強化学習における多様性

恐神貴行

IBM Research – Tokyo

強化学習と関連技術のゲームと実社会における成功の歴史

チェッカー (1956)

www.ibm.com/ibm/history/ibm100/us/en/icons/ibm700series/impacts/

バックギャモン (1992)

researcher.watson.ibm.com/researcher/view_page.php?id=6853

チェス (1997)

www.ibm.com/investor/att/pdf/BAML-AI-Conference-09272018.pdf

ビデオゲーム (2015)

en.wikipedia.org/wiki/Breakout_(video_game)

囲碁 (2016)

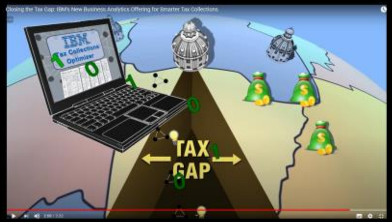
deepmind.com/alphago-korea

ポーカー (2017)

カタログ送付 (1960)

chicagology.com/sears/searscatalogs02/

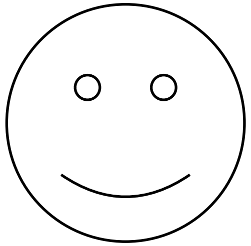
徴税支援 (2010)

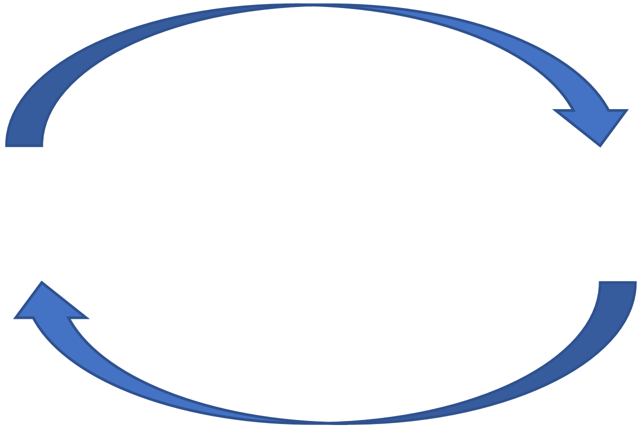


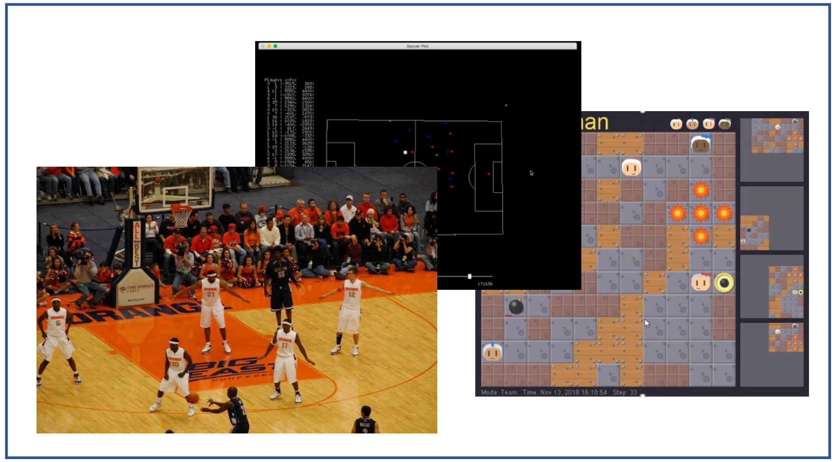
www.youtube.com/watch?v=VGKp13APsBg

Reinforcement learning seeks to find an optimal policy for sequential decision making

Action (A_t)







Agent

$\pi: S_t \rightarrow A_t$

Observation (S_t)

Reward (R_t)

Environment

$S_t, A_t \rightarrow R_t, S_{t+1}$

Goal:

Maximize expected cumulative reward

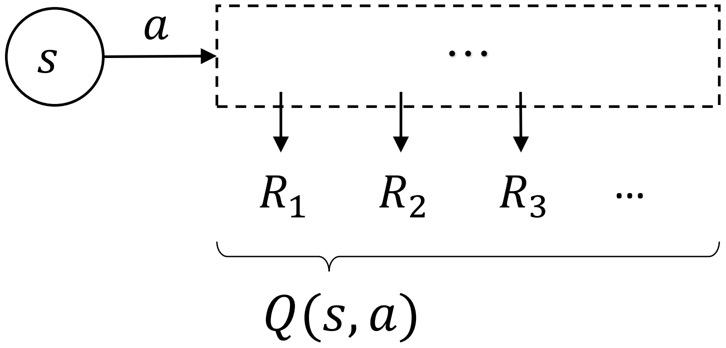
$$\sum_t \gamma^t \, \mathbf{E}^{\pi}[R_t]$$

An approach of reinforcement learning is to learn the action-value function

Action-value function: $Q(s, a)$

- Expected cumulative reward from state s by taking action a and then following optimal policy

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q(s',a')$$



- Assume (relaxed later): Markovian s is fully observable

For most practical tasks, the action-value function needs to be approximated

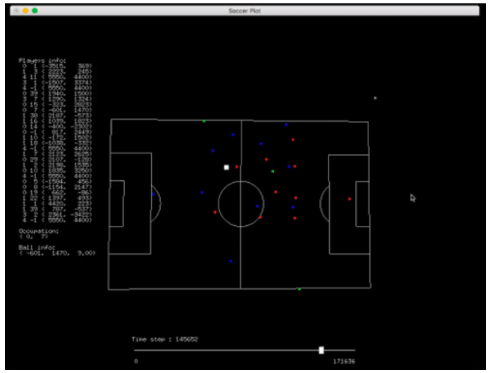
Exponentially large state space

- Combination of multiple factors
  - *e.g.* Factor: “state” of each position

- History of observations

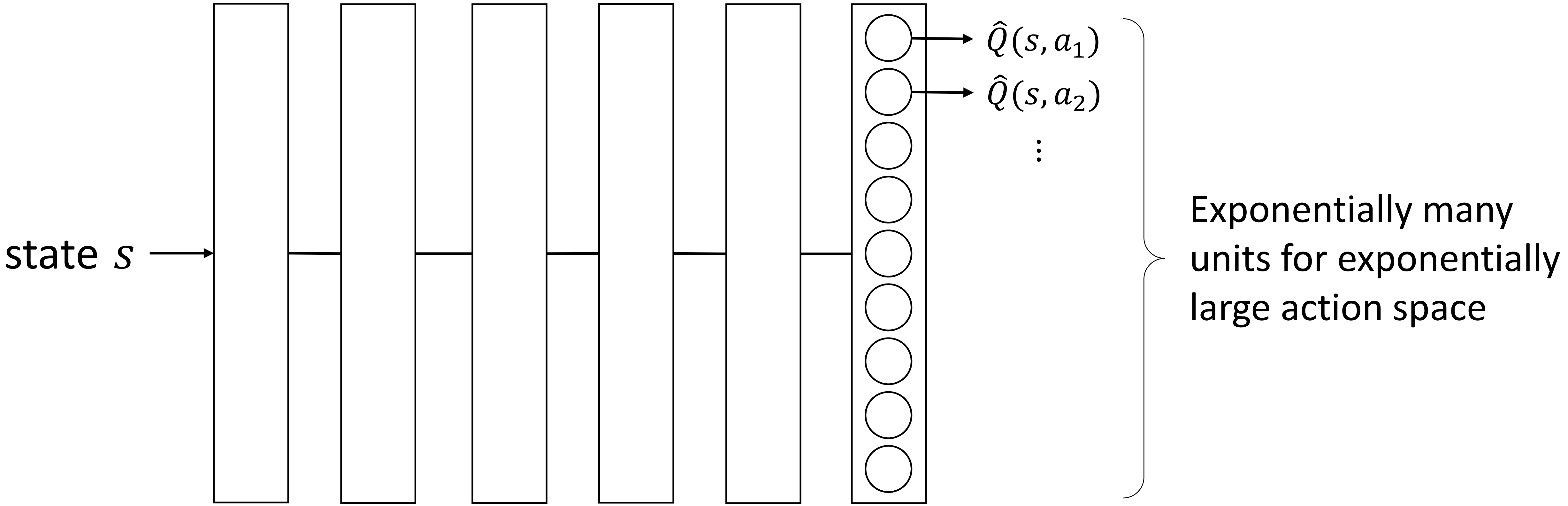
Exponentially large action space

- Combination of multiple “levers”
- Combination of multiple agents



Cannot deal with $Q(s, a)$ in tabular form

Action-value functions approximated with (deep) neural networks



Distributed representation
(*e.g.* each unit for each pixel)

Challenges in collaborative multi-agent reinforcement learning

- Exponentially many combinations of actions (team-actions)

1. How to efficiently evaluate the value of team-actions
2. How to efficiently sample good team-actions

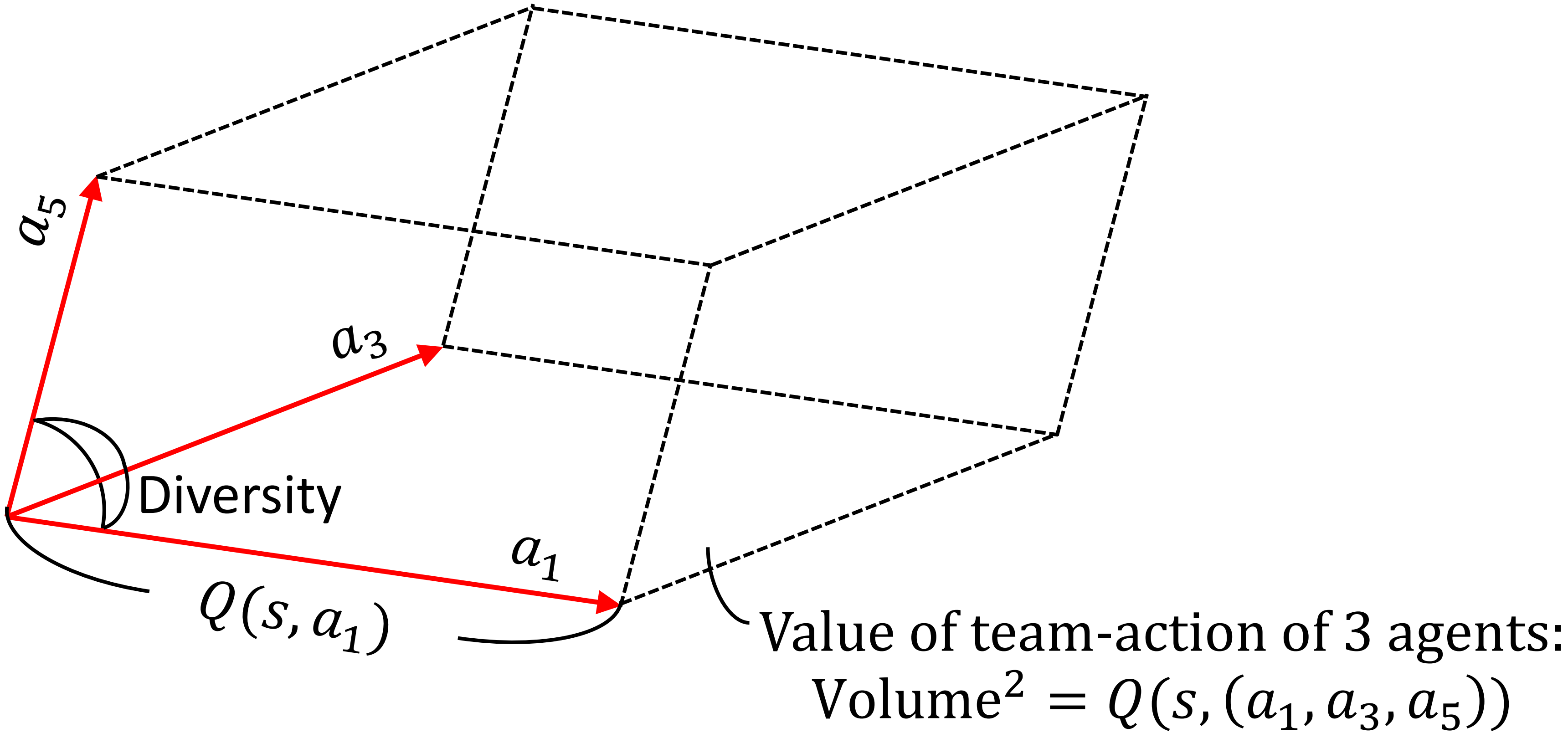
Taking into account diversity in reinforcement learning

Want to take relevant and diverse actions in team sports

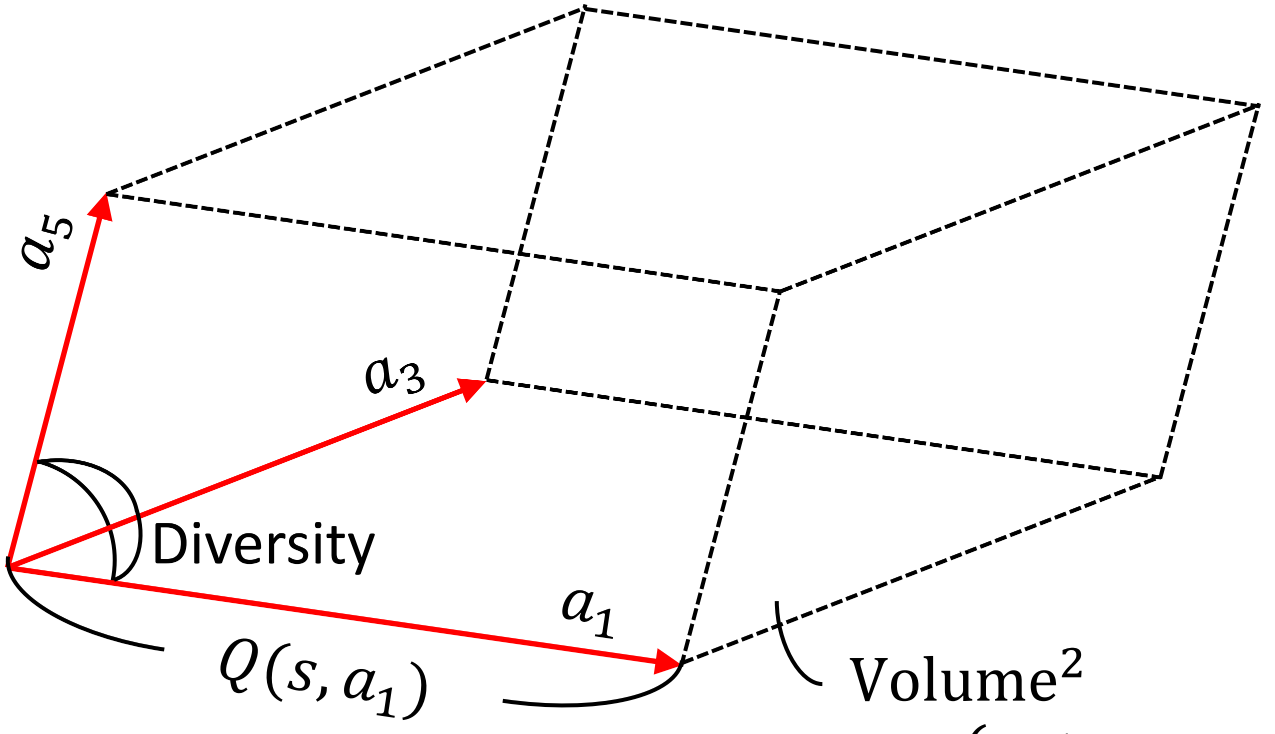
Zone defense

Man-to-man defense

Consider the diversity of actions, in addition to the value (relevance) of each action



Diversity can be represented by determinant



$$\text{Volume}^2 = Q\big(s, (a_1, a_3, a_5)\big) = \det\left(\mathbf{V}(\mathbf{x})\,\mathbf{V}(\mathbf{x})^\top\right)$$

$$\mathbf{V} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} \boxed{v_{11}\ v_{12}\ v_{13}} \\ v_{21}\ v_{22}\ v_{23} \\ \boxed{v_{31}\ v_{32}\ v_{33}} \\ v_{41}\ v_{42}\ v_{43} \\ \boxed{v_{51}\ v_{52}\ v_{53}} \\ \vdots \end{pmatrix}$$

$$\mathbf{L} = \mathbf{V}\,\mathbf{V}^\top$$

$$\mathbf{x} = (1, 0, 1, 0, 1, \ldots) \longleftarrow \text{Indicate selected actions}$$

$$\mathbf{L}(\mathbf{x}) = \mathbf{V}(\mathbf{x})\,\mathbf{V}(\mathbf{x})^\top$$

Our definition of diversity (similarity) in multi-agent reinforcement learning

- The value of team-action is represented by determinant (volume)

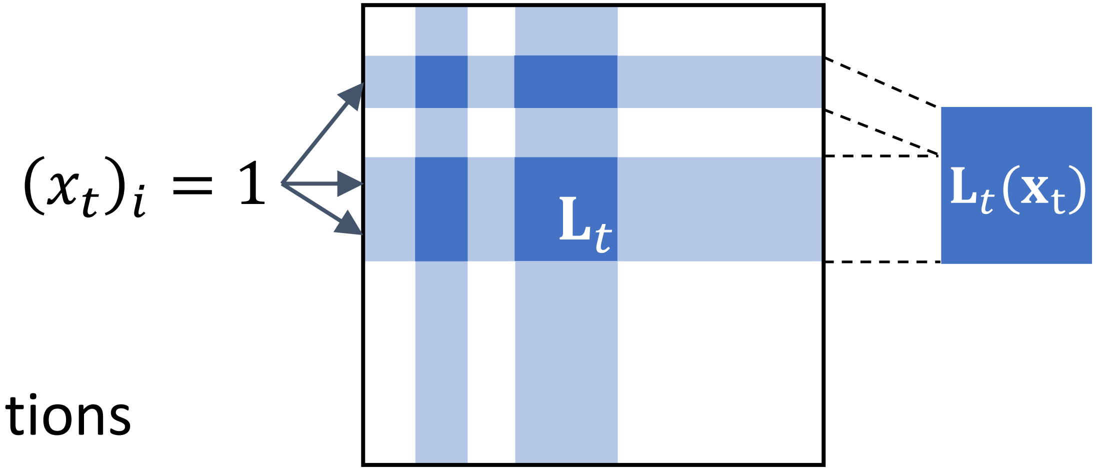
Two actions are **similar** $\Longleftrightarrow$ Value is **low** when the two actions are taken together

Two actions are **dissimilar** $\Longleftrightarrow$ Value is **high** when the two actions are taken together

$\Rightarrow$ We will learn the similarity between actions accordingly

We represent the action-value function with determinant

$$\boxed{Q_\theta(\mathbf{z}_{\le t}, \mathbf{x}_t) = \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t)}$$



- $\mathbf{z}_{\le t} \equiv (\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_t)$: Time−series of observations
  - $\mathbf{z}_{\le t} = s_t$ if Markovian state s_t is observable
- $\mathbf{x}_t \equiv \psi(a_t) \in \{0,1\}^N$: Binary features of team-action a_t
  - *e.g.* $\mathbf{x}_t$ indicates which actions are selected by the team
- $\mathbf{L}_t$: Positive semi-definite matrix (kernel) that can depend on $\mathbf{z}_{\le t}$

Particular structure of the kernel for effective learning

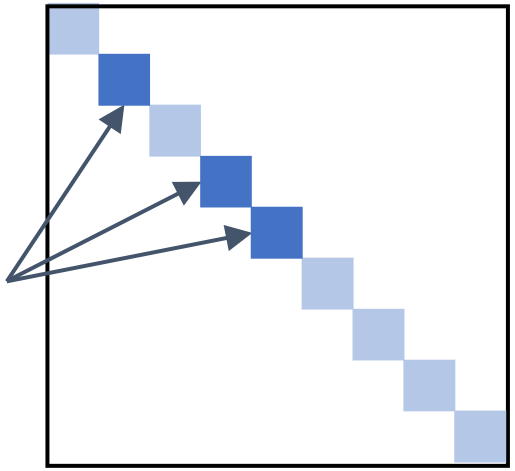
$$\boxed{Q_\theta(\mathbf{z}_{\le t}, \mathbf{x}_t) = \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t)}$$

- $\mathbf{L}_t \equiv \mathbf{V}\,\mathbf{D}_t\,\mathbf{V}^\top$
- $\mathbf{V}$: $N \times K$ matrix ($K \le N$)
- $\mathbf{D}_t \equiv \mathrm{Diagonal}\big(\exp\big(\mathbf{d}_t(\phi)\big)\big)$
- $\mathbf{d}_t(\phi)$: differentiable time−series model with parameter ϕ
  (e.g. RNN, LSTM, DyBM, VAR)

Special case of diagonal kernel reduces to the standard approach of ignoring diversity

- $\mathbf{L}_t = \mathbf{D}_t$ (let $\mathbf{V} = \mathbf{I}$)
- $\mathbf{D}_t \equiv \mathrm{Diag}\big(\exp\big(\mathbf{d}_t(\phi)\big)\big)$
- $\mathbf{d}_t(\phi)$: differentiable time–series model

$(x_t)_i = 1$



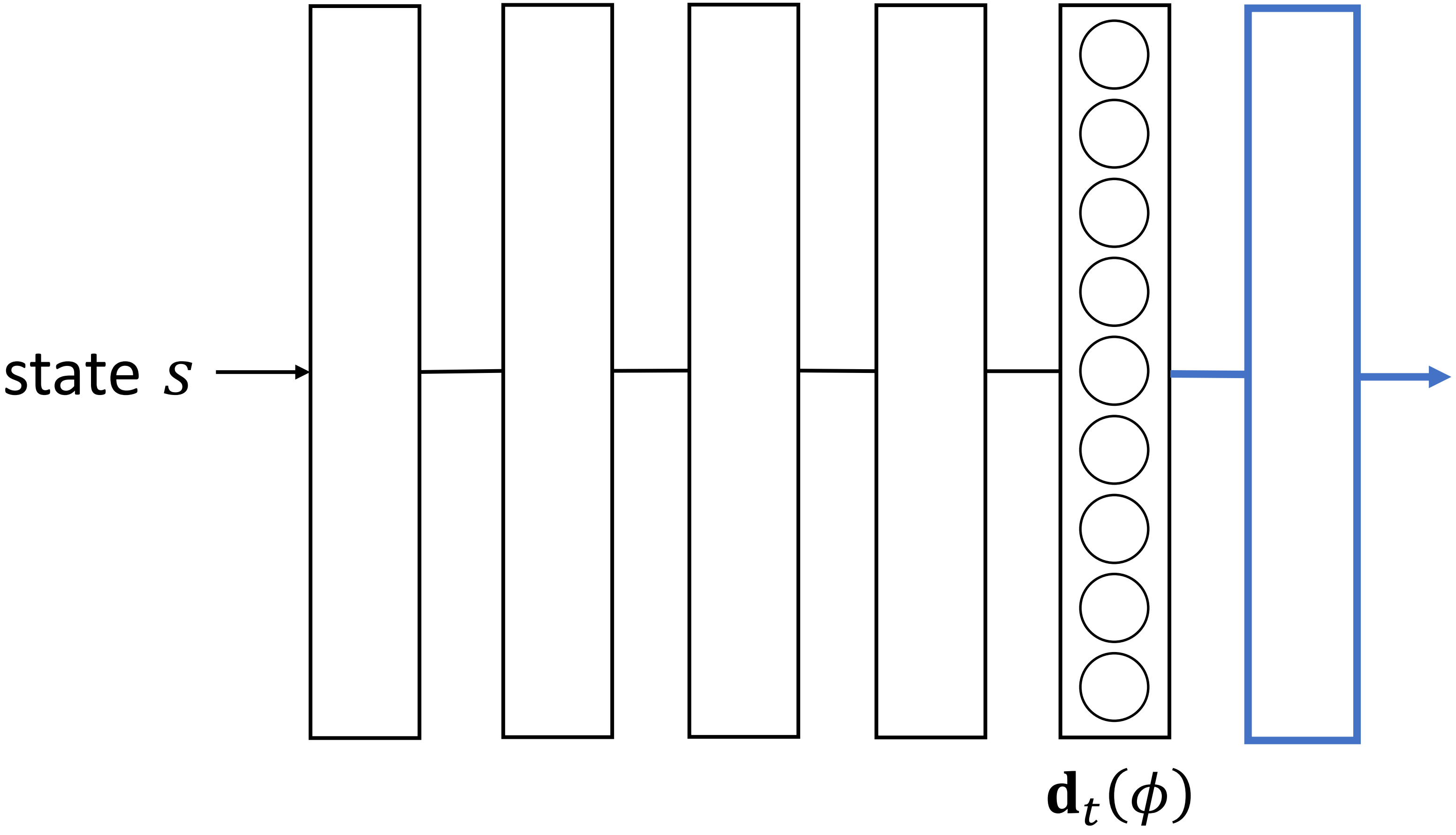
$\mathbf{L}_t$

$$
\begin{aligned}
Q_\theta(\mathbf{z}_{\le t}, \mathbf{x}_t) &= \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t) \\
&= \alpha + \mathbf{d}_t(\phi)^\top \mathbf{x}_t \\
&= \alpha + \sum_{i:(x_t)_i = 1} d_t(\phi)_i
\end{aligned}
$$

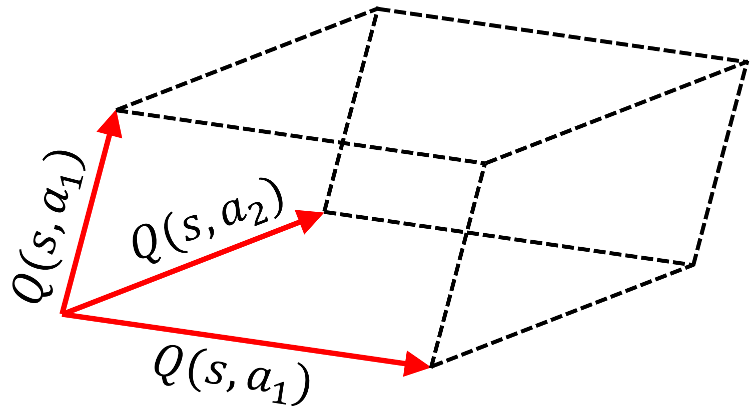
Sum of the values of selected actions

$d_t(\phi)_i$: value of a_i at time t

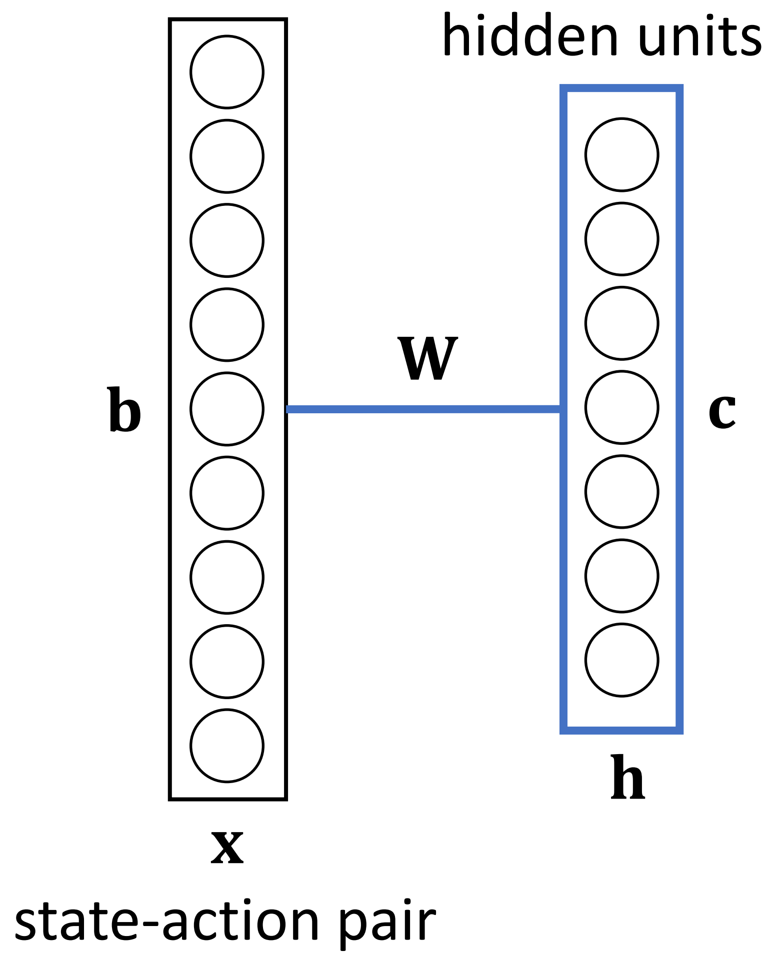
Determinantal layer for diversity



$$Q_\theta(s, \mathbf{x}_t) = \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t)$$
$$\mathbf{L}_t \equiv \mathbf{V}\,\mathrm{Diag}\big(\exp\big(\mathbf{d}_t(\phi)\big)\big)\,\mathbf{V}^\top$$



Prior work uses free-energy of restricted Boltzmann machines [Sallans & Hinton 2001]



$$F(\mathbf{x}) = -\log \sum_{\tilde{\mathbf{h}}} \exp\left(-E(\mathbf{x}, \tilde{\mathbf{h}})\right)$$

$$E(\mathbf{x}, \mathbf{h}) \equiv -\mathbf{b}^\top \mathbf{x} - \mathbf{c}^\top \mathbf{h} - \mathbf{x}^\top \mathbf{W} \mathbf{h}$$

No hidden units

$$F(\mathbf{x}) = -\mathbf{b}^\top \mathbf{x}$$

Bias $\mathbf{b}$ represents individual value of each action

Learning diversity via reinforcement learning

Reinforcement learning with SARSA

- Tabular case

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta \, \mathrm{TD}_t$$

$$\text{where } \mathrm{TD}_t \equiv \underbrace{r_{t+1} + \rho \, Q(s_{t+1}, a_{t+1})}_{\text{Cumulative reward from } t} - \underbrace{Q(s_t, a_t)}_{\text{Cumulative reward from } t}$$

- With functional approximation: $Q_\theta(s, a) \approx Q(s, a)$

$$\theta \leftarrow \theta + \eta \, \mathrm{TD}_t \, \nabla_\theta Q_\theta(s_t, a_t)$$

Can learn our kernel $\mathbf{L}_t$ via SARSA in an end-to-end manner

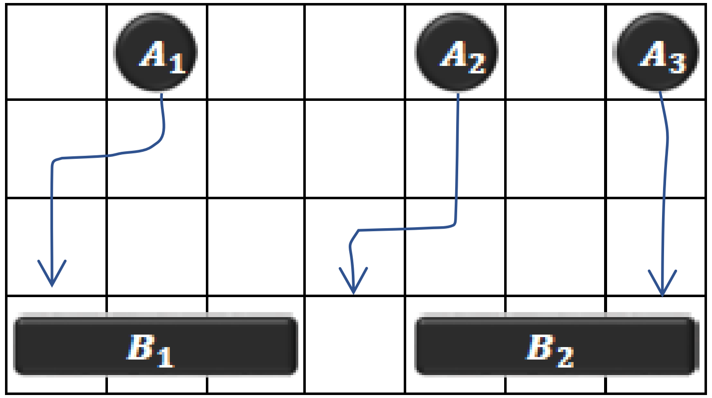
- $Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t)$
  - $\mathbf{L}_t \equiv \mathbf{V}\,\mathbf{D}_t\,\mathbf{V}^\top$
  - $\mathbf{D}_t \equiv \mathrm{Diagonal}\big(\exp\big(\mathbf{d}_t(\phi)\big)\big)$

$$
\begin{aligned}
&\nabla_\alpha Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = 1 \\
&\nabla_{\mathbf{V}(\bar{\mathbf{x}}_t)} Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = \mathbf{0} \\
&\nabla_{\mathbf{V}(\mathbf{x}_t)} Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = 2\,(\mathbf{V}(\mathbf{x}_t)^+)^\top \\
&\nabla_\phi Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = \mathrm{diag}(\mathbf{V}(\mathbf{x}_t)^+\,\mathbf{V}(\mathbf{x}_t))\,\nabla_\phi \mathbf{d}_t(\phi)
\end{aligned}
$$

$$\theta \leftarrow \theta + \eta\,\mathrm{TD}_t\,\nabla_\theta Q_\theta(s_t, a_t)$$

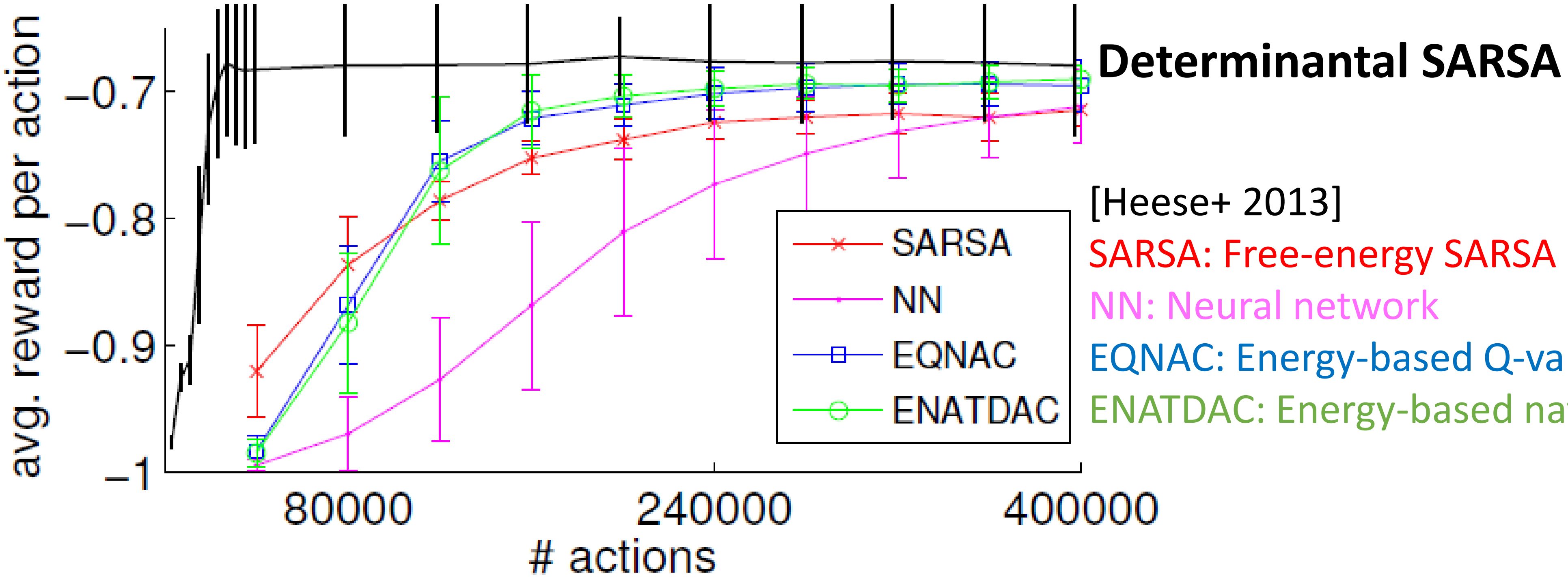
Example: Blocker Task

- Reward
  - $+1$ when an attacker reaches the end zone
  - -1 each step



- We use target positions of agents as the feature of state-action pair
  - $\mathbf{x}_t \equiv \psi(a_t) = s_{t+1} \in \{0, 1\}^{21}$

Determinantal SARSA finds a nearly optimal policy 10 times faster than baseline methods



[Heese+ 2013]
SARSA: Free-energy SARSA
NN: Neural network
EQNAC: Energy-based Q-value natural actor critic
ENATDAC: Energy-based natural TD actor critic

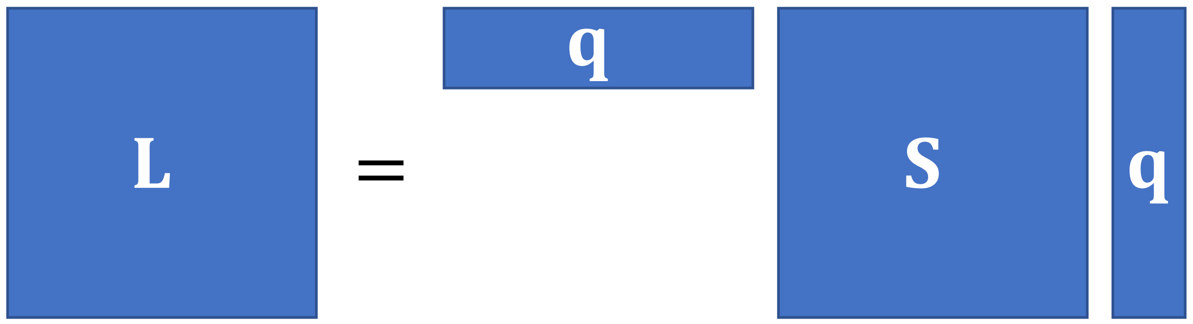
Quality-similarity decomposition of the kernel

- Value, relevance, quality

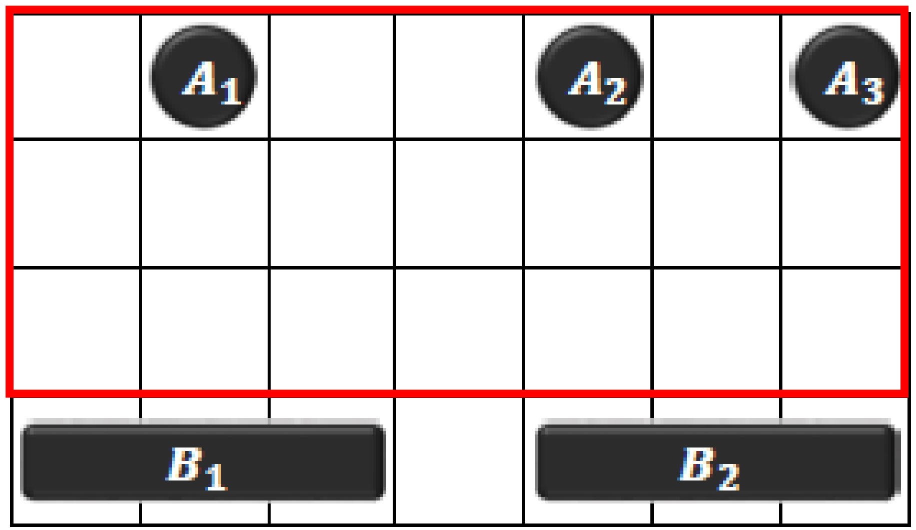
$$q_i = \sqrt{L_{i,i}}$$

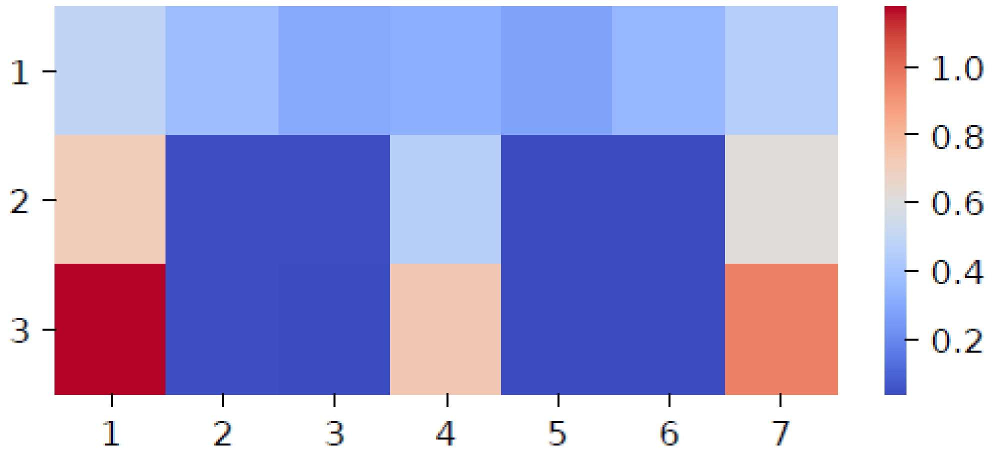
- Similarity

$$S_{i,j} = \frac{L_{i,j}}{\sqrt{L_{i,i}\, L_{j,j}}}$$

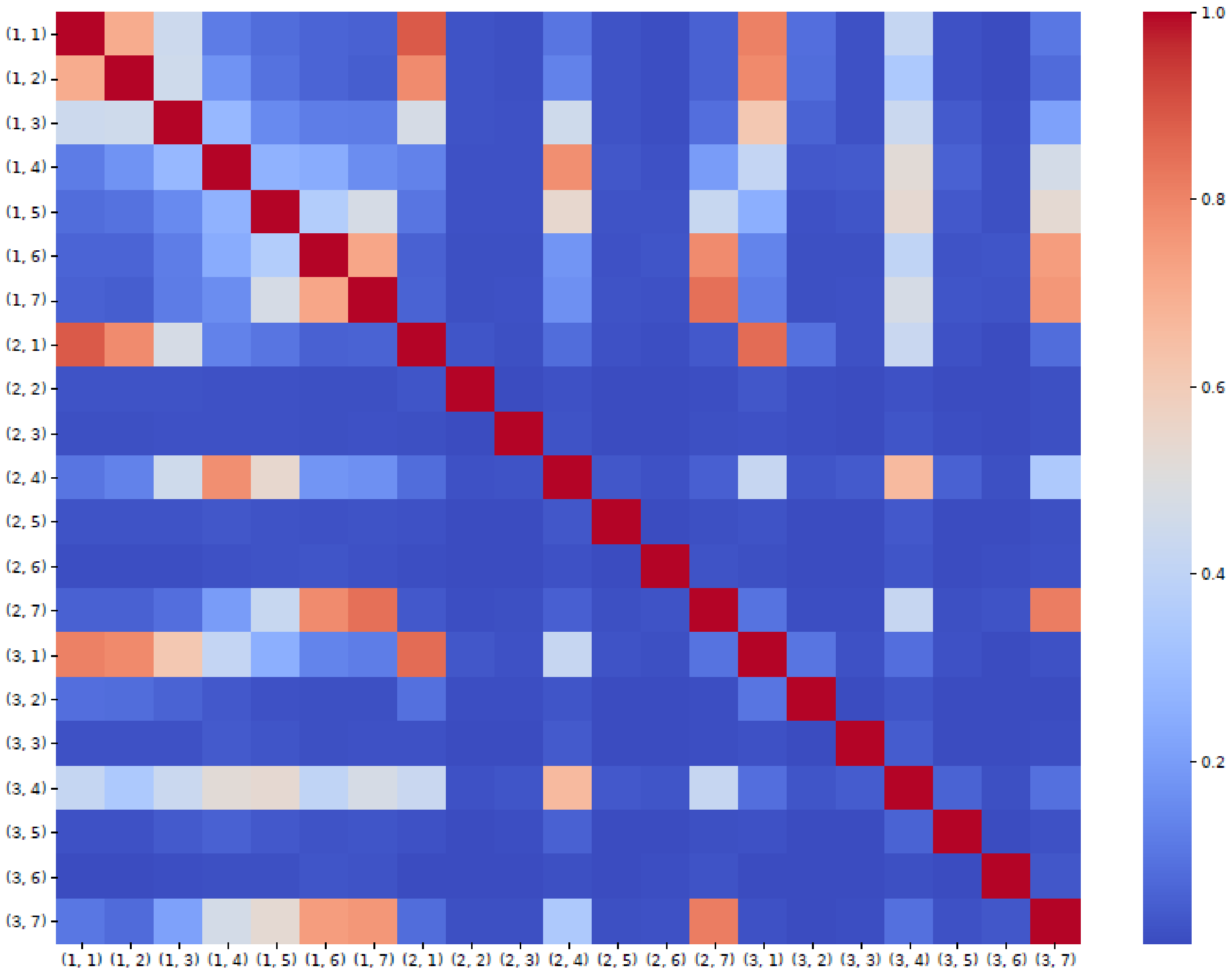


Value of individual actions (next positions) learned by Determinantal SARSA





Similarity between actions (next positions) learned by Determinantal SARSA



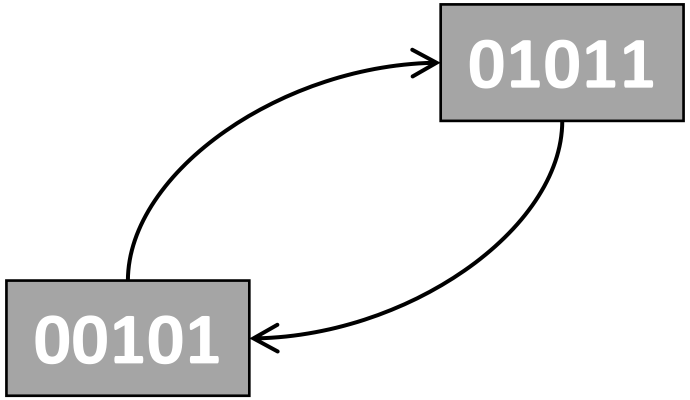
Recall:
Our definition of action similarity

similar ⇔ **low** value when taken together

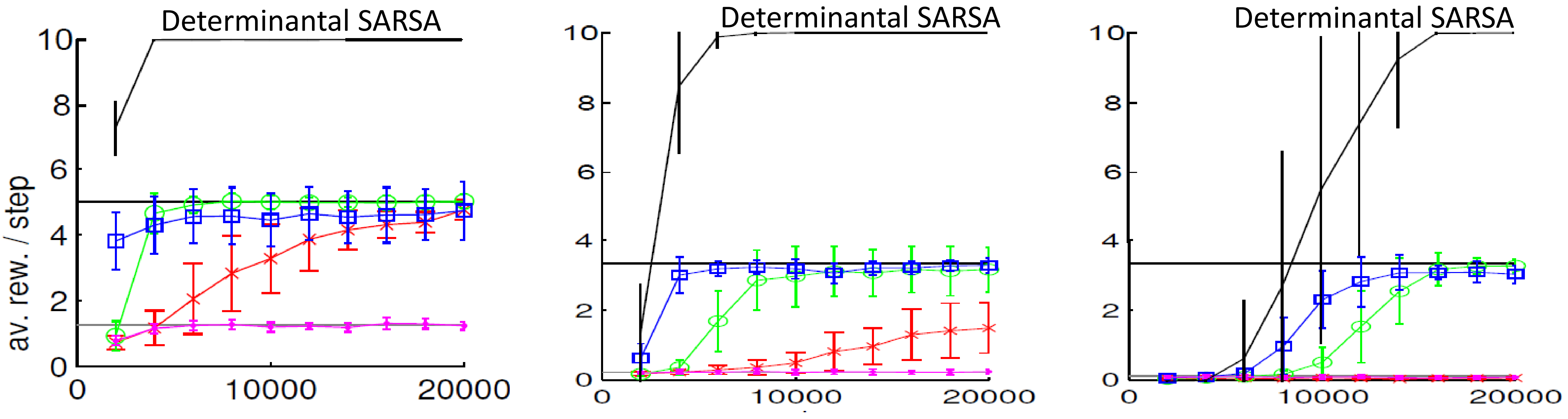
dissimilar ⇔ **high** value when taken together

Stochastic Policy Task

- 2^N actions
- If action matches states
    - Get $+10$ reward
    - Hidden state transitions



Determinantal SARSA finds nearly optimal policies, while baselines suffer from partial observability



(a) Task 1 ($N = 5$) (b) Task 2 ($N = 6$) (c) Task 3 ($N = 8$)

Baselines from [Heese+ 2013]

SARSA: Free-energy SARSA
NN: Neural network
EQNAC: Energy-based Q-value natural actor critic
ENATDAC: Energy-based natural TD actor critic

Choosing diverse actions

Want to sample actions having high value with high probability

- ε-greedy
  - Uniformly at random with probability ε
  - Best action ($a^\star = \text{argmax}_a\, Q(s, a)$) with probability $1 - \varepsilon$

Intractable for large action space

- Boltzmann exploration
  - Take action a with probability $\sim \exp(-\beta\, Q(s, a))$
  - β: inverse temperature

Choosing a diverse team-action with Boltzmann exploration

$\mathbf{x}_t \equiv \psi(a_t) \in \{0,1\}^N$:
Binary features of team-action a_t

- Our value function:

$$Q_\theta(\mathbf{z}_{\le t}, \mathbf{x}_t) = \alpha + \log\det \mathbf{L}_t(\mathbf{x}_t)$$

- Team-action selected according to Boltzmann exploration

$$\pi(\, \mathbf{x}_t \mid \mathbf{z}_{\le t}\,) = \frac{\exp\big(\beta Q_\theta(\mathbf{z}_{\le t}, \mathbf{x}_t)\big)}{\sum_{\tilde{\mathbf{x}}} \exp\big(\beta Q_\theta(\mathbf{z}_{\le t}, \tilde{\mathbf{x}}\,)\big)} = \frac{\det \mathbf{L}_t(\mathbf{x}_t)^\beta}{\sum_{\tilde{\mathbf{x}}} \det \mathbf{L}_t(\tilde{\mathbf{x}})^\beta}$$

Boltzmann exploration can be performed efficiently when $\beta = 1$

$$\pi(\mathbf{x}_t \mid \mathbf{z}_{\leq t}) = \frac{\det \mathbf{L}_t(\mathbf{x}_t)}{\sum_{\tilde{\mathbf{x}}} \det \mathbf{L}_t(\tilde{\mathbf{x}})} = \frac{\det \mathbf{L}_t(\mathbf{x}_t)}{\det(\mathbf{L}_t + \mathbf{I})}$$

Sum over 2^N terms

Determinant of $N \times N$ matrix

Nearly exact Boltzmann exploration with MCMC [Kang 2013 (for $\beta = 1$)]

1. Initialize $\mathbf{x}$
2. Repeat
   - $\mathbf{x} \leftarrow \mathbf{x}'$ with probability $\min\left\{1, \left(\frac{\det \mathbf{L}_t(\mathbf{x}')}{\det \mathbf{L}_t(\mathbf{x})}\right)^\beta\right\}$

- When $\mathbf{x}$' and $\mathbf{x}$ differs by one bit, $\det \mathbf{L}_t(\mathbf{x}')$ can be computed from $\det \mathbf{L}_t(\mathbf{x})$ via rank-one update techniques
  - Assume we can find $a \leftarrow \psi^{-1}(\mathbf{x})$

Simpler approaches: Heuristics for more exploration and exploitation

- For more exploration
  - Uniform with probability ε
  - DPP with probability $1 - \varepsilon$
- For more exploitation
  - Sample M actions according to DPP
  - Choose the best among M
- Hard-core point processes [Matérn 1986]

Any of these approaches are possible once we learn $\mathbf{L}_t$

Centralized Training and Decentralized Execution

So far,

Agents are trained and executed with central control

[Osogami & Raymond, AAAI-19]

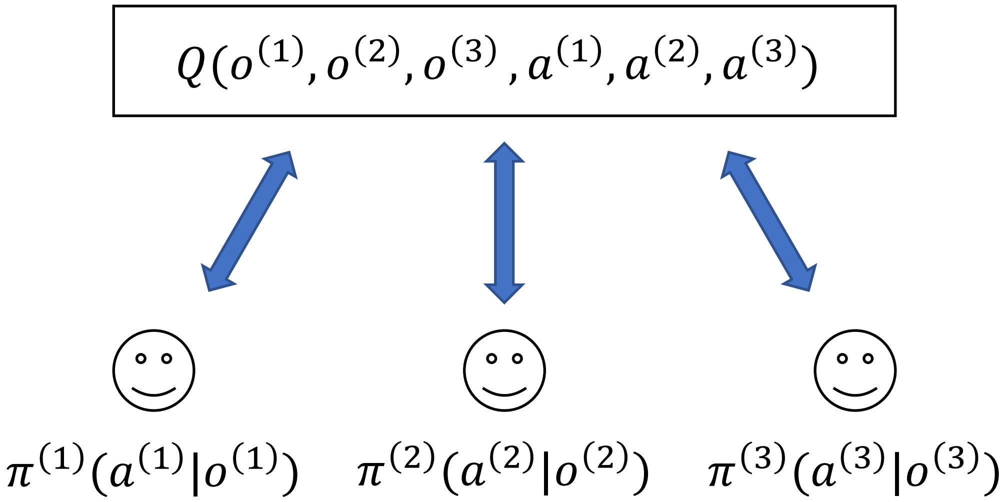
Recent trend is

Centralized Training and Decentralized Execution

[Yang et al., ICML 2020]

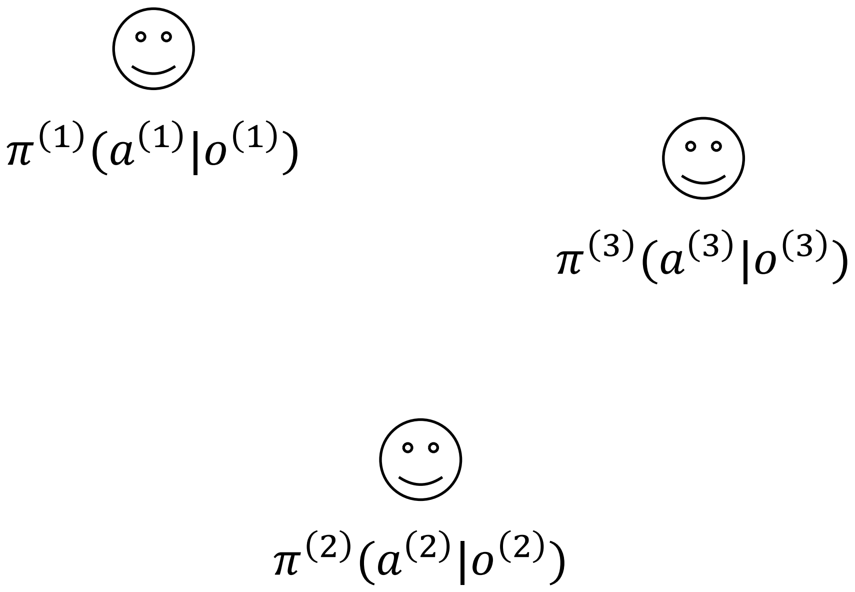
Centralized Training and Decentralized Execution [Lowe+ 2017, Foerster+ 2017]

Centralized training



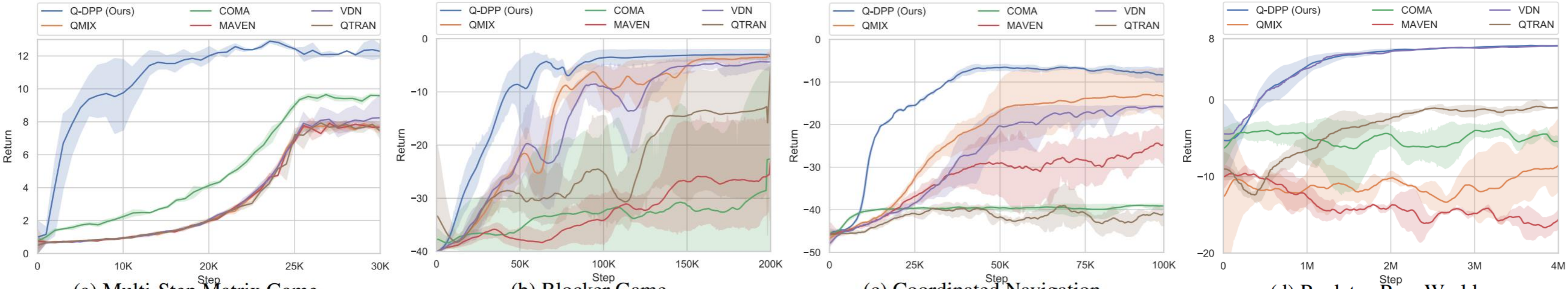
$$\nabla \log \pi^{(i)}\left(a^{(i)} \middle| o^{(i)}\right) \; Q(\mathbf{o}, \mathbf{a})$$

Decentralized execution



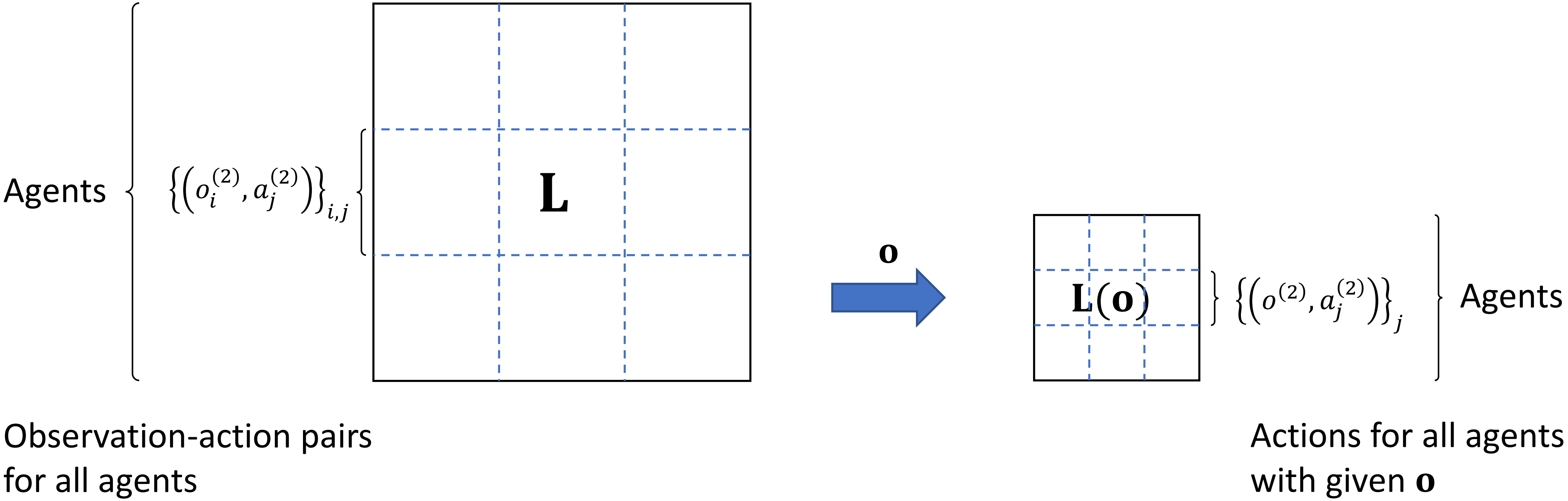
Determinantal RL achieves state-of-the-art in the settings of decentralized execution

[Yang et al., ICML 2020]



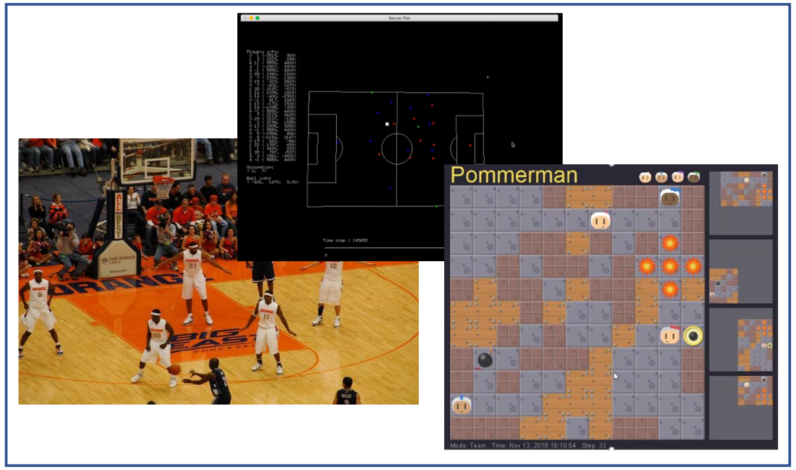
(a) Multi-Step Matrix Game (b) Blocker Game (c) Coordinated Navigation (d) Predator-Prey World

Kernel studied in Yang et al., ICML 2020



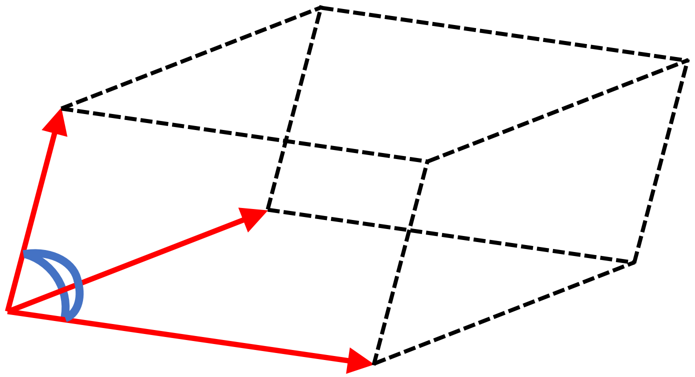
Summary

Need **diversity** in team-actions



Our definition of action similarity

similar	$\Longleftrightarrow$	**low** value when taken together
dissimilar	$\Longleftrightarrow$	**high** value when taken together



Challenge

Exponentially many team-actions

Our solution

$$\boxed{Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = \alpha + \log \det \mathbf{L}_t(\mathbf{x}_t)}$$

Efficient learning

Efficient sampling (*cf.* DPP)

$$\nabla_{\mathbf{V}(\mathbf{x}_t)} Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = 2\,(\mathbf{V}(\mathbf{x}_t)^+)^\top$$
$$\nabla_\phi Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t) = \mathrm{diag}(\mathbf{V}(\mathbf{x}_t)^+ \mathbf{V}(\mathbf{x}_t))\,\nabla_\phi \mathbf{d}_t(\phi)$$

$$\pi(\mathbf{x}_t \mid \mathbf{z}_{\leq t}) = \frac{\exp\big(\beta Q_\theta(\mathbf{z}_{\leq t}, \mathbf{x}_t)\big)}{\sum_{\tilde{\mathbf{x}}} \exp\big(\beta Q_\theta(\mathbf{z}_{\leq t}, \tilde{\mathbf{x}})\big)} = \frac{\det \mathbf{L}_t(\mathbf{x}_t)^\beta}{\sum_{\tilde{\mathbf{x}}} \det \mathbf{L}_t(\tilde{\mathbf{x}})^\beta}$$

References

- T. Osogami and R. Raymond, Determinantal reinforcement learning, AAAI-19
- Y. Yang et al., Multi-Agent Determinantal Q-Learning, ICML 2020