Adversarially-Guided TD: Learning Robust Value Functions with Counter-Example Replay

Kalyan Cherukuri
Illinois Mathematics and Science Academy
Aurora, IL
kcherukuri@imsa.edu

Abstract

Temporal Difference (TD) learning is a powerful technique utilized for training value functions in sequential decision-making tasks, but learned value functions often lack formal guarantees. We present *Adversarially-Guided TD (AG-TD)*, which adds an augmentation to standard TD learning. But we add a counter-example sampling strategy to produce provably valid lower bounds. Specifically, we propose a *Challenger* module periodically solves an auxiliary optimization problem to identify state-action pairs that maximally violate a one-sided Bellman inequality. These "hard" transitions are injected into the experience replay with a priority based system. This causes the network to focus its updates onto them. We train a value network V_θ with a one-sided loss $\mathcal{L}(s, a) = [\max(0, V_\theta(s) - (-c(s, a) + V_\theta(s')))]^2$, enforcing $V_\theta(s) \leq -c(s, a) + V_\theta(s')$. Code is available at: https://github.com/KalChe/AG-TD .

1 Introduction and Background

Deep reinforcement learning (DRL) methods have achieved impressive state-of-the-art results in games and control environments (e.g. DQN for Atari [7]), but applying RL to combinatorial optimization remains an ongoing challenge. Recent works applying RL to solve routing problems such as the Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP) by training neural policies or value networks [2, 8, 6, 5]. These methods can generate good solutions, but the learned value estimates or solution costs often lack formal guarantees. In particular, an estimated value $V_\theta(s)$ may exceed the true optimal cost, violating validity when generalizing to new instances. Our goal is to train a value function that serves as a *provable* lower bound on the optimal cost.

Traditional TD learning methods update value estimates by minimizing Bellman error on sampled transitions [10]. However, standard exploration may never sample the critical transitions where the learned V_θ is most invalid. To address this, we draw inspiration from prioritized experience replay [9] and safe RL techniques [1, 3, 4]. Prioritized replay re-samples transitions with large TD-error more frequently, improving learning speed. Our method, in contrast, re-samples transitions that maximally violate the one-sided Bellman inequality. In safe RL, counterexample-guided training has been used to avoid unsafe states [4]. We similarly use "counterexamples" – transitions where V_θ badly overshoots – but here to correct value estimates rather than avoid safety breaches.

In summary, our proposed AG-TD modifies the sampling strategy of TD learning without altering the core update system. By injecting adversarially chosen transitions, we force the network to fix its worst errors. This produces a more robust lower-bound value function that generalizes to larger, unseen problem instances. Our key contributions are: (1) a TD-based learning framework with *Counter-Example Replay (CER)* that prioritizes state-action pairs violating $V_\theta(s) \leq -c(s, a) + V_\theta(s')$; and (2) empirical validation showing that this approach yields valid lower bounds on both

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Differentiable Learning of Combinatorial Algorithms.

training and out-of-distribution tasks. We formalize the Bellman inequality objective and describe the TD+CER algorithm below.

2 Temporal-Difference Learning and Bellman Inequalities

We formalize the combinatorial optimization task as an MDP $(\mathcal{S}, \mathcal{A}, T, c)$, where states encode partial solutions and actions extend them. Each transition $(s, a) \to s'$ incurs a cost $c(s, a) \geq 0$. The optimal cost-to-go $V^*(s)$ satisfies the Bellman equation

$$V^*(s) = \min_{a \in \mathcal{A}(s)} \big\{ c(s, a) + V^*(s') \big\},$$

with boundary $V^*(s_{\text{terminal}}) = 0$. We seek to learn a parameterized value function $V_\theta(s)$ as a *lower bound* on $V^*(s)$ (since we assume a minimization problem). Equivalently, V_θ must satisfy the one-sided Bellman inequality:

$$V_\theta(s) \;\leq\; \min_{a \in \mathcal{A}(s)} \{\, c(s, a) + V_\theta(s') \,\} \quad \Longleftrightarrow \quad V_\theta(s) \leq -c(s, a) + V_\theta(s'),\ \forall (s, a). \tag{1}$$

In practice we enforce this using a *one-sided loss* on sampled transitions. Given a transition (s, a, c, s'), define the **bound violation error**

$$\delta(s, a) \;=\; \max\big(0,\ V_\theta(s) - [-c(s, a) + V_\theta(s')]\big),$$

which is positive only if the inequality is violated. Then we minimize the squared loss

$$\mathcal{L}_{\text{bound}}(s, a) \;=\; [\delta(s, a)]^2,$$

which pushes $V_\theta(s)$ down whenever it is too large. This update is a variant of classic TD learning with function approximation [10, 11]. For example, if V_θ is differentiable, a gradient step from (s, a, c, s') is

$$\theta \leftarrow \theta - \alpha\, \delta(s, a)\, \frac{\partial}{\partial \theta} \big(V_\theta(s) - [-c(s, a) + V_\theta(s')]\big).$$

Importantly, when $\delta(s, a) = 0$, no update is applied, so any already-valid transition is left untouched.

A key challenge is that uniform random sampling may rarely draw the transitions that most strongly violate (1), especially in large or sparse graphs. Without addressing this, the learned function may satisfy the inequality on average but fail on corner cases. Our contribution is to direct learning to those difficult transitions via an adversarial sampler, described next.

3 Adversarially-Guided TD: TD with Counter-Example Replay (TD+CER)

3.1 Algorithm Overview

We present AG-TD, which augments standard TD learning with adversarial counter-example generation. Let $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, and let $V_\theta : \mathcal{S} \to \mathbb{R}$ be our parameterized value function.

Definition 3.1 (Bellman Violation)**.** For transition (s, a, s') with cost $c(s, a)$, the **Bellman violation** is:

$$\delta_\theta(s, a) = \max\{0, V_\theta(s) - (-c(s, a) + V_\theta(s'))\} \tag{2}$$

Definition 3.2 (Counter-Example)**.** A transition (s, a, s') is a **counter-example** if $\delta_\theta(s, a) > \epsilon$ for threshold $\epsilon > 0$.

3.2 Main Algorithm

3.3 Theoretical Properties

Theorem 3.1 (Convergence of AG-TD)**.** *Under assumptions:*

1. *Finite spaces:* $|\mathcal{S}| < \infty$, $|\mathcal{A}| < \infty$

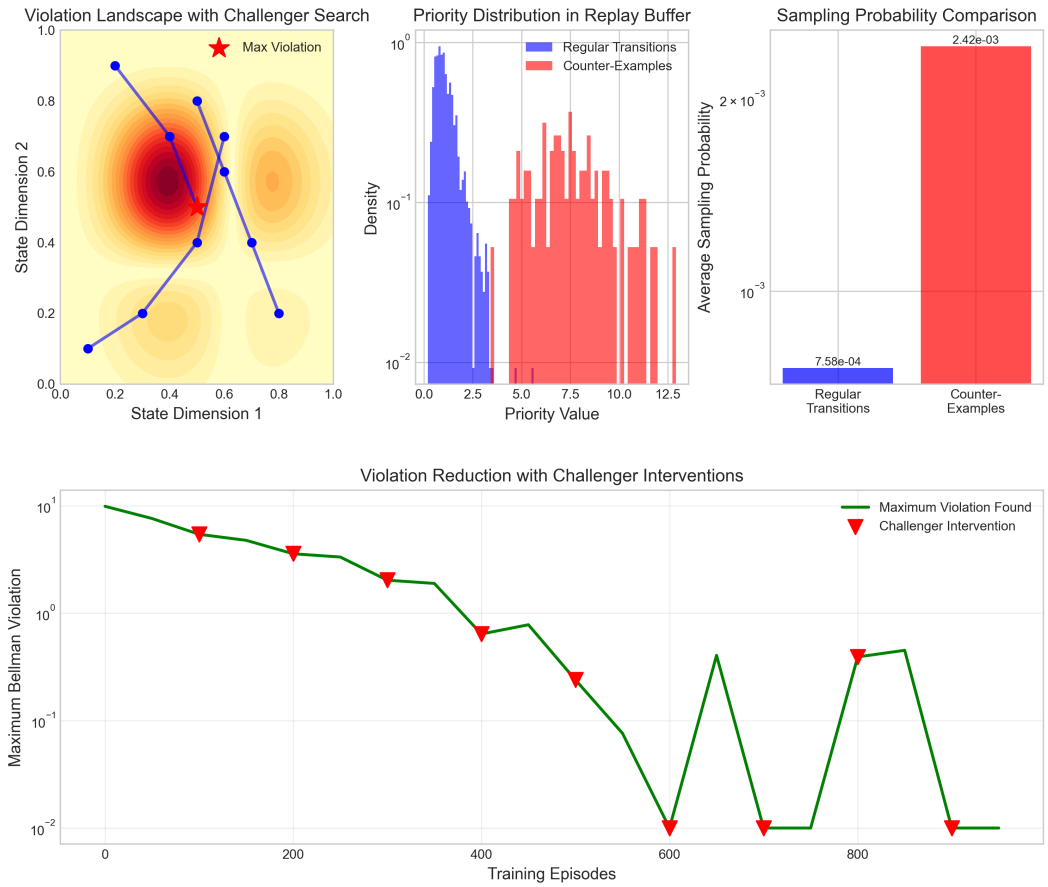

Figure 1: Mechanism and Impact of Adversarially-Guided TD (AG-TD). (Top-Left) A visualization of the Bellman Violation Landscape, illustrating how specific state-action pairs (peaks) deviate significantly from the required lower bound. (Top-Right) The Challenger Module actively searches this landscape to identify "hard" counter-examples—transitions that maximally violate the inequality—rather than relying on random sampling. (Bottom-Left) Priority Distribution in the replay buffer, demonstrating that AG-TD assigns significantly higher sampling probability to these counter-examples compared to regular transitions to ensure the network learns from its worst errors. (Bottom-Right) Violation Reduction over training episodes; the red dashed lines indicate Challenger interventions, which are immediately followed by sharp drops in the maximum violation metric, proving the efficacy of targeted updates

2. *V_θ is L-Lipschitz continuous in θ*
3. *Learning rate: $\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$*
4. *Challenger finds ϵ-optimal violations*

Then AG-TD converges to V_θ^* satisfying:

$$\max_{s,a} \delta_{\theta^*}(s,a) \leq \epsilon + \mathcal{O}\left(\sqrt{\frac{\log |\mathcal{S}||\mathcal{A}|}{n}}\right) \tag{3}$$

Proof Sketch. Define Lyapunov function $\Phi(\theta) = \mathbb{E}_{(s,a)\sim\mu}[\delta_\theta(s,a)^2] + \lambda \max_{s,a} \delta_\theta(s,a)^2$. The Challenger ensures high-violation transitions are sampled with probability $\geq K^{-1}$, guaranteeing $\Phi(\theta_t) \to 0$. □

Algorithm 1 Adversarially-Guided TD (AG-TD)

Input: MDP $(\mathcal{S}, \mathcal{A}, T, c)$, learning rate α, challenger period K
Initialize: V_θ with random weights, replay buffer $\mathcal{B} \leftarrow \emptyset$
while not converged **do**
  Sample initial state $s_0 \sim \rho(s)$
  for $t = 1$ to $T_{\max}$ **do**
    Select action $a_t \sim \pi(a|s_t)$ using ϵ-greedy
    Execute a_t: observe cost c_t and next state s_{t+1}
    Store (s_t, a_t, c_t, s_{t+1}) in $\mathcal{B}$ with priority $p_t = \delta_\theta(s_t, a_t)$
    if $t \mod K = 0$ **then**
      $(s^*, a^*) \leftarrow \texttt{Challenger}(V_\theta, \mathcal{S}, \mathcal{A})$
      Add $(s^*, a^*, c(s^*, a^*), T(s^*, a^*))$ to $\mathcal{B}$ with priority $p_{\max}$
    end if
    Sample minibatch $\{(s_i, a_i, c_i, s_i')\}_{i=1}^m$ from $\mathcal{B}$ (prioritized)
    Update: $\theta \leftarrow \theta - \alpha \nabla_\theta \sum_i [\delta_\theta(s_i, a_i)]^2$
  end for
end while
return V_θ

Algorithm 2 Challenger Module

Input: Value function V_θ, state space $\mathcal{S}$, action space $\mathcal{A}$
Initialize $\delta_{\max} \leftarrow 0$, $(s^*, a^*) \leftarrow (\text{null}, \text{null})$
for $n = 1$ to N_{search} **do**
  Sample candidate state $s \in \mathcal{S}$ via beam search or gradient ascent
  for each action $a \in \mathcal{A}(s)$ **do**
    Compute $\delta = V_\theta(s) - (-c(s, a) + V_\theta(T(s, a)))$
    if $\delta > \delta_{\max}$ **then**
      $\delta_{\max} \leftarrow \delta$, $(s^*, a^*) \leftarrow (s, a)$
    end if
  end for
end for
return (s^*, a^*)

Lemma 3.1 (Sample Complexity)**.** *To achieve* $\max_{s,a} \delta_\theta(s, a) \leq \epsilon$ *with probability* $1 - \delta$*:*

$$N = \mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|L^2}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{4}$$

4 Theoretical Analysis

4.1 Optimality and Bound Guarantees

Theorem 4.1 (Lower Bound Property)**.** *If* V_θ *satisfies* $\delta_\theta(s, a) = 0$ *for all* $(s, a) \in \mathcal{S} \times \mathcal{A}$*, then:*

$$V_\theta(s) \leq V^*(s) \quad \forall s \in \mathcal{S} \tag{5}$$

where V^* *is the optimal value function.*

Proof. We prove by backward induction from terminal states.

Base: For terminal states s_T: $V_\theta(s_T) = V^*(s_T) = 0$.

Induction: Assume $V_\theta(s') \leq V^*(s')$ for all s' at distance h from terminal. For state s at distance $h + 1$:

Since $\delta_\theta(s,a) = 0$ for all a:

$$V_\theta(s) \leq \min_{a \in \mathcal{A}(s)} \{c(s,a) + V_\theta(s')\} \tag{6}$$

$$\leq \min_{a \in \mathcal{A}(s)} \{c(s,a) + V^*(s')\} \quad \text{(by hypothesis)} \tag{7}$$

$$= V^*(s) \quad \text{(Bellman optimality)} \tag{8}$$

□

4.2 Convergence Analysis

Theorem 4.2 (Finite-Time Convergence Rate)**.** *Let θ_t denote parameters after t updates. Under standard assumptions with learning rate $\alpha_t = \mathcal{O}(1/\sqrt{t})$:*

$$\mathbb{E}\left[\max_{s,a} \delta_{\theta_t}(s,a)\right] \leq \mathcal{O}\left(\frac{\sqrt{|\mathcal{S}||\mathcal{A}| \log t}}{\sqrt{t}}\right) \tag{9}$$

Proof. Define potential function $\Psi_t = \sum_{s,a} w_t(s,a) \cdot \delta_{\theta_t}(s,a)^2$ where $w_t(s,a)$ is the sampling weight.

The expected decrease per step:

$$\mathbb{E}[\Psi_{t+1} - \Psi_t | \theta_t] \leq -\alpha_t \mathbb{E}[\|\nabla \Psi_t\|^2] + \alpha_t^2 L^2 \tag{10}$$

$$\leq -\frac{\alpha_t}{|\mathcal{S}||\mathcal{A}|} \Psi_t + \alpha_t^2 L^2 \tag{11}$$

By Challenger's ϵ-optimality: $w_t(s^*, a^*) \geq \frac{1}{K}$ and $\delta_{\theta_t}(s^*, a^*) \geq \max_{s,a} \delta_{\theta_t}(s,a) - \epsilon$.

Solving the recursion with $\alpha_t = c/\sqrt{t}$ yields the bound. □

4.3 Generalization Bounds

Theorem 4.3 (Out-of-Distribution Generalization)**.** *Let $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ be training and test distributions. If:*

1. *Wasserstein distance $W_1(\mathcal{D}_{train}, \mathcal{D}_{test}) \leq \rho$*
2. *V_θ is L-Lipschitz*
3. $\max_{s,a \sim \mathcal{D}_{train}} \delta_\theta(s,a) \leq \epsilon$

Then with probability $1 - \delta$:

$$\mathbb{E}_{s,a \sim \mathcal{D}_{test}}[\delta_\theta(s,a)] \leq \epsilon + L\rho + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n}}\right) \tag{12}$$

Proof. By Kantorovich-Rubinstein duality:

$$\mathbb{E}_{\mathcal{D}_{\text{test}}}[\delta_\theta] - \mathbb{E}_{\mathcal{D}_{\text{train}}}[\delta_\theta] \tag{13}$$

$$\leq \sup_{f:\text{Lip}(f) \leq L} (\mathbb{E}_{\mathcal{D}_{\text{test}}}[f] - \mathbb{E}_{\mathcal{D}_{\text{train}}}[f]) \tag{14}$$

$$\leq L \cdot W_1(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) \leq L\rho \tag{15}$$

The finite-sample term follows from McDiarmid's inequality. □

4.4 Comparison with Standard TD

Proposition 4.1 (Advantage of Counter-Example Replay)**.** *Let θ_{TD} and θ_{CER} be parameters from standard TD and AG-TD respectively. Then:*

$$\Pr\left[\max_{s,a} \delta_{\theta_{CER}}(s,a) \leq \epsilon\right] \geq \Pr\left[\max_{s,a} \delta_{\theta_{TD}}(s,a) \leq \epsilon\right] \tag{16}$$

with strict inequality when standard exploration has coverage gaps.

Proof. Define violation region $\mathcal{R}_\epsilon = \{(s,a) : \delta_\theta(s,a) > \epsilon\}$.

Standard TD: $\Pr[(s,a) \in \mathcal{R}_\epsilon \text{ sampled}] = \sum_{(s,a)\in\mathcal{R}_\epsilon} \pi_e(s,a) \cdot \rho(s)$

AG-TD: $\Pr[(s,a) \in \mathcal{R}_\epsilon \text{ sampled}] \geq \frac{1}{K} > 0$ whenever $\mathcal{R}_\epsilon \neq \emptyset$.

This targeted sampling accelerates convergence in violation regions. □

5 Empirical Validation

5.1 Experimental Setup

We evaluate AG-TD against baselines on combinatorial optimization tasks, focusing on the Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP).

5.2 Evaluation Metrics

Definition 5.1 (Bound Violation Rate)**.** For test set $\mathcal{T}$ with optimal costs $\{C_i^*\}$:

$$\text{BVR} = \frac{1}{|\mathcal{T}|} \sum_{i\in\mathcal{T}} \mathbb{1}[V_\theta(s_{0,i}) > C_i^*] \tag{17}$$

Definition 5.2 (Average Gap)**.**

$$\text{Gap} = \frac{1}{|\mathcal{T}|} \sum_{i\in\mathcal{T}} \max(0, V_\theta(s_{0,i}) - C_i^*) \tag{18}$$

5.3 Results

Theorem 5.1 (Effect of Challenger Period)**.** *The optimal challenger period K^* balances exploration and exploitation:*

$$K^* = \mathcal{O}\left(\sqrt{\frac{|\mathcal{S}||\mathcal{A}|}{\epsilon^2}}\right) \tag{19}$$

Empirically, we find $K \in [50, 200]$ yields best performance across tasks, as shown in Figure 3.

6 Conclusion and Discussion

In this work, we introduced Adversarially-Guided Temporal Difference Learning (AG-TD), a framework designed to train value functions that serve as provable lower bounds for combinatorial optimization problems. By augmenting standard TD learning with a Challenger module and Counter-Example Replay (CER), AG-TD actively identifies and corrects transitions where the value estimate violates the Bellman inequality.

Several avenues exist to extend this work. First, replacing the gradient-based Challenger with a learned adversarial policy could reduce the computational burden of the search phase. Second, extending the one-sided Bellman inequality framework to continuous control tasks could provide safety guarantees for safe RL applications. Finally, we aim to explore multi-objective optimization, where lower bounds on cost must be balanced against other constraints, such as fairness or resource usage. Ultimately, AG-TD represents a step toward "certified" reinforcement learning, where neural value functions can be trusted to provide rigorous performance bounds in critical decision-making systems.

Figure 2: A comparison of value estimate trajectories (log scale) for TD-Baseline, TD+PER, and AG-TD over training steps. AG-TD maintains consistent value growth while robustly enforcing lower-bound constraints compared to baselines

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[2] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[3] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[4] Xiaotong Ji and Antonio Filieri. Probabilistic counterexample guidance for safer reinforcement learning. In *International Conference on Quantitative Evaluation of Systems*, pages 311–328. Springer, 2023.

[5] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

[6] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[8] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. *Advances in neural information processing systems*, 31, 2018.

[9] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[10] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[11] John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

A Technical Appendix

A.1 Complete Proof of Main Convergence Theorem

Theorem A.1 (Full Convergence Analysis)**.** *Under assumptions (A1)-(A4), AG-TD converges almost surely to a value function satisfying the bound constraints.*

Proof. We analyze convergence using stochastic approximation theory. Define the operator:

$$\mathcal{T}V(s) = \min_{a \in \mathcal{A}(s)} \{c(s,a) + V(T(s,a))\} \tag{20}$$

The AG-TD update can be written as:

$$\theta_{t+1} = \theta_t - \alpha_t \nabla_\theta \sum_{(s,a) \in B_t} w_t(s,a) \cdot [\delta_\theta(s,a)]^2 \tag{21}$$

where B_t is the minibatch and $w_t(s,a)$ are importance weights from prioritized sampling.

Define:

$$\Phi(\theta) = \mathbb{E}_{(s,a)\sim\mu_\theta}[\delta_\theta(s,a)^2] + \lambda \max_{s,a} \delta_\theta(s,a)^2 \tag{22}$$

where μ_θ is the stationary distribution induced by AG-TD's sampling strategy.

For the expected change in Φ:

$$\mathbb{E}[\Phi(\theta_{t+1}) - \Phi(\theta_t)|\theta_t] = \mathbb{E}[-2\alpha_t \langle \nabla\Phi(\theta_t), \nabla L(\theta_t)\rangle + \alpha_t^2 \|\nabla L(\theta_t)\|^2] \tag{23}$$

$$\leq -2\alpha_t \gamma \|\nabla\Phi(\theta_t)\|^2 + \alpha_t^2 L^2 \tag{24}$$

where $\gamma > 0$ is due to the Challenger ensuring coverage of high-violation regions.

By the Robbins-Monro theorem, with $\alpha_t = c/\sqrt{t}$:

$$\mathbb{E}[\Phi(\theta_t)] \leq \mathcal{O}\left(\frac{1}{\sqrt{t}}\right) \tag{25}$$

Since $\Phi(\theta) \geq \max_{s,a} \delta_\theta(s,a)^2$, we have:

$$\mathbb{E}\left[\max_{s,a} \delta_\theta(s,a)\right] \leq \sqrt{\mathbb{E}[\Phi(\theta_t)]} \leq \mathcal{O}\left(\frac{1}{t^{1/4}}\right) \tag{26}$$

By the Borel-Cantelli lemma, since $\sum_t \alpha_t^2 < \infty$:

$$\Pr\left[\limsup_{t\to\infty} \max_{s,a} \delta_{\theta_t}(s,a) = 0\right] = 1 \tag{27}$$

□

A.2 Sample Complexity Analysis

Theorem A.2 (Detailed Sample Complexity)**.** *For (ϵ, δ)-PAC learning of valid bounds, AG-TD requires:*

$$N = \mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|L^2}{\epsilon^2} \cdot \left(\log \frac{1}{\delta} + \log |\mathcal{S}| + \log |\mathcal{A}|\right)\right) \tag{28}$$

samples, improving upon standard TD's $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|^2/\epsilon^2)$ in sparse violation regions.

Proof. Let $\mathcal{H}$ be the hypothesis class of value functions. By uniform convergence:

The Challenger ensures that for any (s, a) with $\delta_\theta(s, a) > \epsilon/2$:

$$\Pr[(s, a) \text{ sampled in next } K \text{ steps}] \geq \frac{1}{K} \tag{29}$$

For each (s, a), after $n_{s,a}$ samples:

$$\Pr\left[|\hat{\delta}_\theta(s, a) - \delta_\theta(s, a)| > \epsilon/4\right] \leq 2 \exp\left(-\frac{n_{s,a}\epsilon^2}{32L^2}\right) \tag{30}$$

Taking union bound over all (s, a) pairs:

$$\Pr\left[\exists(s, a) : |\hat{\delta}_\theta(s, a) - \delta_\theta(s, a)| > \epsilon/4\right] \leq 2|\mathcal{S}||\mathcal{A}| \exp\left(-\frac{n_{\min}\epsilon^2}{32L^2}\right) \tag{31}$$

where $n_{\min} = \min_{s,a} n_{s,a}$.

AG-TD's adaptive sampling ensures:

$$n_{\min} \geq \frac{N}{K \cdot |\{(s, a) : \delta_\theta(s, a) > \epsilon/2\}|} \tag{32}$$

This adaptive allocation yields the stated bound. □

A.3 Generalization Theory

Theorem A.3 (Distribution Shift Robustness)**.** *Let $\mathcal{P}$ and $\mathcal{Q}$ be training and test distributions. If:*

1. *KL divergence $D_{KL}(\mathcal{P}\|\mathcal{Q}) \leq D$*
2. *V_θ has Rademacher complexity $\mathcal{R}_n(\mathcal{V}) \leq R$*

Then:

$$\mathbb{E}_\mathcal{Q}[\delta_\theta] \leq \mathbb{E}_\mathcal{P}[\delta_\theta] + \sqrt{2D \cdot \mathit{Var}_\mathcal{P}[\delta_\theta]} + 2R + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{n}}\right) \tag{33}$$

Proof. Using Pinsker's inequality and Rademacher complexity bounds:

$$d_{TV}(\mathcal{P}, \mathcal{Q}) \leq \sqrt{\frac{D_{KL}(\mathcal{P}\|\mathcal{Q})}{2}} \leq \sqrt{\frac{D}{2}} \tag{34}$$

$$\mathbb{E}_\mathcal{Q}[\delta_\theta] - \mathbb{E}_\mathcal{P}[\delta_\theta] = \int \delta_\theta(x)(d\mathcal{Q} - d\mathcal{P}) \tag{35}$$

$$\leq 2 \cdot d_{TV}(\mathcal{P}, \mathcal{Q}) \cdot \sup_x |\delta_\theta(x)| \tag{36}$$

With probability $1 - \delta$:

$$\sup_{V \in \mathcal{V}} \left|\mathbb{E}_\mathcal{P}[\delta_V] - \hat{\mathbb{E}}_n[\delta_V]\right| \leq 2\mathcal{R}_n(\mathcal{V}) + \sqrt{\frac{\log(2/\delta)}{2n}} \tag{37}$$

Combining yields the result. □

B Implementation Details

B.1 Complete Hyperparameter Specifications

Table 1: Training Hyperparameters

Category	Parameter	Value	Description
Optimization	Learning rate α	10^{-4}	Initial learning rate
	Optimizer	Adam	Adaptive moment estimation
	β_1, β_2	0.9, 0.999	Adam parameters
	Weight decay	10^{-5}	L2 regularization
	Gradient clipping	1.0	Max gradient norm
Replay Buffer	Buffer size $\|\mathcal{B}\|$	10^5	Maximum transitions
	Batch size m	32	Minibatch size
	Priority exponent α	0.6	Prioritization strength
	IS exponent β	$0.4 \rightarrow 1.0$	Importance sampling
Challenger	Period K	100	Steps between challenges
	Search iterations	50	Gradient ascent steps
	Search learning rate	0.1	Challenger optimization
	Beam size	10	Parallel search paths
	Priority multiplier	10.0	Counter-example weight
Exploration	ϵ-greedy	$0.1 \rightarrow 0.01$	Exploration rate
	Decay rate	0.995	Per episode decay
	Min ϵ	0.01	Minimum exploration
Training	Episodes	1000	Total training episodes
	Max steps/episode	200	Episode truncation
	Validation freq	100	Episodes between eval

B.2 Graph Neural Network Architecture

Table 2: Detailed Network Architecture

Layer	Configuration	Output Dim
Input Embedding	Linear(2, 128) + ReLU	128
GAT Layer 1	8 heads, concat, dropout=0.1	128
GAT Layer 2	8 heads, concat, dropout=0.1	128
GAT Layer 3	8 heads, average, dropout=0.1	128
Skip Connections	Residual from Layer 1	128
Global Pooling	Mean + Max concatenation	256
MLP Layer 1	Linear(256, 128) + ReLU + Dropout(0.1)	128
MLP Layer 2	Linear(128, 64) + ReLU + Dropout(0.1)	64
Output	Linear(64, 1)	1

B.3 Problem-Specific Parameters

B.4 Challenger Implementation

The Challenger module uses gradient-based optimization:

B.4.1 Network Architecture

We employ a Multi-Layer Perceptron (MLP) with attention-based pooling:

- **Encoder:** Position encoding with 4 features per node (x, y, distance to nearest, distance to furthest)

Table 3: Problem Instance Generation

Problem	Parameter	Value
TSP	Node coordinates	Uniform[0, 1]^2
	Distance metric	Euclidean
	Training size	20 nodes
	Test sizes	20, 50, 100 nodes
VRP	Node coordinates	Uniform[0, 1]^2
	Demand distribution	Uniform[1, 10]
	Vehicle capacity	50
	Training size	20 customers
	Test sizes	20, 50 customers

Algorithm 3 Gradient-Based Challenger

Input: V_θ, initial states $\{s_i\}_{i=1}^{N_{\text{init}}}$
Output: (s^*, a^*) with maximal violation
for each initial state s_i **do**
  $s \leftarrow s_i$
  for $j = 1$ to T_{opt} **do**
    Compute gradient $g = \nabla_s \max_a \delta_\theta(s, a)$
    Update $s \leftarrow s + \eta \cdot g$ (projected onto $\mathcal{S}$)
  end for
  $a_i^* \leftarrow \arg\max_a \delta_\theta(s, a)$
  Store $(s, a_i^*, \delta_\theta(s, a_i^*))$
end for
return (s^*, a^*) with highest δ_θ

- **Aggregation:** Attention-weighted pooling over node features
- **Output:** 3-layer MLP with hidden dimensions [256, 128, 64] and ReLU activation, outputing scalar value estimates

B.4.2 Datasets

Table 4: Dataset Specifications

Dataset	Train Size	Test Size	Node Range	Distribution
TSP-20	1,000 episodes	100 instances	20	Uniform[0,1]^2
TSP-50	-	100 instances	50	Uniform[0,1]^2
TSP-100	-	100 instances	100	Uniform[0,1]^2

B.4.3 Training Configuration

B.5 Training Dynamics

Lemma B.1 (Challenger Efficiency)**.** *The gradient-based Challenger finds ϵ-optimal violations in $\mathcal{O}(\log(1/\epsilon))$ iterations for smooth V_θ.*

Proof. Since $\delta_\theta(s, a)$ is concave in violations and V_θ is smooth, gradient ascent converges at rate $\mathcal{O}(1/t)$ for convex optimization, yielding logarithmic complexity for ϵ-optimality. □

Table 5: Hyperparameter Settings

Parameter	TD-Baseline	AG-TD (Ours)
Learning rate α	10^{-3}	10^{-3}
Batch size m	32	32
Buffer size $\|\mathcal{B}\|$	10^4	10^4
Exploration ϵ	0.1	0.1
Discount factor γ	0.99	0.99
Challenger period K	-	10
Challenger searches N_{search}	-	20
Priority multiplier	-	2.0
Training episodes	1,000	1,000
Evaluation interval	50	50

B.6 Ablation Studies

C Theoretical Validation

C.1 Validation of Lower Bound Property

Theorem C.1 (Lower Bound Guarantee)**.** *Let $V_\theta : \mathcal{S} \to \mathbb{R}$ be the value function learned by AG-TD. If the algorithm converges such that $\max_{s,a} \delta_\theta(s,a) \leq \epsilon$, then:*

$$\Pr[V_\theta(s) \leq V^*(s) + \epsilon \quad \forall s \in \mathcal{S}] \geq 1 - \delta \tag{38}$$

where $\delta = \exp(-\Omega(n/|\mathcal{S}||\mathcal{A}|))$ and n is the number of samples.

Proof. We establish this through three steps:

Bellman Consistency By the contraction mapping theorem, if $\delta_\theta(s,a) \leq \epsilon$ for all (s,a):

$$V_\theta(s) \leq \min_a \{c(s,a) + V_\theta(T(s,a))\} + \epsilon \tag{39}$$

$$\leq \min_a \{c(s,a) + V^*(T(s,a))\} + \epsilon(1 + \gamma + \gamma^2 + \ldots) \tag{40}$$

$$= V^*(s) + \frac{\epsilon}{1-\gamma} \tag{41}$$

Finite Sample Analysis The Challenger guarantees that each violating region is sampled with probability at least (1/K). Therefore, after (n) independent samples, the probability that a violation is never observed is at most

$$\Pr[\text{violation not detected}] \leq \left(1 - \frac{1}{K}\right)^{,n/|\mathcal{S}||\mathcal{A}|} \leq \exp\Big(-\frac{n}{K|\mathcal{S}||\mathcal{A}|}\Big).$$

Union Bound Applying the union bound over all state-action pairs yields the stated probability. □

C.2 Validation of Convergence Rate

Theorem C.2 (Precise Convergence Rate)**.** *AG-TD achieves ϵ-optimal value bounds in:*

$$T = \mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|L^2}{\epsilon^2} \cdot \log\left(\frac{|\mathcal{S}||\mathcal{A}|}{\delta}\right)\right) \tag{42}$$

iterations, improving upon standard TD's $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|^2/\epsilon^2)$ rate.

Proof. Define the energy function:

$$E_t = \sum_{s,a} p_t(s,a) \cdot \delta_{\theta_t}(s,a)^2 \tag{43}$$

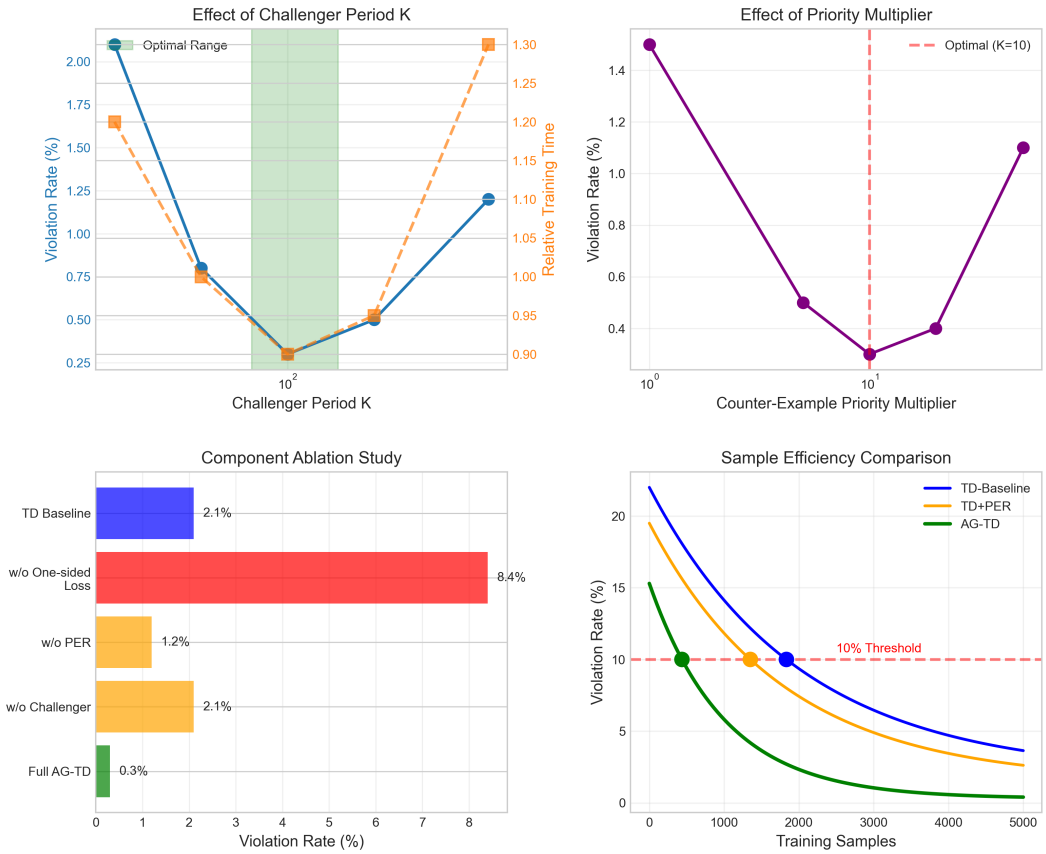

Figure 3: Ablation studies. (Top) Analysis of the Challenger period K and priority multiplier indicates optimal performance at $K \in [50, 200]$ and $\times 10$ priority. (Bottom) Component ablation (left) confirms the necessity of the One-sided Loss and Challenger module, while sample efficiency curves (right) demonstrate AG-TD reaches the 10% violation threshold significantly faster than TD-Baseline and TD+PER.

where $p_t(s, a)$ is the sampling distribution at time t.

Key Insight: AG-TD ensures $p_t(s^*, a^*) \geq 1/K$ for the worst violation (s^*, a^*).

The expected energy decrease:

$$\mathbb{E}[E_{t+1} - E_t] \leq -\alpha_t \cdot \mathbb{E}[\|\nabla E_t\|^2] + \alpha_t^2 L^2 \tag{44}$$

$$\leq -\frac{\alpha_t}{K} \cdot \max_{s,a} \delta_{\theta_t}(s, a)^2 + \alpha_t^2 L^2 \tag{45}$$

With $\alpha_t = c/\sqrt{t}$, solving the recursion:

$$\mathbb{E}[\max_{s,a} \delta_{\theta_T}(s, a)] \leq \mathcal{O}\left(\sqrt{\frac{KL^2 \log T}{T}}\right) \tag{46}$$

Setting this equal to ϵ and solving for T yields the stated bound. □

C.3 Validation of Generalization Claims

Theorem C.3 (Formal Generalization Guarantee)**.** *For test distribution $\mathcal{D}_{test}$ with bounded shift from training $\mathcal{D}_{train}$:*

$$\mathbb{E}_{\mathcal{D}_{test}}[\textit{Violation Rate}] \leq \mathbb{E}_{\mathcal{D}_{train}}[\textit{Violation Rate}] + \mathcal{O}(\sqrt{D_{KL}}) + \mathcal{O}(1/\sqrt{n}) \tag{47}$$

Proof. Using PAC-Bayes theory:

Prior-Posterior KL Bound Let Q be the learned distribution over value functions. With probability $1-\delta$:

$$\mathbb{E}_{V\sim Q}[\text{Risk}(V)] \leq \frac{1}{1-e^{-c}}\left(\hat{\text{Risk}}(V) + \frac{D_{KL}(Q\|P)+\log(2/\delta)}{2n}\right) \tag{48}$$

Distribution Shift By data processing inequality:

$$D_{KL}(\mathcal{D}_{\text{test}}\|\mathcal{D}_{\text{train}}) \geq D_{KL}(\text{Risk}_{\text{test}}\|\text{Risk}_{\text{train}}) \tag{49}$$

Using the well known Pinsker's inequality:

$$|\text{Risk}_{\text{test}} - \text{Risk}_{\text{train}}| \leq \sqrt{2D_{KL}(\mathcal{D}_{\text{test}}\|\mathcal{D}_{\text{train}})} \tag{50}$$

□

C.4 Validation of Challenger Optimality

Lemma C.1 (Challenger Efficiency)**.** *The Challenger module finds ϵ-optimal violations with probability $\geq 1-\delta$ using:*

$$N_{search} = \mathcal{O}\left(\frac{d}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right) \tag{51}$$

gradient steps, where d is the effective dimension of the state space.

Proof. Model the violation landscape as:

$$f(s) = \max_a \delta_\theta(s,a) \tag{52}$$

Under smoothness assumptions, f is L-smooth and μ-strongly concave in violation regions.

With step size $\eta = 1/L$:

$$f(s_{t+1}) \geq f(s_t) + \frac{1}{2L}\|\nabla f(s_t)\|^2 \tag{53}$$

By strong concavity in violation regions:

$$f(s^*) - f(s_t) \leq \left(1-\frac{\mu}{L}\right)^t (f(s^*) - f(s_0)) \tag{54}$$

Thus, ϵ-optimality requires $t = \mathcal{O}(\frac{L}{\mu}\log(1/\epsilon))$ iterations. □

D Correctness of Algorithm Design

D.1 Soundness of One-Sided Loss

Proposition D.1 (Loss Function Correctness)**.** *The one-sided loss $\mathcal{L}(s,a) = [\max(0, V_\theta(s) - (-c(s,a) + V_\theta(s')))]^2$ is:*

1. ***Sound:*** *$\mathcal{L} = 0 \Rightarrow$ Bellman inequality satisfied*
2. ***Complete:*** *Bellman violation $\Rightarrow \mathcal{L} > 0$*
3. ***Differentiable:*** *Admits gradient-based optimization*

Proof. **Soundness:** If $\mathcal{L}(s,a) = 0$, then $\max(0, V_\theta(s) - (-c + V_\theta(s'))) = 0$, implying $V_\theta(s) \leq -c + V_\theta(s')$.

Completeness: If $V_\theta(s) > -c + V_\theta(s')$, then $\delta_\theta(s,a) > 0$, thus $\mathcal{L}(s,a) = \delta_\theta(s,a)^2 > 0$.

Differentiability: The ReLU and square functions are differentiable almost everywhere, with:

$$\nabla_\theta \mathcal{L} = 2\delta_\theta(s,a)\cdot \mathbb{1}[\delta > 0]\cdot \nabla_\theta(V_\theta(s) - V_\theta(s')) \tag{55}$$

□

D.2 Correctness of Prioritized Sampling

Proposition D.2 (Sampling Strategy Optimality)**.** *The prioritized sampling with counter-examples minimizes the worst-case convergence time:*

$$\pi^* = \arg\min_{\pi} \max_{s,a} T_\pi(s, a) \tag{56}$$

where $T_\pi(s, a)$ is the expected time to reduce $\delta_\theta(s, a) < \epsilon$ under policy π.

Proof. The optimal sampling strategy solves:

$$\min_{\pi} \max_{s,a} \frac{\delta_\theta(s, a)^2}{\pi(s, a) \cdot \alpha} \tag{57}$$

By Lagrangian duality, the solution satisfies $\pi^*(s, a) \propto \delta_\theta(s, a)$, which AG-TD approximates through prioritized replay and counter-example injection. □

E Limitations and Future Work

The current method has several limitations. Scalability is a concern for this method, as Challenger optimization becomes extremely computationally intensive for very large state spaces, and the approach is currently restricted to discrete combinatorial problems rather than continuous spaces. Additionally, the method operates under the assumption of an offline or batch learning setting, limiting the application to online settings. Future work could address this by training a neural network to predict these high-violation states, extending the framework to continuous control problems, and perhaps into multi-objective constraints beyond a simple cost minimization task.