

# Towards Characterizing Divergence in Deep Q-Learning

arXiv 2019

Citation: 40

OpenAI, UCB

*Joshua Achiam, Ethan Knight, Pieter Abbeel*

# Motivation

The **Deadly Triad** of DQN:

Once we put "**bootstrapping**", "**off-policy learning**", "**function approximation**" together, they will lead to **divergence** in DQN.

However, **the conditions under which divergence occurs are not well-understood.**

# Main Ideas

Why dose DQN diverge under deadly triad?

How about analyzing DQN with NTK?

# The Result of Analysis

- The main reason why DQN diverge is **Over-generalization** and **improper(too large or too small) learning rate**.
- The **network architecture** seems to affect the convergence of DQN

# Outline

- Motivation
- Main Ideas
- The Result of Analysis
- Analysis Setup
- NTK of DQN
- Building Intuition for Divergence with NTK
- PreQN
- Experiments

# Analysis Setup

## Q-Function

The optimal Q-function  $Q^*$ , which is known to satisfy the optimal Bellman equation:

$$Q^*(s, a) = E_{s' \sim P}[R(s, a, s') + \gamma \max_{a'} Q^*(s', a')]$$

The value iteration of Q-learning is

$$Q_{k+1}(s, a) = E_{s, a \sim P}[Q_k(s, a) + \alpha_k(r + \gamma \max_{a'} Q_k(s', a') - Q_k(s, a))]$$

# Bellman Operator

Define an optimal Bellman operator  $\tau^* : Q \rightarrow Q$  be the operator on Q-functions

$$Q^* = \tau^* Q^*$$

The operator  $\tau^*$  is a contraction map

Thus, the value iteration of Q-learning can be represented as

$$Q_{k+1}(s, a) = E_{s,a \sim P}[Q_k(s, a) + \alpha_k(\hat{\tau}^* Q_k(s, a) - Q_k(s, a))]$$

$$\hat{\tau}^* Q_k(s, a) = r + \gamma \max_{a'} Q_k(s', a')$$

The optimal policy  $\pi^*$  can be obtained with  $\pi^*(s) = \arg \max_a Q^*(s, a)$  after the value iteration  $Q_{k+1} = \tau^* Q_k$  converges



# NTK of DQN

The Bellman equation of DQN with the **experience distribution**  $\rho$  in replay buffer

$$Q_{k+1}(s, a) = E_{s,a \sim \rho} [Q_k(s, a) + \alpha_k (\hat{\tau}^* Q_k(s, a) - Q_k(s, a))]$$

$$\hat{\tau}^* Q_k(s, a) = r + \gamma \max_{a'} Q_k(s', a')$$

The TD error  $\delta_t$  with minibatch sampled from replay buffer  $\rho$

$$\begin{aligned} \delta_t &= E_{s,a \sim \rho} [\tau^* Q(s_t, a_t) - Q(s_t, a_t)] \\ &= E_{s,a \sim \rho} [r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \end{aligned}$$

Update the weights

$$\begin{aligned} \theta' &= \theta + \alpha \nabla_{\theta} \delta_t \\ &= \theta + \alpha E_{s,a \sim \rho} [(\tau^* Q_{\theta}(s, a) - Q_{\theta}(s, a)) \nabla_{\theta} Q_{\theta}(s, a)] \end{aligned} \quad (5)$$

# NTK of DQN

The Taylor Expansion of  $Q$  around  $\theta$  at a state-action pair  $(\bar{s}, \bar{a})$ .

$$Q_{\theta'}(\bar{s}, \bar{a}) = Q_{\theta}(\bar{s}, \bar{a}) + \nabla_{\theta} Q_{\theta}(\bar{s}, \bar{a})^{\top} (\theta' - \theta)$$

Combine with Eq. 5

$$\theta' - \theta = \alpha E_{s,a \sim \rho} [(\tau^* Q_{\theta}(s, a) - Q_{\theta}(s, a)) \nabla_{\theta} Q_{\theta}(s, a)]$$

Thus, the Q-values before and after an update are related by:

$$Q_{\theta'}(\bar{s}, \bar{a}) = Q_{\theta}(\bar{s}, \bar{a}) + \alpha E_{s,a \sim \rho} [k_{\theta}(\bar{s}, \bar{a}, s, a) (\tau^* Q_{\theta}(s, a) - Q_{\theta}(s, a))]$$

$$k_{\theta}(\bar{s}, \bar{a}, s, a) = \nabla_{\theta} Q_{\theta}(\bar{s}, \bar{a})^{\top} \nabla_{\theta} Q_{\theta}(s, a) \quad (9)$$

Where  $k_{\theta}(\bar{s}, \bar{a}, s, a)$  is NTK

# Building Intuition for Divergence with NTK

## Theorem 1

The Q function is represented as a vector in  $\mathbb{R}^{|S||A|}$ , and the Q-values before and after an update are related by:

$$Q_{\theta'} = Q_{\theta} + \alpha K_{\theta} D_{\rho} (\tau^* Q_{\theta} - Q_{\theta}) \quad (10)$$

where  $K_{\theta} \in \mathbb{R}^{|S||A| \times |S||A|}$  is the matrix of entries given by the NTK  $k_{\theta}(\bar{s}, \bar{a}, s, a)$ , and  $D_{\rho}$  is a matrix with entries given by  $\rho(s, a)$ , the distribution from the replay buffer.

Consider the operator  $\mathcal{U}_3$  given by

$$\mathcal{U}_3 Q = Q + \alpha K D_\rho(\tau^* Q - Q) \quad (14)$$

### Lemma 3

Under the same conditions as Theorem 1, the Q-values before and after an update are related by

$$Q_{\theta'} = \mathcal{U}_3 Q_\theta \quad (15)$$

## Theorem 2

Let indices  $i, j$  refer to state-action pairs. **Suppose** that  $K$  and  $\rho$  satisfy the conditions:

$$\forall i, \alpha K_{ii} \rho_i < 1 \quad (16)$$

$$\forall i, (1 + \gamma) \sum_{j \neq i} |K_{ij}| \rho_j \leq (1 - \gamma) K_{ii} \rho_i \quad (17)$$

Then  $\mathcal{U}_3$  is a contraction on  $Q$  in the sup norm, with fixedpoint  $Q^*$ .

## Proof of Theorem 2

$$\begin{aligned}
[\mathcal{U}_3 Q_1 - \mathcal{U}_3 Q_2]_i &= [(Q_1 + \alpha K D_\rho(\tau^* Q_1 - Q_1)) - (Q_2 + \alpha K D_\rho(\tau^* Q_2 - Q_2))]_i \\
&= [(Q_1 - Q_2) + \alpha K D_\rho((\tau^* Q_1 - Q_1) - (\tau^* Q_2 - Q_2))]_i \\
&= \sum_j \delta_{ij} [Q_1 - Q_2]_j + \alpha \sum_j K_{ij} \rho_j [(\tau^* Q_1 - Q_1) - (\tau^* Q_2 - Q_2)]_j \\
&= \sum_j (\delta_{ij} - \alpha K_{ij} \rho_j) [Q_1 - Q_2]_j + \alpha \sum_j K_{ij} \rho_j [\tau^* Q_1 - \tau^* Q_2]_j \\
&\leq \sum_j (|\delta_{ij} - \alpha K_{ij} \rho_j| + \alpha \gamma |K_{ij}| \rho_j) \|Q_1 - Q_2\|_\infty
\end{aligned}$$

Thus we can obtain a modulus as  $\beta(K) = \max_i \sum_j (|\delta_{ij} - \alpha K_{ij} \rho_j| + \alpha \gamma |K_{ij}| \rho_j)$

We'll break it up into on-diagonal and off-diagonal parts, and assume that  $\alpha K_{ii} \rho_i \leq 1$  :

$$\begin{aligned}
\beta(K) &= \max_i \sum_j (|\delta_{ij} - \alpha K_{ij} \rho_j| + \alpha \gamma |K_{ij}| \rho_j) \\
&= \max_i ((|1 - \alpha K_{ii} \rho_i| + \alpha \gamma K_{ii} \rho_i) + (1 + \gamma) \alpha \sum_{j \neq i} |K_{ij}| \rho_j) \\
&= \max_i ((1 - \alpha K_{ii} \rho_i + \alpha \gamma K_{ii} \rho_i) + (1 + \gamma) \alpha \sum_{j \neq i} |K_{ij}| \rho_j) \\
&= \max_i (1 - (1 - \gamma) \alpha K_{ii} \rho_i + (1 + \gamma) \alpha \sum_{j \neq i} |K_{ij}| \rho_j)
\end{aligned}$$

According to Banach Fixed-Point Theorem, if  $\beta(K) < 1$ ,  $[\mathcal{U}_3 Q_1 - \mathcal{U}_3 Q_2]_i$  would converge

Thus,

$$\forall i, \beta(K) < 1$$

$$\forall i, \max_i (1 - (1 - \gamma)\alpha K_{ii}\rho_i + (1 + \gamma)\alpha \sum_{j \neq i} |K_{ij}|\rho_j) < 1$$

$$\forall i, 1 - (1 - \gamma)\alpha K_{ii}\rho_i + (1 + \gamma)\alpha \sum_{j \neq i} |K_{ij}|\rho_j < 1$$

$$\forall i, (1 + \gamma) \sum_{j \neq i} |K_{ij}|\rho_j < (1 - \gamma)K_{ii}\rho_i$$

$$\forall i, \frac{(1 + \gamma)}{(1 - \gamma)} \sum_{j \neq i} |K_{ij}|\rho_j < K_{ii}\rho_i$$

Note that this is a quite restrictive condition, since for  $\gamma$  high (EX: 0.99),  $(1 + \gamma)/(1 - \gamma)$  will be quite large, and the left hand side has a sum over all off-diagonal terms in a row.



## Intuitions

- The stability and convergence of Q-learning is tied to the generalization properties of DQN.
- DQNs with more aggressive generalization (larger off-diagonal terms in  $K_\theta$ ) are less likely to demonstrate stable learning.
- The architecture of network will affect to the stability and convergence of Q-learning.

## Preconditioned Q-Networks(PreQN)

Denote  $\Phi_{\theta}^T \in \mathbb{R}^{d|S||A|}$  as the matrix whose columns are  $\nabla_{\theta} Q_{\theta}(s, a)$ . With Taylor expansion, we have

$$Q_{\theta'} \approx Q_{\theta} + \Phi_{\theta}^T (\theta' - \theta) \quad (19)$$

However, to stabilize the update, since we've known that the Q-learning is stable and satisfy the Banach's fix point theorem, we want to make the update of DQL close to the update of Q-learning. That is, satisfy the following relation

$$Q_{\theta'} \approx Q_{\theta} + \alpha(\tau^* Q_{\theta} - Q_{\theta}) \quad (20)$$

We can simply reorganize the equation

$$Q_{\theta'} - Q_{\theta} \approx \Phi_{\theta}^T (\theta' - \theta) \quad (19-1)$$

$$Q_{\theta'} - Q_{\theta} \approx \alpha(\tau^* Q_{\theta} - Q_{\theta}) \quad (20-1)$$

Combine Eq. (19-1) and Eq. (20-1)

$$\Phi_{\theta}^T (\theta' - \theta) \approx \alpha(\tau^* Q_{\theta} - Q_{\theta})$$

$$(\Phi_{\theta}^T \Phi_{\theta}) \Phi_{\theta}^{-1} (\theta' - \theta) \approx \alpha(\tau^* Q_{\theta} - Q_{\theta})$$

$$K_{\theta} \Phi_{\theta}^{-1} (\theta' - \theta) \approx \alpha(\tau^* Q_{\theta} - Q_{\theta})$$

We get

$$\theta' \approx \theta + \alpha \Phi_{\theta} K_{\theta}^{-1} (\tau^* Q_{\theta} - Q_{\theta}) \quad (21)$$

$K_{\theta}^{-1}$  is the preconditioner to calibrate the ill condition of the update.

We form  $K_\theta$  for the minibatch, find the least-squares solution  $Z$  to

$$K_\theta Z = \tau^* Q_\theta - Q_\theta$$

Thus, the calibrated update is

$$Z = K_\theta^{-1} (\tau^* Q_\theta - Q_\theta)$$

---

**Algorithm 1** PreQN (in style of DDPG)

---

```
1: Given: initial parameters  $\theta, \phi$  for  $Q, \mu$ , empty replay buffer  $\mathcal{D}$ 
2: Receive observation  $s_0$  from environment
3: for  $t = 0, 1, 2, \dots$  do
4:   Select action  $a_t = \mu_\phi(s_t) + \mathcal{N}_t$ 
5:   Step environment to get  $s_{t+1}, r_t$  and terminal signal  $d_t$ 
6:   Store  $(s_t, a_t, r_t, s_{t+1}, d_t) \rightarrow \mathcal{D}$ 
7:   if it's time to update then
8:     for however many updates do
9:       Sample minibatch  $B = \{(s_i, a_i, r_i, s'_i, d_i)\}$  from  $\mathcal{D}$ 
10:      For each transition in  $B$ , compute TD errors:
```

$$\Delta_i = r_i + \gamma(1 - d_i)Q_\theta(s'_i, \mu_\phi(s'_i)) - Q_\theta(s_i, a_i)$$

```
11:      Compute minibatch  $K_\theta$  matrix and find least-squares
      solution  $Z$  to  $K_\theta Z = \Delta$ 
12:      Compute proposed update for  $Q$  with:
```

$$\theta' = \theta + \alpha_q \sum_{(s,a) \in B} Z(s, a) \nabla_\theta Q_\theta(s, a)$$

```
13:      Exponentially decay  $\theta'$  towards  $\theta$  until
```

$$\cos(Q_{\theta'} - Q_\theta, \mathcal{T}^* Q_\theta - Q_\theta) \geq \eta,$$

```
      then set  $\theta \leftarrow \theta'$ .
```

```
14:      Update  $\mu$  with:
```

$$\phi \leftarrow \phi + \alpha_\mu \frac{1}{|B|} \sum_{s \in B} \nabla_\phi Q_\theta(s, \mu_\phi(s))$$

```
15:      end for
```

```
16:    end if
```

```
17:  end for
```

---

# Algorithm

For the minibatch, compute the update

$$\theta' = \theta + \alpha \sum_{(s,a) \in B} Z(s, a) \nabla_\theta Q_\theta(s, a)$$

To make the updated Q-function close to the target Bellman optimality, we use linesearch to achieve the criterion

$$\cos(Q_{\theta'} - Q_\theta, \tau^* Q_\theta - Q_\theta) \geq \eta$$

# Experiments

## Metrics

We consider the ratio of the average off-diagonal row entry to the on-diagonal entry,  $R_i$  as "row ratio":

$$R_i(K) = \frac{1}{N} \frac{\sum_{j \neq i} |K_{ij}|}{K_{ii}}$$

where  $N$  is the size of the square matrix  $K$ .

The larger off-diagonal entries, the higher row ratio.

The higher row ratio, the greater generalization, but less stability and convergence

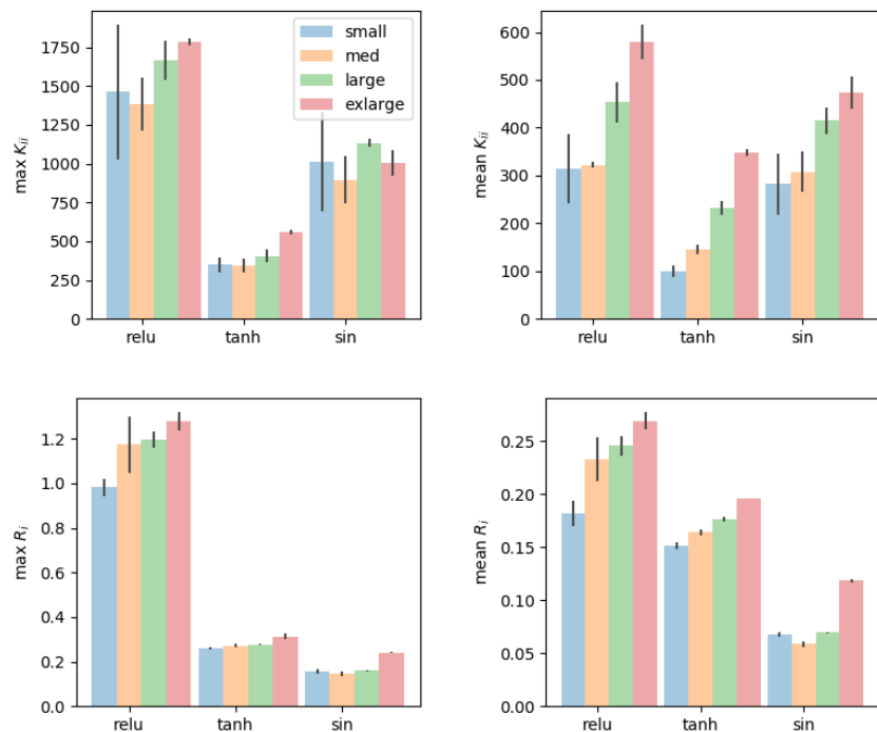


Figure 4. NTK analysis for randomly-initialized networks with various activation functions, where the NTKs were formed using 1000 steps taken by a rails-random policy in the **Ant-v2** gym environment (with the same data used across all trials). Networks are MLPs with widths of 32, 64, 128, 256 hidden units (small, med, large, exlarge respectively) and 2 hidden layers. Each bar is the average over 3 random trials (different network initializations).

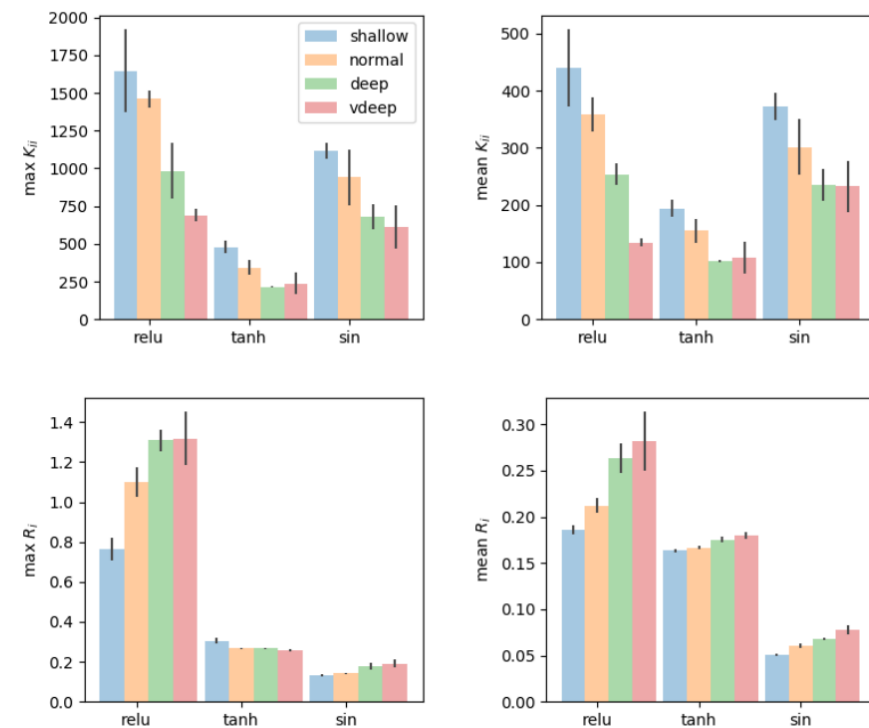
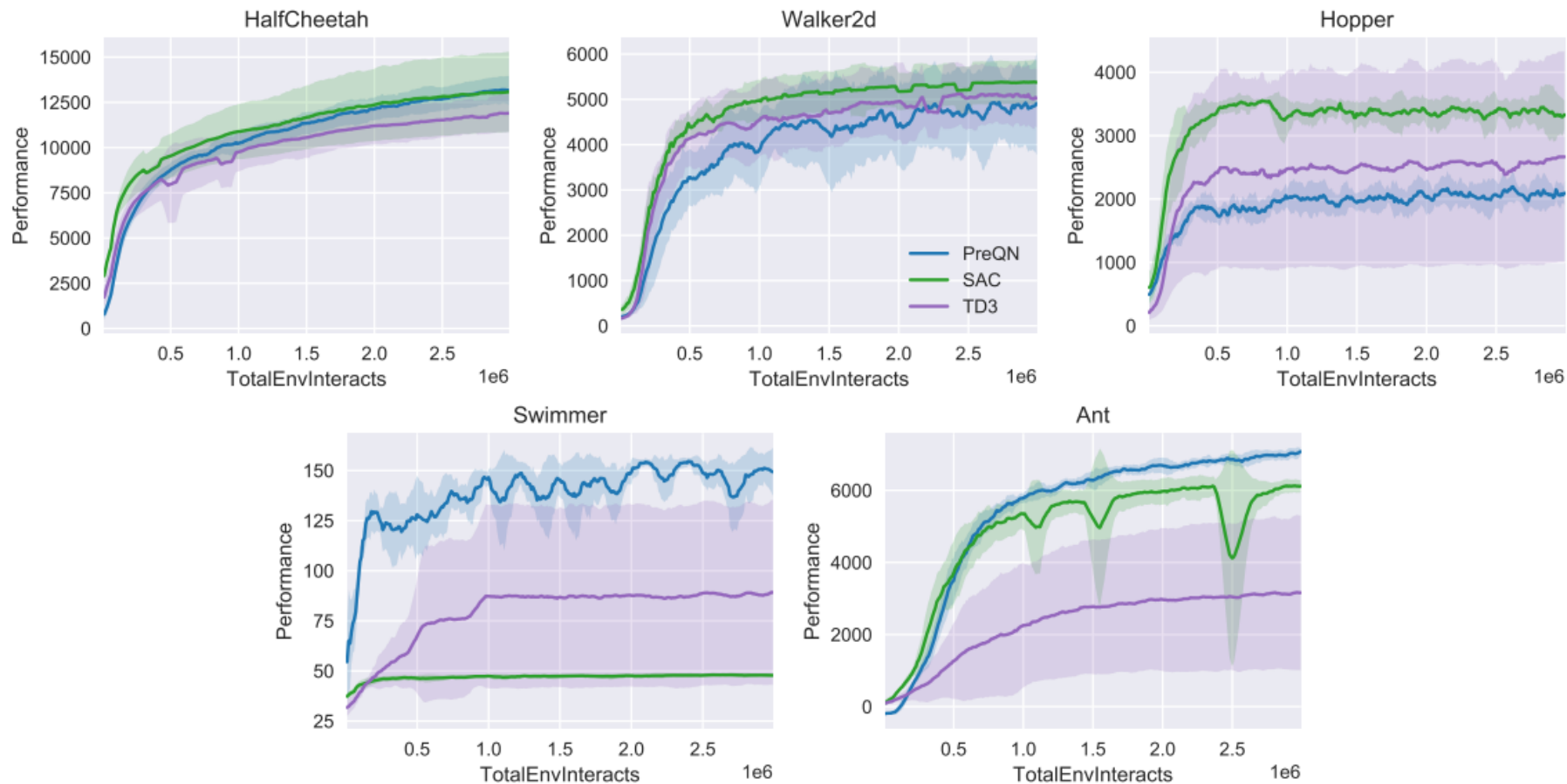


Figure 7. NTK analysis for randomly-initialized networks with various activation functions, where the NTKs were formed using 1000 steps taken by a rails-random policy in the **Ant-v2** gym environment (with the same data used across all trials). Networks are MLPs with depths of 1, 2, 3, 4 hidden layers (shallow, normal, deep, vdeep respectively) and 64 units per layer. Each bar is the average over 3 random trials (different network initializations).

- Relu nets commonly have the **largest on-diagonal elements and row ratio** (so they should learn quickly and generalize aggressively)
- Sin networks have **low off-diagonal elements and lowest row ratio**.
- Diagonal elements tend to increase with width and decrease with depth, across activation functions.
- Row ratios tend to increase with depth across activation functions, and do not clearly correlate with width.





*Figure 2. Benchmarking PreQN against TD3 and SAC on standard OpenAI Gym MuJoCo environments. Curves are averaged over 7 random seeds. PreQN is stable and performant, despite not using target networks. The PreQN experiments used sin activations; the TD3 and SAC experiments used relu activations.*

In some of experiments, PreQN outperform than TD3 and SAC. PreQN is more stable trivially.

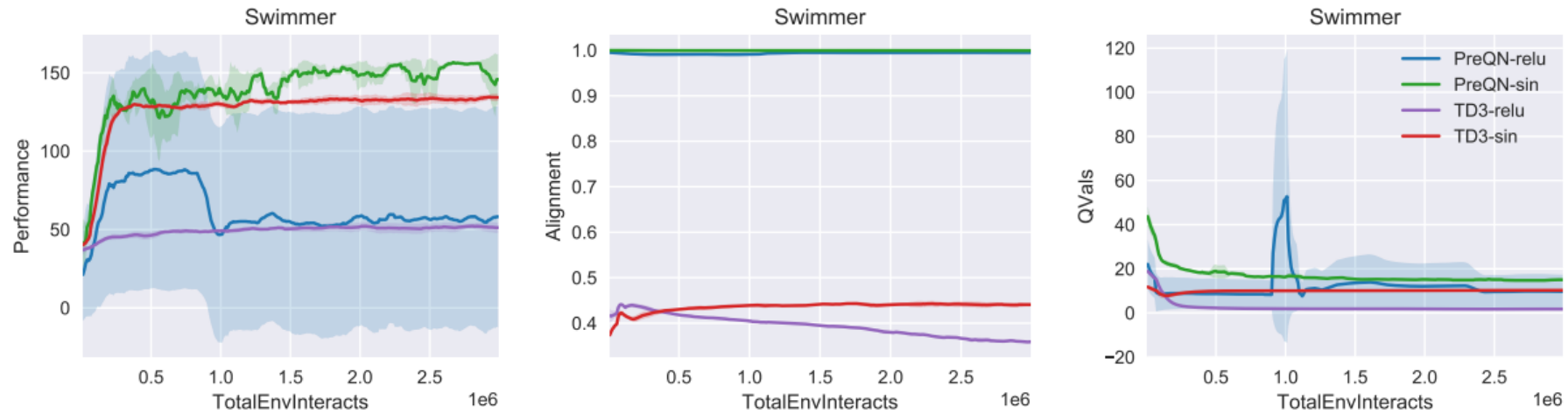


Figure 13. Comparison between PreQN and TD3 for relu and sin activation functions in the Swimmer-v2 gym environment. Results averaged over 3 random seeds.

PreQN seems great.

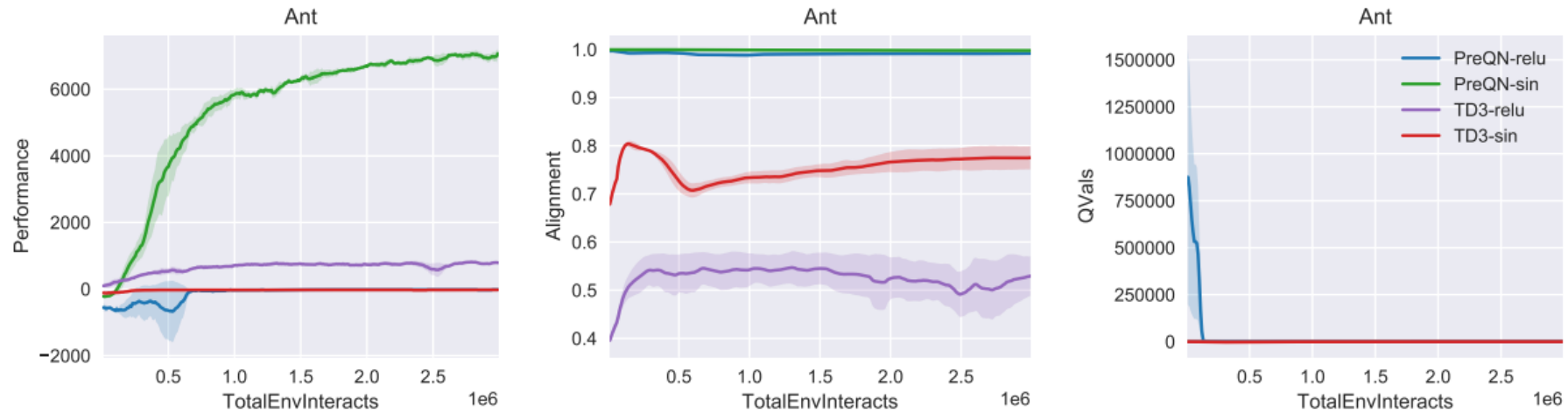


Figure 14. Comparison between PreQN and TD3 for relu and sin activation functions in the Ant-v2 gym environment. Results averaged over 3 random seeds.

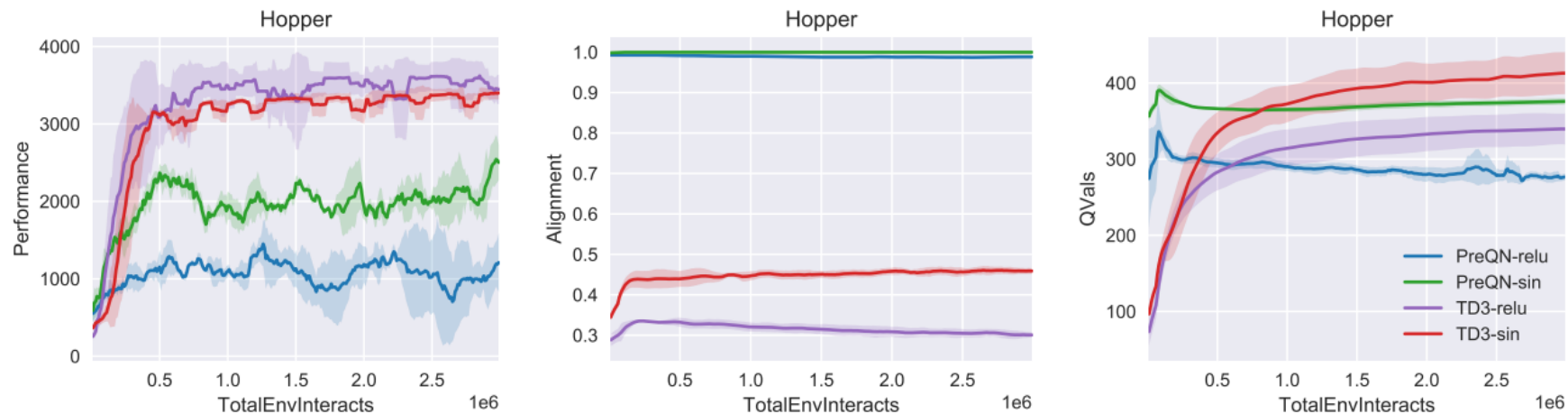


Figure 11. Comparison between PreQN and TD3 for relu and sin activation functions in the Hopper-v2 gym environment. Results averaged over 3 random seeds.

But some of experiments are not.

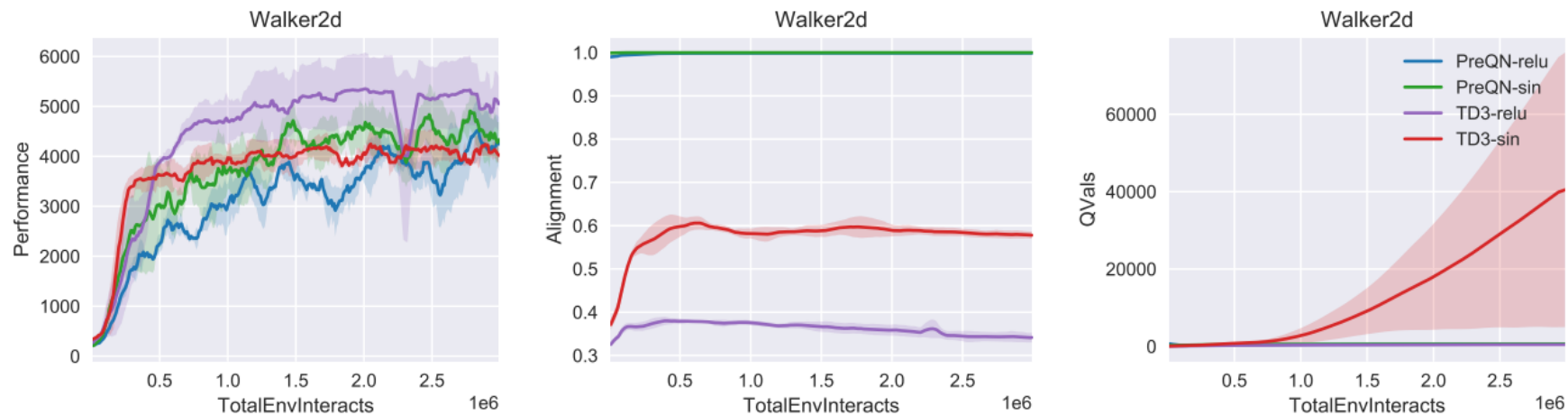


Figure 12. Comparison between PreQN and TD3 for relu and sin activation functions in the Walker2d-v2 gym environment. Results averaged over 3 random seeds.

# Appendix

## Theorem 3

Consider a **sequence of updates**  $\{\mathcal{U}_0, \mathcal{U}_1, \dots\}$ , with each  $\mathcal{U}_i : Q \rightarrow Q$  Lipschitz continuous, with Lipschitz constant  $\beta_i$ , with respect to a norm  $\|\cdot\|$ . Furthermore, **suppose all  $\mathcal{U}_i$  share a common fixed-point,  $\tilde{Q}$** . Then for any initial point  $Q_0$ , the sequence of iterates produced by  $Q_{i+1} = \mathcal{U}_i Q_i$  satisfies:

$$\|\tilde{Q} - Q_i\| \leq \left( \prod_{k=0}^{i-1} \beta_k \right) \|\tilde{Q} - Q_0\|$$

Furthermore, if there is an iterate  $j$  such that  $\forall k \leq j, \beta_k \in [0, 1)$ , the sequence  $\{\mathcal{U}_0, \mathcal{U}_1, \dots\}$  converges to  $\tilde{Q}$ .

Roughly speaking, this theorem says that **if you sequentially apply different contraction maps with the same fixed-point, you will attain that fixed-point which is also optimal point  $Q^*$  in DQL.**

## Contraction Map

Let  $X$  be a vector space with norm  $\| \cdot \|$ , and  $f$  a function from  $X$  to  $X$ . If  $\forall x, y \in X$ ,  $f$  satisfies

$$\|f(x) - f(y)\| \leq \beta \|x - y\|$$

with  $\beta \in [0, 1)$ , then  $f$  is called a contraction map with modulus  $\beta$

# Banach Fixed-Point Theorem

Let  $f$  be a contraction map,  $\exists x_u$  st  $f(x_u) = x_u$ .

## Properties

- $x_u$  is an unique fixed-point.
- Because  $f$  is a contraction map,  $x_u$  can be obtained by the repeated application of  $f$ : for any point  $x_0 \in X$ , if we define a sequence of points  $\{x_n\}$  such that  $x_n = f(x_{n-1})$ ,  $\lim_{n \rightarrow \infty} x_n = x$ .

# Intuitions

- Q-values for missing (or under-represented) state-action pairs are adjusted by **generalization with errors**. **Bootstrapping then propagates those errors** through the Q-values for all other state-action pairs.



# Reference

- [Washington University - Line Search Methods](#)