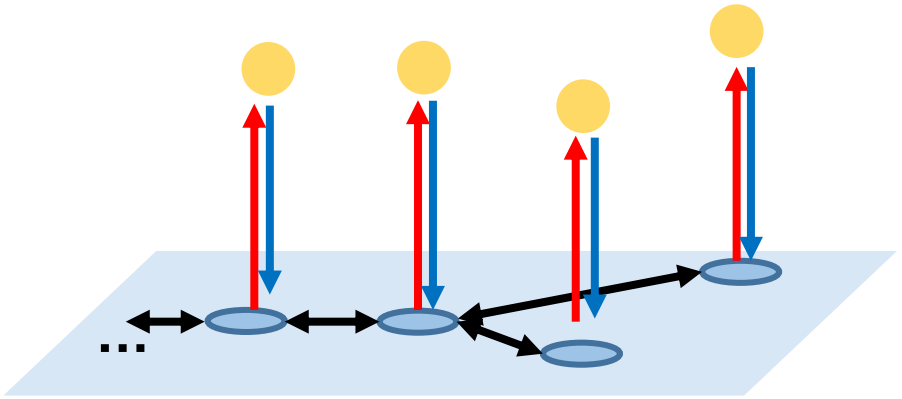


Scalable RL for networked systems by exploiting network structure!

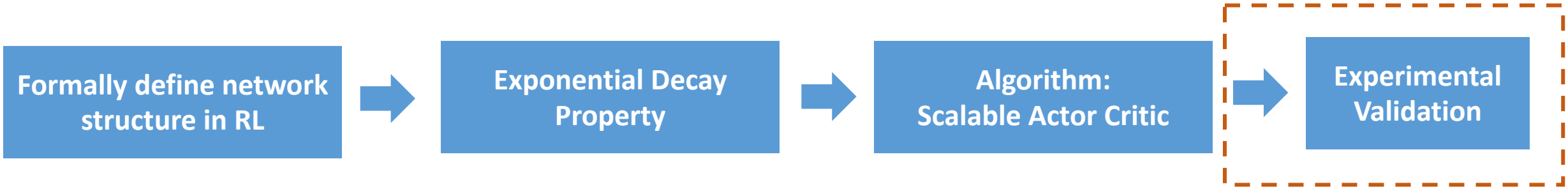
Off-the-shell RL treats the system as blackbox…



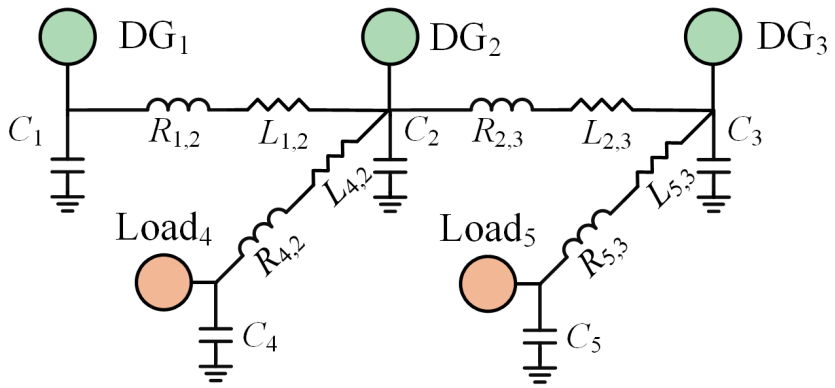
But we know there is a "network structure" underlying the system



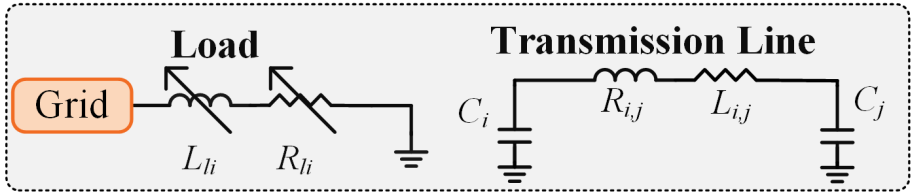
Road Map



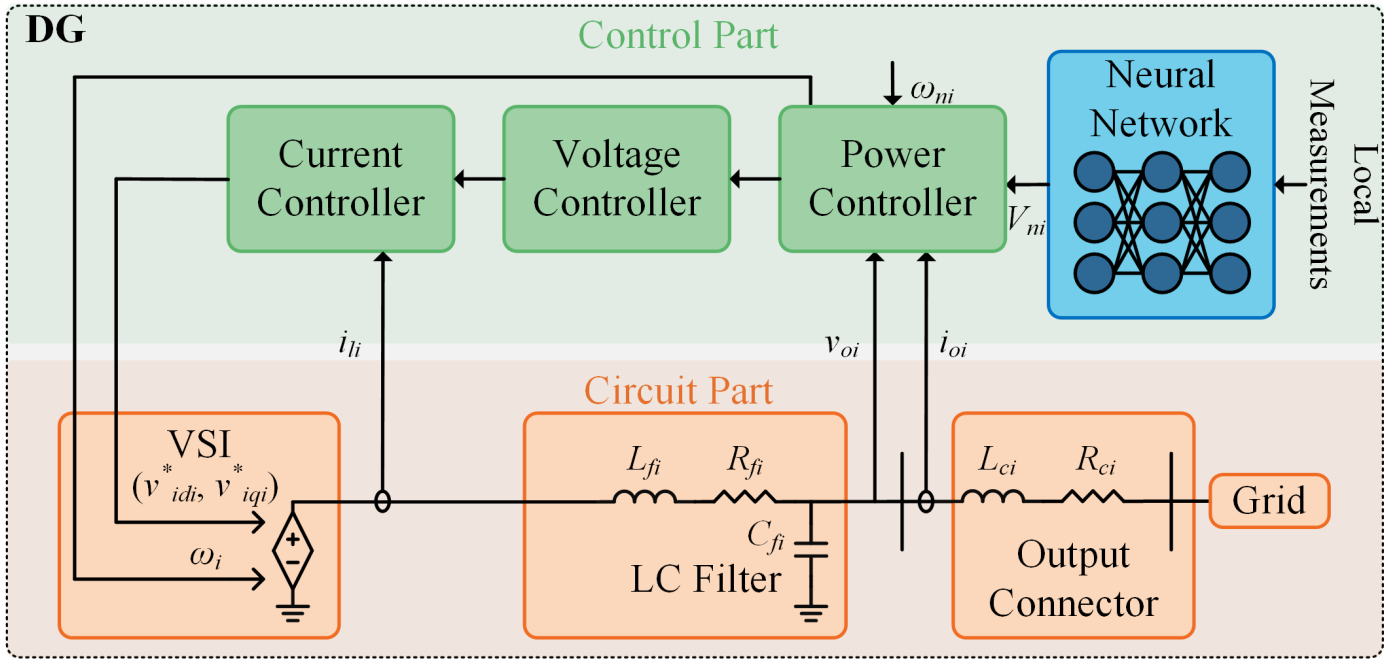
Microgrid Voltage Control



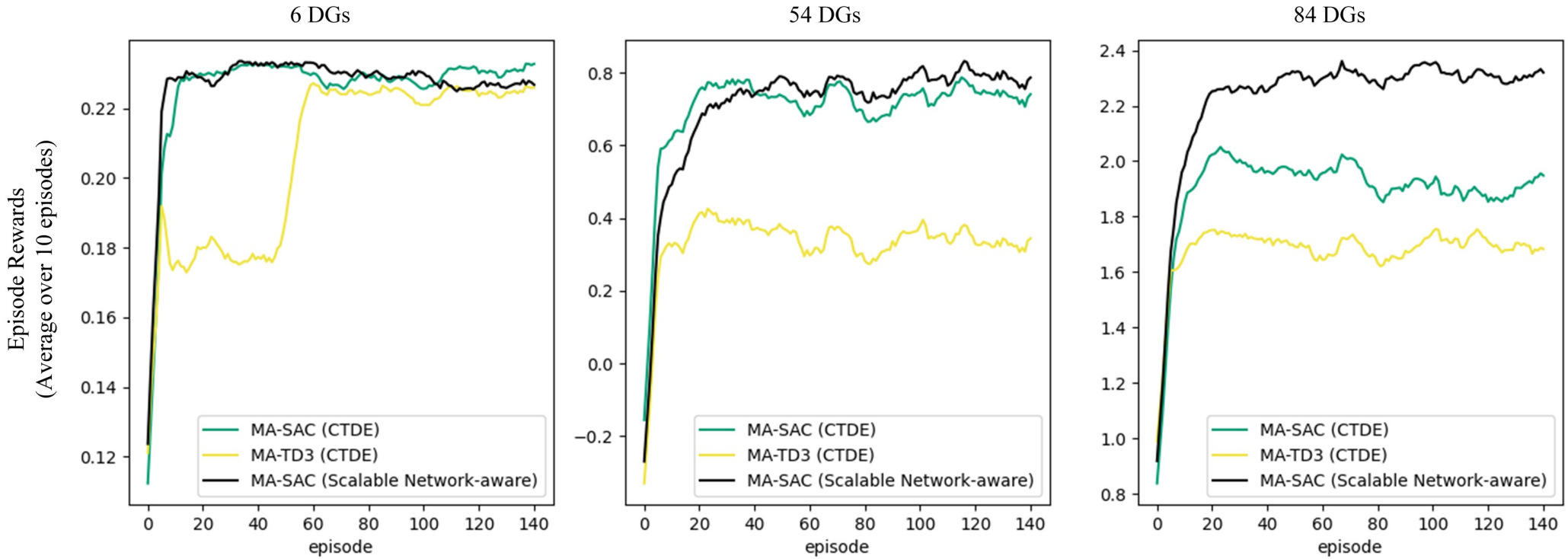
(a) Diagram of a Microgrid



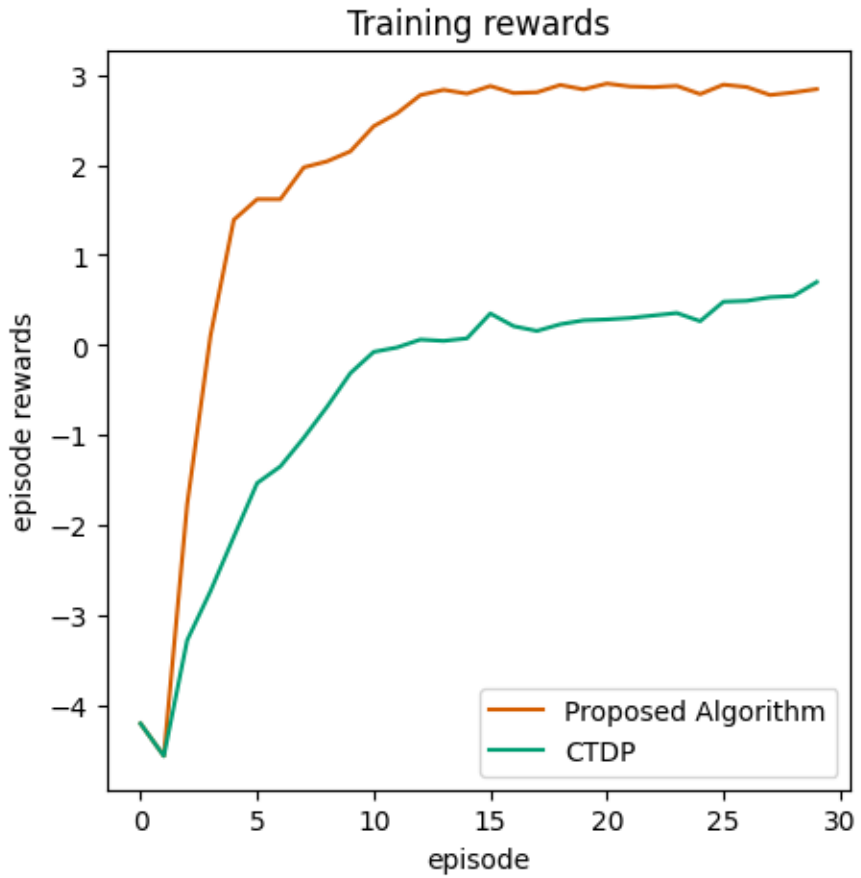
(b) Load and transmission line models



(c) DG model


6 DGs
54 DGs
84 DGs
Episode Rewards
(Average over 10 episodes)
episode
MA-SAC (CTDE)
MA-TD3 (CTDE)
MA-SAC (Scalable Network-aware)

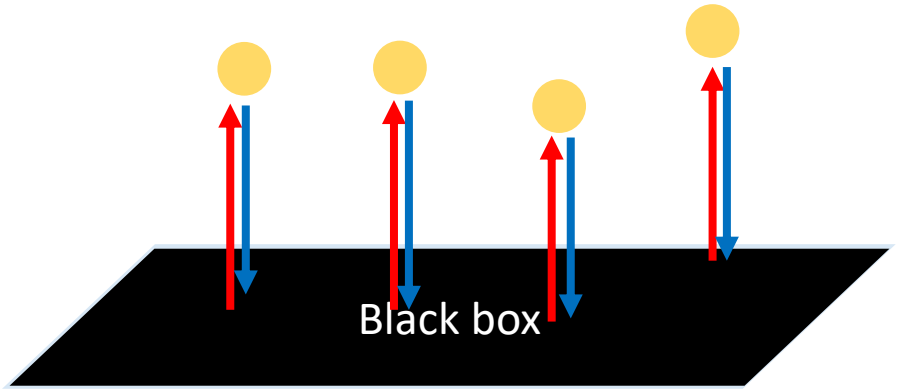
120 DGs



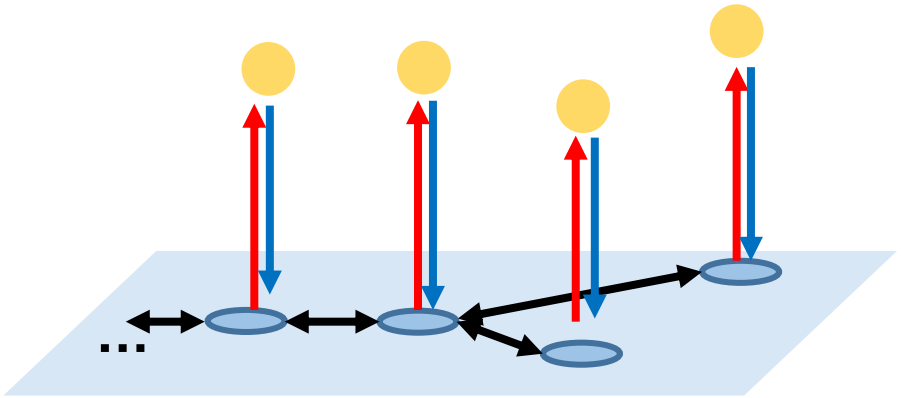
We can efficiently train on 120 DGs, which is ***larger than 100 agents***

For similar microgrid control problems, ***the SOTA is 40 DGs***

Off-the-shell RL treats the system as blackbox…



But we know there is a "network structure" underlying the system



Exploit network structure to do RL in a **scalable** manner!