GEOMETRIC INSIGHTS FOR NONLINEAR TD CONVERGENCE

joint work with David Brandfonbrener
REINFORCEMENT LEARNING

- General framework to learn how to interact in complex, high-dimensional environments.

TD-Gammon, Tesauro’92

Deepmind’16

Deepmind’17
REINFORCEMENT LEARNING

Deepmind’17

Hide and Seek, OpenAI’19

OpenAI’19

Alphastar, Deepmind’19
Universal quantum control through deep reinforcement learning

Murphy Yuezhen Niu$^{1,2}$, Sergio Boixo$^{6}$, Vadim N. Smelyanskiy$^{2}$ and Hartmut Neven$^{2}$

Reinforcement Learning for Integer Programming: Learning to Cut

Yunhao Tang
Columbia University
yt2541@columbia.edu

Shipra Agrawal*
Columbia University
sa3305@columbia.edu

Yuri Faenza
Columbia University
yf2414@columbia.edu
Remarkable ability to discover useful policies in large environments.

High-dimensional, noisy, observations.
Remarkable ability to discover useful policies in large environments.

High-dimensional, noisy, observations.

Yet, with

- poor sample efficiency.

- limited theoretical guarantees.
MARKOV DECISION PROCESSES

- **Mathematical Setup:** \( \mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \rho) \)

  - \( \mathcal{S} \): state space (might be discrete or continuous).
  - \( \mathcal{A} \): space of actions (assumed the same for all states).
  - \( \mathcal{P}(s' | s, a) \): Markov transition probability kernel.
  - \( \rho \): initial state distribution.
  - \( R(s, a) \): instantaneous reward.
  - \( \gamma \): discount factor, assume \( 0 \leq \gamma < 1 \).
MARKOV DECISION PROCESSES

- **Mathematical Setup:** \( \mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, \rho) \)

  - \( \mathcal{S} \): state space (might be discrete or continuous).
  - \( \mathcal{A} \): space of actions (assumed the same for all states).
  - \( \mathcal{P}(s', s, a) \): Markov transition probability kernel.
  - \( \rho \): initial state distribution.
  - \( R(s, a) \): instantaneous reward.
  - \( \gamma \): discount factor, assume \( 0 \leq \gamma < 1 \).

- **Goal:** Find a policy \( \pi : \mathcal{S} \rightarrow \mathcal{A} \) that maximizes expected sum of discounted rewards:

\[
\max_{\pi} \mathbb{E}_{\rho, \mathcal{P}} \sum_{k} \gamma^k R(s_k, a_k) \quad \text{subject to} \quad s_{k+1} \sim \mathcal{P}(s' | s_k, a_k), \quad s_0 \sim \rho, \quad a_k = \pi(s_k)
\]
KEY CHALLENGES OF RL

- Exploration/Exploitation tradeoff:
  - Unknown environment, need to uncover potential rewards while exploiting known good strategies.
KEY CHALLENGES OF RL

- Exploration/Exploitation tradeoff:
  - Unknown environment, need to uncover potential rewards while exploiting known good strategies.

- Credit Assignment
  - Valid strategies may pay off at later stages.
KEY CHALLENGES OF RL

- **Exploration/Exploitation tradeoff:**
  - Unknown environment, need to uncover potential rewards while exploiting known good strategies.

- **Credit Assignment**
  - Valid strategies may pay off at later stages.

- **High-dimensional, complex observations.**
  - Need to learn good state representations.
THREE PARADIGMS OF RL

- **Model-based**: estimate dynamics $\hat{P}$ and then solve the resulting optimal control problem (planning and simulation).
THREE PARADIGMS OF RL

- **Model-based**: estimate dynamics \( \hat{P} \) and then solve the resulting optimal control problem (planning and simulation).

- **Approximate Dynamic Programming**: Exploit recurrence structure in optimal policy (Q-learning):
  - Estimation: Given a policy \( \pi \), compute the Value of a state \( s \):
    \[
    V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); s_0 = s, a_k = \pi(s_k).
    \]
  - Temporal-Difference (TD) learning enforces \( V^\pi \) to satisfy rec.
  - Control: Modify \( \pi \) greedily from estimated value functions.
THREE PARADIGMS OF RL

- **Model-based**: estimate dynamics $\hat{P}$ and then solve the resulting optimal control problem (planning and simulation).

- **Approximate Dynamic Programming**: Exploit recurrence structure in optimal policy (Q-learning):
  - Estimation: Given a policy $\pi$, compute the Value of a state $s$:
    \[
    V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); \ s_0 = s, \ a_k = \pi(s_k).
    \]
  - Temporal-Difference (TD) learning enforces $V^\pi$ to satisfy rec.
  - Control: Modify $\pi$ greedily from estimated value functions.

- **Policy gradient**: Bypass both model and value, optimize directly over parameters of policy. *Essentially a derivative-free method.*
State and Action Spaces can be huge ($2^{170}$ for GO) or even infinite and high-dimensional.

In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|S| \cdot |A|$.
State and Action Spaces can be huge ($2^{170}$ for GO) or even infinite and high-dimensional.

In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|S| \cdot |A|$.

Such structure can be incorporated by function approximation, i.e., appropriate parametrisations of value functions, policies, and model dynamics: $\theta \mapsto \{V^\pi_\theta(s), s \in S\}$.

In generic cases, efficient function approximation will be non-linear.

State and Action Spaces can be huge ($2^{170}$ for GO) or even infinite and high-dimensional.

In absence of structural/modeling assumptions, sample complexity will be at least linear with respect to $|S| \cdot |A|$.

Such structure can be incorporated by function approximation, ie appropriate parametrisations of value functions, policies, and model dynamics: $\theta \mapsto \{V_{\theta^\pi}(s), s \in S\}$.

In generic cases, efficient function approximation will be non-linear.


How to learn with guarantees using nonlinear approx?
THIS WORK

- Focus on Value estimation with non-linear function approximation: convergence of non-linear TD learning.

- Interplay between MDP and function approximation geometry: we establish convergence conditions.

- Key geometric properties of function approximation:
  - Homogeneity
  - “Includes” linear functions → Residual architecture.
Recall the value function associated to a current policy:

\[ V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); \quad s_0 = s, a_k = \pi(s_k). \]
Recall the value function associated to a current policy:

\[ V^\pi(s) := \mathbb{E} \sum_{k} \gamma^k R(s_k, a_k); \quad s_0 = s, \quad a_k = \pi(s_k). \]

It is the unique solution of the Bellman equation:

\[ V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} V^\pi(s'), \text{ with } \bar{R}(s) = \mathbb{E} R(s, \pi(s)). \]
Recall the value function associated to a current policy:

\[ V^\pi(s) := \mathbb{E} \sum_k \gamma^k R(s_k, a_k); \quad s_0 = s, a_k = \pi(s_k). \]

It is the unique solution of the Bellman equation:

\[ V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim P(\cdot|s)} V^\pi(s') , \text{ with } \bar{R}(s) = \mathbb{E} R(s, \pi(s)). \]

The most popular algorithm to estimate it is Temporal-Difference learning [Sutton, Samuel].

Given transition \((s, \bar{R}(s), s')\) and step-size \(\alpha_k\)

\[ V^{(k+1)}(s) = V^{(k)}(s) + \alpha_k \left( R(s, a) + \gamma V^{(k)}(s') - V^{(k)}(s) \right). \]
Recall the value function associated to a current policy:

\[ V^\pi(s) := \mathbb{E} \sum_{k} \gamma^k R(s_k, a_k); \ s_0 = s, a_k = \pi(s_k). \]

It is the unique solution of the Bellman equation:

\[ V^\pi(s) = \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} V^\pi(s') , \text{ with } \bar{R}(s) = \mathbb{E} R(s, \pi(s)). \]

The most popular algorithm to estimate it is Temporal-Difference learning [Sutton, Samuel].

- Given transition \((s, \bar{R}(s), s')\) and step-size \(\alpha_k\)

\[ V^{(k+1)}(s) = V^{(k)}(s) + \alpha_k \left( R(s, a) + \gamma V^{(k)}(s') - V^{(k)}(s) \right) . \]

- Under appropriate conditions, we have \(V^{(k)} \to V\) as \(k \to \infty\).

[Robbins & Munro, 50s]
This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.
This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.

Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution $\mu$. 
This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.

Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution $\mu$.

As $\alpha_k \rightarrow 0$, the expected dynamics of TD become

$$\dot{V}(s) = \mu(s) \left( \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s)} [V(s')] - V(s) \right)$$
This algorithm can be seen as taking a stochastic gradient step with respect to the expected squared Bellman error.

Continuous-time interpretation: suppose $\mathcal{P}$ defines an aperiodic, irreducible Markov chain, with stationary distribution $\mu$.

as $\alpha_k \to 0$, the expected dynamics of TD become

$$\dot{V}(s) = \mu(s) \left( \bar{R}(s) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot | s)}[V(s')] - V(s) \right)$$

In matrix form, using Bellman equation $V^\pi = \bar{R} + \gamma \mathcal{P} V^\pi$,

$$\dot{V} = D_\mu (\bar{R} + \gamma \mathcal{P} V - V) = -A(V - V^\pi), \text{ with}$$

$$A := D_\mu (I - \gamma \mathcal{P}), \; D_\mu = \text{diag}(\mu).$$
Fact: $A$ is a “positive-definite”, non-symmetric, matrix, ie

$$x^\top Ax > 0 \text{ when } \|x\| > 0.$$  

[Sutton, ’88]
CONSISTENCY OF TD-LEARNING: TABULAR CASE

- Fact: $A$ is a “positive-definite”, non-symmetric, matrix, ie $x^\top Ax > 0$ when $\|x\| > 0$. [Sutton,’88]

- Consequence: $V(t)$ converges (linearly) to $V^\pi$ as $t \to \infty$. 
Fact: $A$ is a “positive-definite”, non-symmetric, matrix, i.e. $x^\top Ax > 0$ when $\|x\| > 0$. [Sutton,’88]

Consequence: $V(t)$ converges (linearly) to $V^\pi$ as $t \to \infty$.

However, this algorithm currently computes an independent quantity for each $s \in S$ (the “tabular” case).
Fact: $A$ is a “positive-definite”, non-symmetric, matrix, i.e.
\[ x^\top Ax > 0 \text{ when } \|x\| > 0. \]  
[Sutton,'88]

Consequence: $V(t)$ converges (linearly) to $V^\pi$ as $t \to \infty$.

However, this algorithm currently computes an independent quantity for each $s \in S$ (the “tabular” case).

Infeasible in any typical large-scale scenario.
To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^{d}$ and $\theta \mapsto V_\theta \in \mathbb{R}^{\vert S \vert}$ differentiable.
To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_{\theta} \in \mathbb{R}^{|S|}$ differentiable.

**TD(0) “semi-gradient” algorithm [Sutton]:**

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_{\theta} V_{\theta^{(k)}}(s) \left( \bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s) \right).$$
To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{|S|}$ differentiable.

**TD(0) “semi-gradient” algorithm:**

$$
\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_\theta V_{\theta^{(k)}}(s) \left( \bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s) \right).
$$

Such update approximates the stochastic gradient of the squared Bellman error $\Delta(\theta) := \|V_\theta - \bar{R} - \gamma \mathcal{P} V_\theta\|^2$.
FUNCTION APPROXIMATION

- To overcome such blowup, one considers function approximation. Let $\theta \in \mathbb{R}^d$ and $\theta \mapsto V_\theta \in \mathbb{R}^{1^{\left|S\right|}}$ differentiable.

- TD(0) “semi-gradient” algorithm:

$$
\theta^{(k+1)} = \theta^{(k)} + \alpha_k \nabla_\theta V_{\theta^{(k)}}(s) \left( \bar{R}(s) + \gamma V_{\theta^{(k)}}(s') - V_{\theta^{(k)}}(s) \right).
$$

- Such update approximates the stochastic gradient of the squared Bellman error $\Delta(\theta) := \|V_\theta - \bar{R} - \gamma \mathcal{P}V_\theta\|^2$

  - Problem: an unbiased estimator of $\nabla_\theta \Delta(\theta)$ requires two samples $s'$ from the environment ("double-sample" problem):

$$
\nabla_\theta \Delta(\theta) := 2(V_\theta - \bar{R} - \gamma \mathcal{P}V_\theta) \cdot (\nabla_\theta V_\theta - \gamma \mathcal{P}\nabla_\theta V_\theta)
$$

- This breaks convergence guarantees of stochastic optimization.
In continuous time, the corresponding ODE becomes

\[
\dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi)
\]
In continuous time, the corresponding ODE becomes

\[ \dot{\theta} = -\nabla V(\theta)^\top A(V(\theta) - V^\pi) \]

Two known regimes where this ODE converges:

- Linear function approximation [Tsitsiklis & Van Roy’97]:
  \[
  V(\theta) = \Phi \theta \quad \rightarrow \quad \dot{\theta} = -\Phi^\top A\Phi(\theta - \theta^*) , \quad \theta^* = (\Phi^\top A\Phi)^{-1}\Phi^\top AV^\pi.
  \]
In continuous time, the corresponding ODE becomes

\[ \dot{\theta} = -\nabla V(\theta)^\top A (V(\theta) - V^\pi) \]

Two known regimes where this ODE converges:

- Linear function approximation [Tsitsiklis & Van Roy'97]:

\[ V(\theta) = \Phi \theta \quad \rightarrow \quad \dot{\theta} = -\Phi^\top A \Phi (\theta - \theta^*) , \quad \theta^* = (\Phi^\top A \Phi)^{-1} \Phi^\top A V^\pi. \]

- Reversible Markov Chain [Ollivier,'18].

\[ A = A^\top \quad \rightarrow \quad \dot{\theta} = -\nabla \| V(\theta) - V^\pi \|_A^2 , \quad (\langle x, y \rangle_A := x^\top A y). \]
In continuous time, the corresponding ODE becomes

\[ \dot{\theta} = -\nabla V(\theta) \top A (V(\theta) - V^\pi) \]

Two known regimes where this ODE converges:

- Linear function approximation [Tsitsiklis & Van Roy’97]:
  \[ V(\theta) = \Phi \theta \quad \rightarrow \quad \dot{\theta} = -\Phi \top A \Phi (\theta - \theta^*) , \quad \theta^* = (\Phi \top A \Phi)^{-1} \Phi \top A V^\pi. \]

- Reversible Markov Chain [Ollivier,’18].
  \[ A = A^\top \quad \rightarrow \quad \dot{\theta} = -\nabla \| V(\theta) - V^\pi \|_A^2 , \quad (\langle x, y \rangle_A := x \top A y). \]

Alternative Strategies to TD to ensure convergence:

- “Two-time-scale” algorithms [Dai et al., Borkar et al, Chung et al]
CONSISTENCY IN THE GENERAL CASE?
CONSISTENCY IN THE GENERAL CASE?

- Convergence is not generic.
CONSISTENCY IN THE GENERAL CASE?

- Convergence is not generic.
- Divergence example from [Tsitsiklis & van Roy]:

![Diagram of MRP](image)
HOMOGENEOUS NON-LINEAR MODELS

- $f : \mathbb{R}^k \rightarrow \mathbb{R}^m$ is $h$-homogenous for $h \in \mathbb{R}$ if

  $$\forall x, \forall \alpha > 0, \ f(\alpha x) = \alpha^h f(x)$$

- If $\sigma$ is a homogeneous activation function, then neural networks using $\sigma$ are also homogeneous (wrt parameters).
**HOMOGENEOUS NON-LINEAR MODELS**

- $f : \mathbb{R}^k \to \mathbb{R}^m$ is $h$-homogenous for $h \in \mathbb{R}$ if
  \[
  \forall x, \forall \alpha > 0, \quad f(\alpha x) = \alpha^h f(x)
  \]

- If $\sigma$ is a homogeneous activation function, then neural networks using $\sigma$ are also homogeneous (wrt parameters).

- Homogeneous function approximation prevents divergence:

  **Theorem [BB’19]:** Let $\theta \mapsto V(\theta)$ be $h$-homogeneous and $l$-Holder. Then for each $\epsilon > 0$ and any initial $\theta_0$, we have
  \[
  \liminf_{t \to \infty} \|V(\theta_t)\|_\mu \leq \frac{\|\bar{R}\|_\mu}{1-\gamma} + \epsilon.
  \]
HOMOGENEOUS NON-LINEAR MODELS

- $f : \mathbb{R}^k \rightarrow \mathbb{R}^m$ is $h$-homogenenous for $h \in \mathbb{R}$ if
  \[ \forall x, \forall \alpha > 0, f(\alpha x) = \alpha^h f(x) \]

- If $\sigma$ is a homogeneous activation function, then neural networks using $\sigma$ are also homogeneous (wrt parameters).

- Homogeneous function approximation prevents divergence:

  **Theorem [BB’19]:** Let $\theta \mapsto V(\theta)$ be $h$-homogeneous and $l$-Holder. Then for each $\epsilon > 0$ and any initial $\theta_0$, we have

  \[ \liminf_{t \to \infty} \|V(\theta_t)\|_{\mu} \leq \frac{\|\bar{R}\|_{\mu}}{1 - \gamma} + \epsilon. \]

- In the worst case, homogeneous TD is not worse than using the 0 function baseline:

  \[ \|0 - V^*\|_{\mu} \simeq \frac{\|\bar{R}\|_{\mu}}{1 - \gamma} \]

- Stronger baseline?
HOMOGENEOUS RESIDUAL MODELS

- \( f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \rightarrow \mathbb{R}^m \) is residual-homogeneous if

\[
f(x_1, x_2) = \Phi x_1 + g(x_2), \quad \text{with} \ g \ \text{homogeneous.}
\]
HOMOGENEOUS RESIDUAL MODELS

- \( f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \to \mathbb{R}^m \) is residual-homogeneous if 
  \[ f(x_1, x_2) = \Phi x_1 + g(x_2), \text{ with } g \text{ homogeneous}. \]

- With residual-homogeneous, we are provably not worse than using linear models:

**Theorem [BB’19]:** Let \((\theta_1, \theta_2) \mapsto V(\theta_1, \theta_2)\) be residual-homogeneous and \(l\)-Holder. Then for each \(\epsilon > 0\) and any initial \(\theta_0\), we have 
  \[
  \liminf_{t \to \infty} \| V(\theta_t) - V^\pi \|_\mu \leq \frac{2\| V^\pi - \Pi_\Phi V^\pi \|_\mu}{1 - \gamma} + \epsilon.
  \]

- Similar guarantee as in non-convex optimization using Resnets [Shamir’18].
**HOMOGENEOUS RESIDUAL MODELS**

- $f : \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \to \mathbb{R}^m$ is **residual-homogeneous** if $f(x_1, x_2) = \Phi x_1 + g(x_2)$, with $g$ homogeneous.

- With residual-homogeneous, we are provably not worse than using linear models:

**Theorem [BB'19]:** Let $(\theta_1, \theta_2) \mapsto V(\theta_1, \theta_2)$ be residual-homogeneous and $l$-Holder. Then for each $\epsilon > 0$ and any initial $\theta_0$, we have

$$\liminf_{t \to \infty} \| V(\theta_t) - V^\pi \|_\mu \leq \frac{2\| V^\pi - \Pi_\Phi V^\pi \|_\mu}{1 - \gamma} + \epsilon.$$

- Similar guarantee as in non-convex optimization using Resnets [Shamir'18].

- Generically, no global convergence. Role of overparametrisation?
Recall that reversible dynamics result in gradient descent [Ollivier’18]. Can we leverage this property?
Recall that reversible dynamics result in gradient descent [Ollivier’18]. Can we leverage this property?

**Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is

$$\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{||S_A v||^2 + ||A v||^2}{||R_A v||^2}, \text{ with }$$

$$A = D_\mu(I - \gamma \mathcal{P}), S_A = (A + A^\top)/2, R_A = (A - A^\top)/2.$$
Recall that reversible dynamics result in gradient descent [Ollivier’18]. Can we leverage this property?

**Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is

$$
\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|SAv\|^2 + \|Av\|^2}{\|RAv\|^2},
$$

with

$$
A = D_\mu(I - \gamma \mathcal{P}), \quad SA = (A + A^\top)/2, \quad RA = (A - A^\top)/2.
$$

**Global convergence with well-conditioned function approximation:**

**Theorem [BB’19]:** Assume that $\kappa(\nabla V(\theta)\nabla V(\theta)^\top) < \rho(\mathcal{P})$ for all $\theta$. Then $V(\theta(t)) \to V^\pi$ as $t \to \infty$. 
Recall that reversible dynamics result in gradient descent [Ollivier’18]. Can we leverage this property?

**Definition:** The reversibility coefficient of a Markov Chain $\mathcal{P}$ is

$$\rho(\mathcal{P}) = \inf_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|S_A v\|^2 + \|A v\|^2}{\|R_A v\|^2},$$

with

$$A = D_\mu(I - \gamma \mathcal{P}), S_A = (A + A^\top)/2, R_A = (A - A^\top)/2.$$

Global convergence with well-conditioned function approximation:

**Theorem [BB’19]:** Assume that $\kappa(\nabla V(\theta)\nabla V(\theta)^\top) < \rho(\mathcal{P})$ for all $\theta$. Then $V(\theta(t)) \to V^\pi$ as $t \to \infty$.

- Observe that $\kappa(\nabla V(\theta)\nabla V(\theta)^\top) < \infty$ requires $d > |\mathcal{S}|$.
- Open: underparametrised case with extra smoothness?
Convergence of Value Estimation is the weakest possible guarantee.

Rate of convergence currently only known for linear models [Bandhari et al.'18]. Non-linear case?
Convergence of Value Estimation is the weakest possible guarantee.

Rate of convergence currently only known for linear models [Bandhari et al.’18]. Non-linear case?

Our current analysis does not measure sample efficiency.
Convergence of Value Estimation is the weakest possible guarantee.

Rate of convergence currently only known for linear models [Bandhari et al.‘18]. Non-linear case?

Preliminary analysis: does not measure sample efficiency.

Sample Complexity of “model-free” RL

- Tabular Case [Jin et al.‘18], [Azar et al.‘17], [Brunskill et al.].
- Linear Function Approximation [Jin et al.‘19], [Brandfonbrener et al.‘19]
- Policy Gradients [Argawal et al.‘19]
CONCLUSIONS

- Analysis of TD-learning using nonlinear function approximation.
- Interplay between geometry of function approximation (homogeneity, conditioning, linear baseline) and environment (reversibility).
- Learning with guarantees under such conditions.
- Next: Further exploit regularity of reward/environment to reduce overparametrisation.
- Next: From of value estimation to policy update, tighter link between environment and network parametrisation.
Thanks!

Reference: