Adaptive Temporal-Difference Learning for Policy Evaluation with Per-State Uncertainty Estimates

Hugo Penedones*, Carlos Riquelme*, Damien Vincent, Hartmut Maennel, Timothy Mann, Andre Barreto, Sylvain Gelly, Gergely Neu

Google DeepMind

upf.

In 20 seconds. Temporal Differences learning (TD) propagates approximation errors for policy evaluation. We adaptively choose between the TD and MC targets via MC confidence intervals.

Basic Definitions

Value Function for Policy π:

$$v^{\pi}(s) = \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t \, r(S_t, \pi(S_t)) \,\middle|\, S_0 = s\right]$$

Mean Squared Value Error (MSVE):

$$\mathrm{MSVE}(\widehat{V}) = \mathop{\mathbf{E}}_{S_0 \sim \mu_0}\left[\left(v(S_0) - \widehat{V}(S_0)\right)^2\right]$$

Monte Carlo Target:

$$T_{\mathrm{MC}}\left(s_t^{(i)}\right) := \sum_{k=0}^{n_i - t - 1} \gamma^k \, r\left(s_{t+k}^{(i)}\right)$$

$\mathbf{TD_0}$ Target:

$$T_{\mathrm{TD}(0)}\left(s_t^{(i)}\right) := r\left(s_t^{(i)}\right) + \gamma \, \widehat{V}\left(s_{t+1}^{(i)}\right)$$

Confidence Intervals

We fit **m** networks V_i on the MC target data:

D = {(s, T_{MC}(s)) s ∈ S}.

Given a new state **s**, get **m** predictions $v_i = V_i(s)$.

Assumption: $v_1, v_2, \ldots, v_m$ ~ **F** for some **F**.

Then, compute its **predictive interval** at level 1-α.

In the paper we assume **F = N(μ, σ²)**, for unknown (μ, σ) leading to:

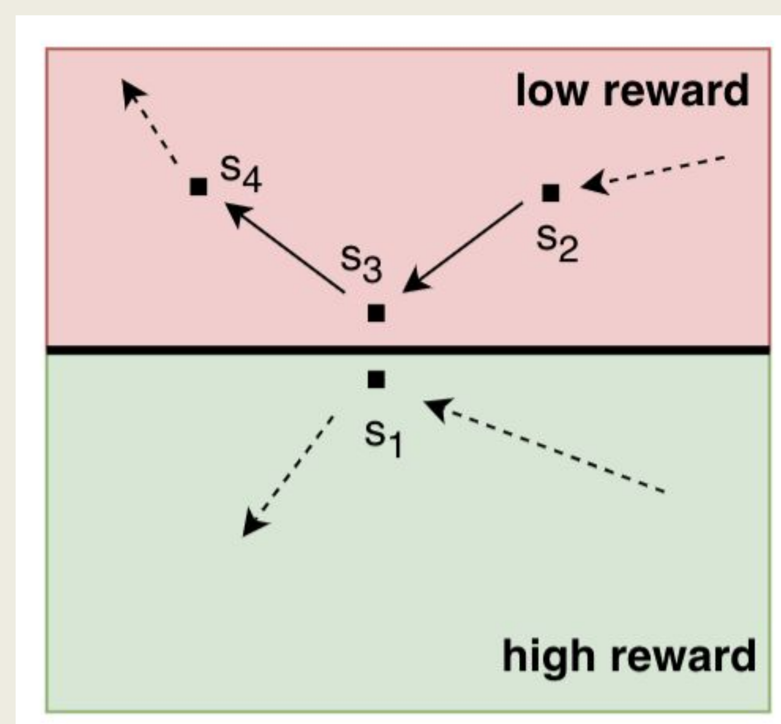
$$\bar{v} - z_\alpha \hat{\sigma}_m \sqrt{1 + 1/m} \leq v_{m+1} \leq \bar{v} + z_\alpha \hat{\sigma}_m \sqrt{1 + 1/m}$$

where $\bar{v} = \sum_i v_i / m$, and $\hat{\sigma}_m^2 = \sum_i (v_i - \bar{v})^2 / (m-1)$

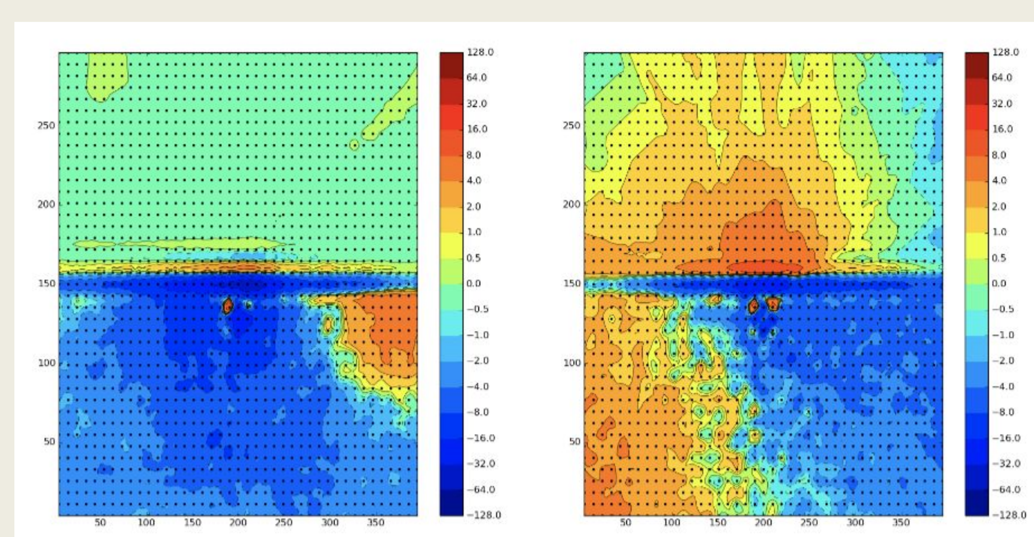
Other ways to estimate uncertainty can be used.

The Problem

Value Leakage with **Function Approximation**



Function approximation tradeoffs (s_1 and s_3) will be propagated through TD updates to s_2.



MC

TD

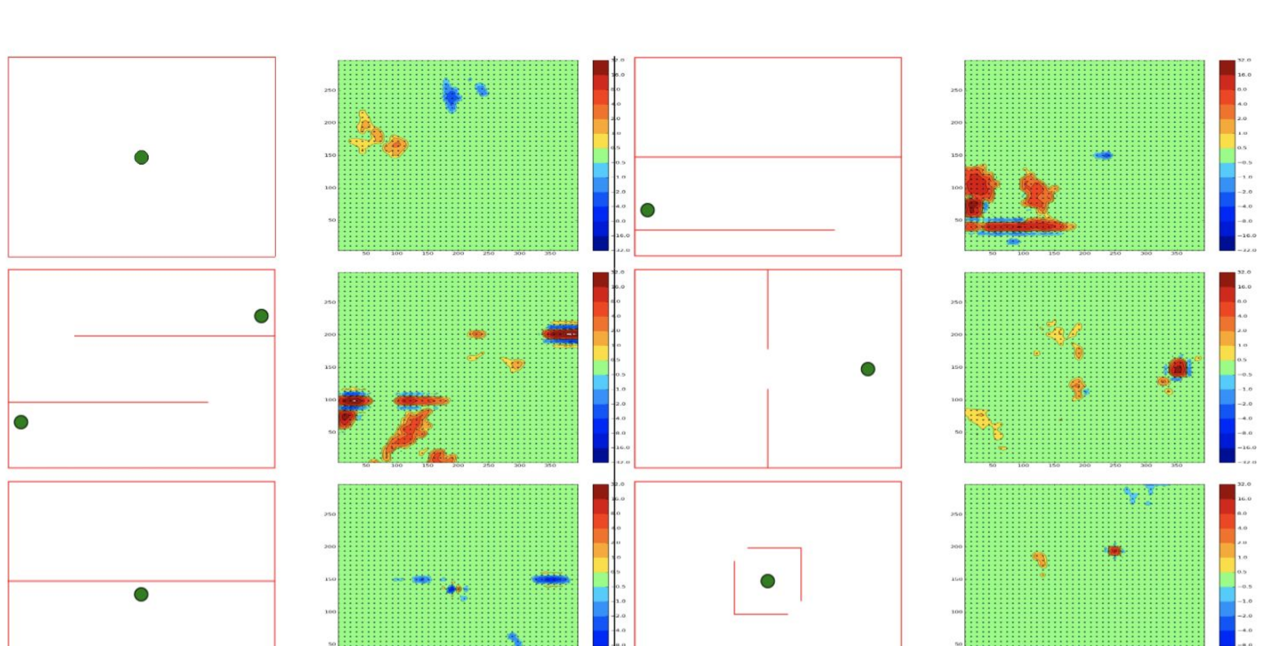
MC versus TD on a simple environment with 2 rooms completely separated by a wall with a single target with positive reward right under the wall.

For each state s, the heatmaps show:

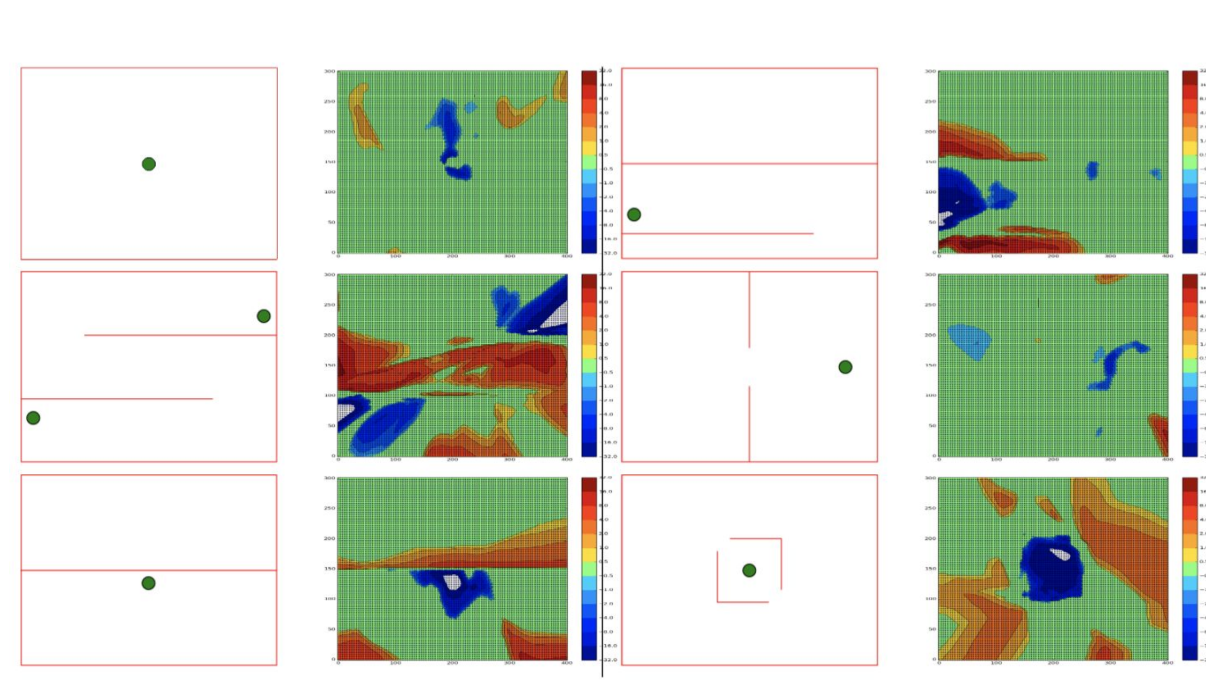
$$\hat{V}(s) - V(s)$$

The true value on the upper half of the plane is zero. MC overestimates the values of a narrow region right above the wall, due to function approximation limitations. With TD, these unavoidable approximation errors also occur, but things get worse when **bootstrap updates propagate errors** to much larger regions.

Confidence in Practice



MC intervals mismatch with ground true value function



TD and MC intervals mismatch. Use MC there.

The Adaptive TD Algorithm

Input: Confidence level $\alpha \in (0,1)$. Trajectories $\tau_1, \ldots, \tau_n$ generated by policy π.

Let S be the set of visited states in $\tau_1, \ldots, \tau_n$. Initialize $\widehat{V}(s) = 0$, for all s.
Compute Monte-Carlo returns dataset as in (3): $D_{\mathrm{MC}} = \{(s, T_{MC}(s))_{s \in S}\}$.
Fit confidence function to D_{MC}: $\mathrm{CI}_{MC}^{\alpha}(s) := (L_{MC}^{\alpha}(s), U_{MC}^{\alpha}(s))$.
repeat
  for $i = 1$ **to** n **do**
    for $t = 1$ **to** $|\tau_i| - 1$ **do**
      $s_t^{(i)}$ is the t-th state of τ_i.
      $T_{\mathrm{TD}(0)} = r(s_t^{(i)}) + \gamma \, \widehat{V}(s_{t+1}^{(i)})$
      if $T_{\mathrm{TD}(0)} \in (L_{MC}^{\alpha}(s_t^{(i)}), U_{MC}^{\alpha}(s_t^{(i)}))$ **then**
        $T_{i,t} \leftarrow T_{\mathrm{TD}(0)}$
      else
        $T_{i,t} \leftarrow (L_{MC}(s_t^{(i)}) + U_{MC}(s_t^{(i)}))/2$
      end if
      Use target $T_{i,t}$ to fit $\widehat{V}(s_t^{(i)})$.
    end for
  end for
until epochs exceeded

Algorithm 1: Adaptive TD

1. Train an ensemble of value networks with MC target.
2. Train a single value network with an adaptive target:

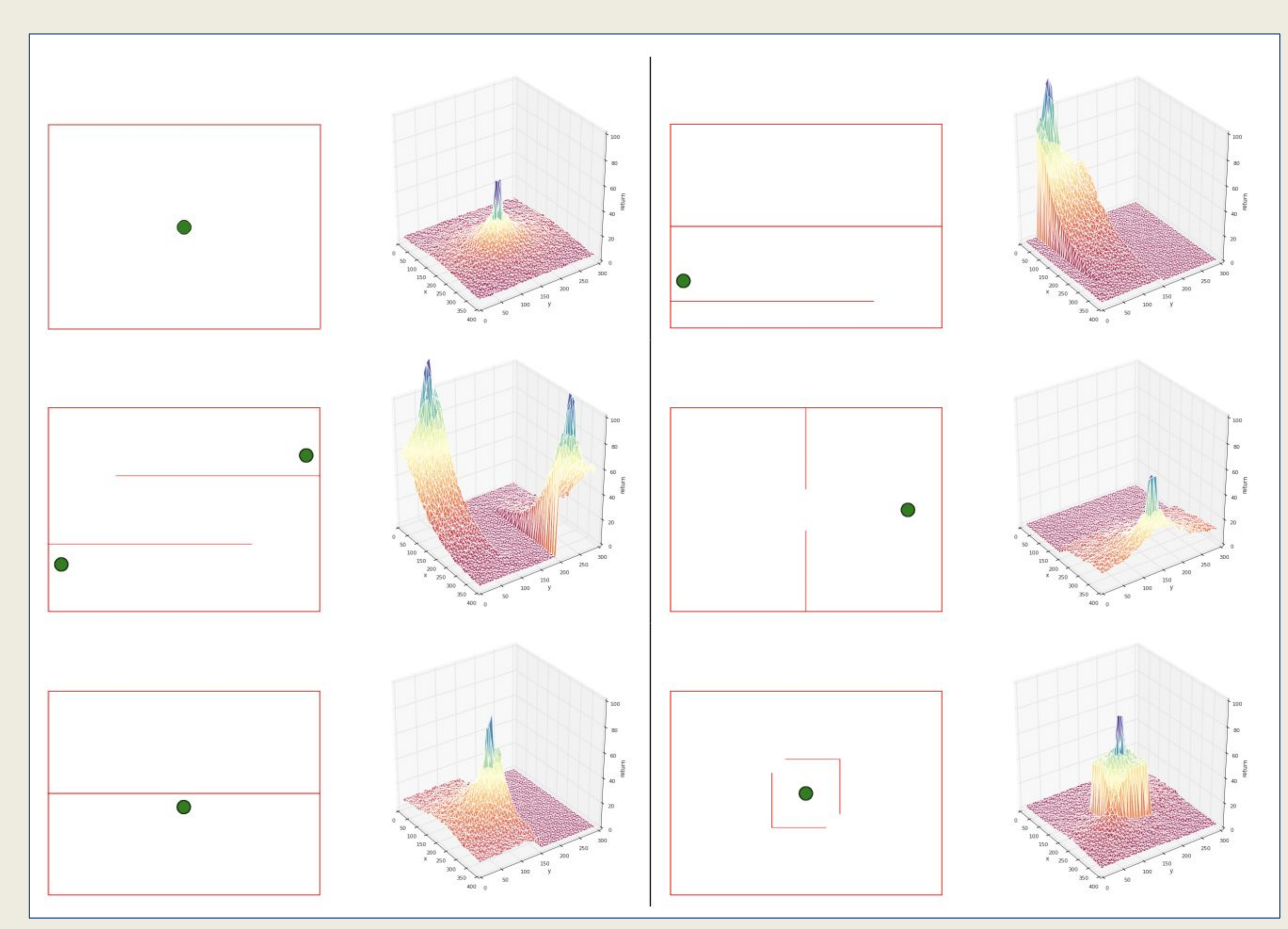
   for each training state s:

   a. By default, use TD target.

   b. If TD target doesn't fall in MC interval ($\mathrm{low}_{\mathrm{MC}}$, $\mathrm{high}_{\mathrm{MC}}$):

      i. Use ($\mathrm{low}_{\mathrm{MC}}$ + $\mathrm{high}_{\mathrm{MC}}$) / 2 as target.

Lab-2D Experiments



MSVE Results