# Regularized Anderson Acceleration for Off-Policy Deep Reinforcement Learning

Wenjie Shi, Shiji Song, Hui Wu, Ya-Chu Hsu, Cheng Wu, Gao Huang
Tsinghua University, Beijing, China
Email: shiwj16@mails.tsinghua.edu.cn

NEURAL INFORMATION PROCESSING SYSTEMS

清華大學 Tsinghua University

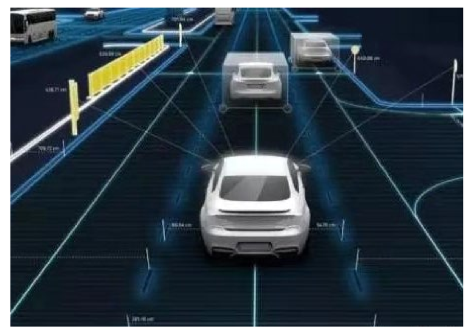
## MOTIVATION

### Sample inefficiency

- An enormous number of trials
- Long training time


AlphaGo Zero
millions of self-play

StarCraft
several days



Autonomous Driving
real physical scenarios

**Existing methods**

- To learn the model of system dynamics.
- To reuse past experience (off-policy).

**Main observations**

- RL is closely linked to fixed-point problem.
- Anderson acceleration is a method capable of speeding up the computation of fixed point iterations.

## METHOD

**Fixed-point problem**:
Given $g: \mathbb{R}^n \to \mathbb{R}^n$, solve $x = g(x)$.

**RL problem**:
Given the optimality Bellman operator $\mathcal{T}$: solve $Q^\pi = \mathcal{T}Q^\pi$.

**Algorithm FPI**. Fixed-Point Iteration:
For $k = 0,1,\ldots$
Set $x_{k+1} = g(x_k)$.

**Algorithm PI**. Policy Iteration:
For $k = 0,1,\ldots$
Set $Q^{\pi_{k+1}} = \mathcal{T}Q^{\pi_k}$.

**Challenges**

- Sweeping entire state-action space is intractable.
- Function approximation errors are unavoidable.
- The solution may suffer from ill-conditioning.

**Algorithm AA**. Anderson Acceleration:
For $k = 0,1,\ldots$
Set $F_k = (f_{k-m+1},\ldots,f_k)$, where $f_i = g(x_i) - x_i$.
Solve $\alpha^k = (\alpha_1^k,\ldots,\alpha_m^k)^T$:
$min_\alpha \ ||F_k\alpha||_2$ s.t. $\sum \alpha_i = 1$.
Set $x_{k+1} = \sum_{i=1}^m \alpha_i^k g(x_{k-m+i})$.

**Algorithm AA for PI.**
For $k = 0,1,\ldots$
Set $\Delta_k = (\delta_{k-m+1},\ldots,\delta_k)$,
where $\delta_i = \mathcal{T}Q^{\pi_i} - Q^{\pi_i}$.
Solve $\alpha^k = (\alpha_1^k,\ldots,\alpha_m^k)^T$:
$\boxed{min_\alpha \ ||\Delta_k\alpha||_2 \text{ s.t. } \sum \alpha_i = 1.}$
Set $Q^{\pi_{k+1}} = \sum_{i=1}^m \alpha_i^k \mathcal{T}Q^{\pi_{k-m+i}}$.

**Algorithm RAA**. Regularized AA:
……
$\boxed{min_\alpha \ ||\widetilde{\Delta}_k\alpha||_2^2 + \lambda||\alpha||_2^2}$
……

**Proposition 1:** *Consider two identical PIs $P_1$ and $P_2$ with function approximation. $P_2$ is implemented with RAA, whereas $P_1$ is only implemented with AA. Let $\alpha^k$ and $\tilde{\alpha}^k$ be the coefficient vectors of $P_1$ and $P_2$ respectively.*

$$\|\tilde{\alpha}^k\|_2 \le \sqrt{\frac{\lambda + \|\widetilde{\Delta}_k\|_2^2}{m\lambda}}, \quad \|\tilde{\alpha}^k - \alpha^k\|_2 \le \frac{\|\widetilde{\Delta}_k^T\widetilde{\Delta}_k - \Delta_k^T\Delta_k\|_2 + \lambda}{\lambda}\|\alpha^k\|_2$$

## EXPERIMENTS

### Comparative evaluation

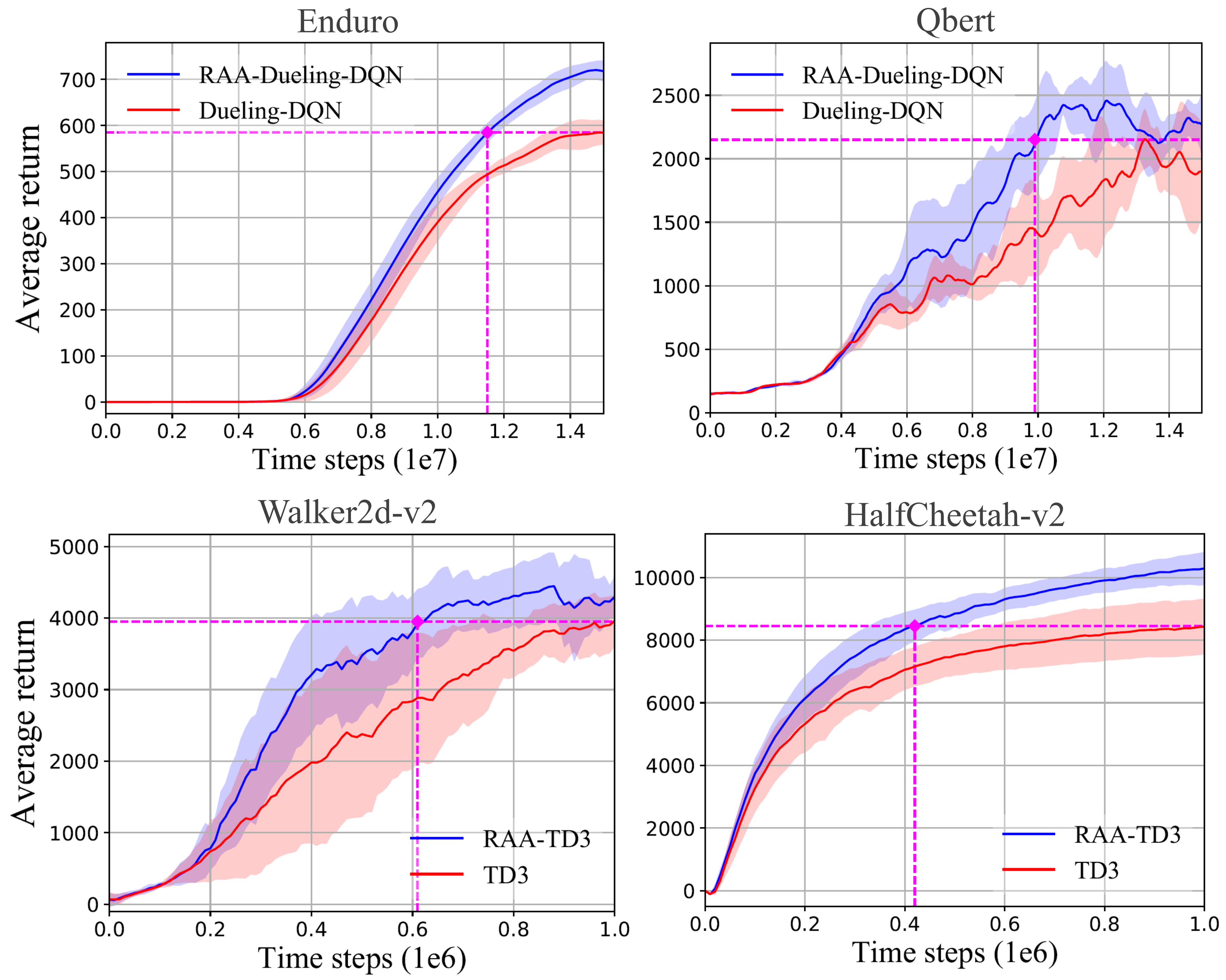



Figure 1: Learning Curves of Dueling-DQN, TD3 and their RAA variants on discrete and continuous control tasks.

### Ablation studies

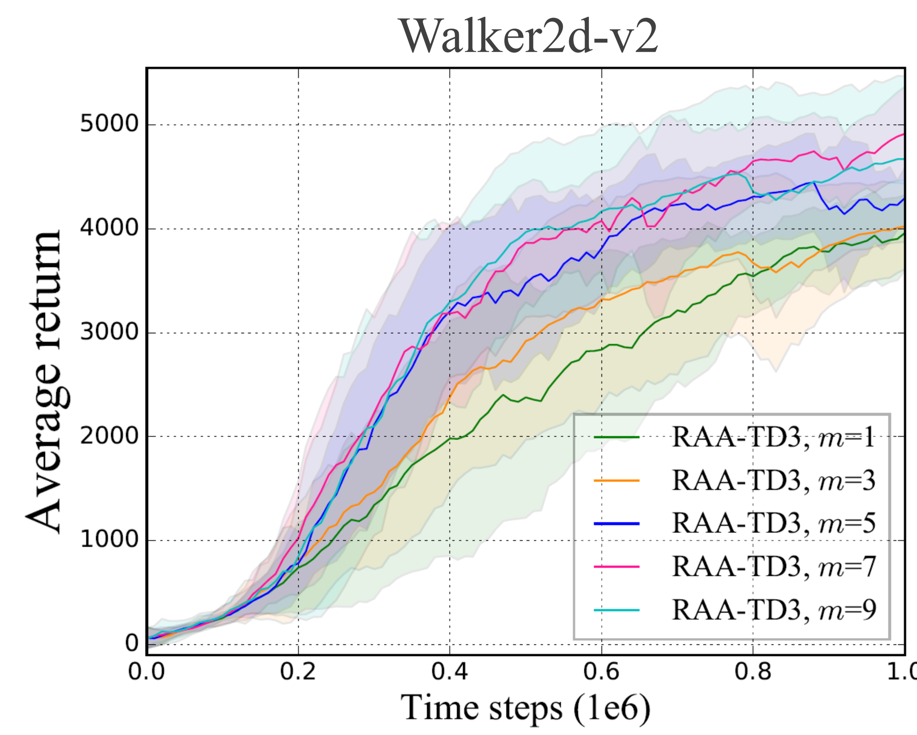



Figure 2: Larger order $m$ leads to faster convergence and better final performance.

GitHub

arXiv

Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver CANADA