TARGET VALUE AS POTENTIAL: REWARD SHAPING IN DEEP REINFORCEMENT LEARNING BY TRANSFERRING PRIOR KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose Target Value as Potential (TAP), a simple reward shaping method that accelerates learning in deep reinforcement learning (RL) by transferring prior knowledge. Although reward shaping provides a natural mechanism for leveraging prior knowledge, theoretical analysis and empirical results show that its effectiveness depends on the representation quality of the prior knowledge and that priors constructed using existing methods may not reliably improve performance. To address this challenge, TAP introduces a new approach to constructing priors by using only the critic's target value as the shaping potential within the classical Potential-Based Reward Shaping (PBRS) framework. Consequently, TAP requires neither additional representational structures nor extra hyperparameters. Our analysis and experiments show that TAP accelerates convergence and improves cumulative returns over standard DRL baselines, including TD3 and D4PG. Across tasks in the DeepMind Control Suite, TAP consistently outperforms existing reward shaping methods, such as Dynamic Potential-Based Reward Shaping (DPBRS), and more general methods for incorporating prior knowledge into RL, such as Heuristic-Guided Reinforcement Learning (HuRL). Furthermore, initial results integrating TAP with MR.Q, a representation-learning method for pixel-based observations, also results in improvements across two visual control environments. These results suggest that TAP remains effective even when the shaping potential is derived from high-dimensional learned representations rather than true environment states.

1 INTRODUCTION

Reward shaping is a widely used technique for improving learning efficiency in reinforcement learning. Within the standard MDP framework, potential-based reward shaping (PBRS) provides a principled approach with policy invariance: an additive shaping term preserves the set of optimal policies if and only if it can be expressed as a potential difference (Ng et al., 1999). Reward shaping also provides a natural mechanism for **transferring** prior information across tasks by encoding reusable structure in the shaping potential. While this classical framework is well understood and effective in practice, a central **challenge** lies in specifying shaping potentials that encode useful prior information without degrading learning performance.

Existing approaches to specifying shaping potentials can be broadly grouped into two categories. **1)** External priors construct shaping signals from structured knowledge, such as simplified or relaxed task variants, planning heuristics (Hoffmann & Nebel, 2001; Buffet & Hoffmann, 2010; Adamczyk et al., 2023), or via policy transfer from previously learned tasks (Brys et al., 2015). While effective when such knowledge is well aligned with the task, their performance depends critically on **the quality and relevance of the prior**. Prior work has shown that the choice of representation—whether through reward shaping or value initialization— significantly impacts the learning outcomes both positively and negatively (Koenig & Simmons, 1996; Ng et al., 1999). **2)** Data- or experience-driven approaches instead learn shaping signals from data, for example by fitting auxiliary reward models or potentials (Brys et al., 2015; Koenig & Simmons, 1996; Burda et al., 2018; Devidze et al., 2022; Ma et al., 2024; Hu et al., 2020; Grzes & Kudenko, 2009), or by introducing intrinsic-motivation bonuses that encourage novelty and state-space coverage (Burda et al., 2018; Tang et al., 2017; Mavor-Parker et al., 2022). However, such learned signals can be **noisy or misaligned with the task objective**, leading to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

distraction, slower convergence, or suboptimal policies. Canonical curiosity-driven methods such as ICM and VIME exhibit similar sensitivities when poorly scaled or misspecified (Pathak et al., 2017; Houthooft et al., 2016).

A common **limitation** of the above approaches is that effective learning acceleration often depends on either learning a separate reward model (Ma et al., 2024; Hu et al., 2020; Grzes & Kudenko, 2008) or designing a shaping signal that is informative about the target task's long-term value structure (Grzes & Kudenko, 2009; Ng et al., 1999). Both results in additional complexity and potential sources of error that may even hinder learning. Moreover, methods that do not adhere to the PBRS formulation generally lack policy invariance guarantees (Burda et al., 2018; Tang et al., 2017; Badnava et al., 2023).

Together, these limitations motivate a method that constructs its shaping potential from quantities already available to the learner, avoiding auxiliary reward models and separately specified potentials. TAP realizes this idea within the PBRS framework by directly using the critic's target value as the shaping potential.

Contributions. 1) We introduce Target Value as Potential (TAP), a reward-shaping method that constructs shaping potentials directly from critic-derived target values. TAP can be readily integrated into standard off-policy actor-critic pipelines without introducing additional shaping-specific hyperparameters. 2) We develop three realizations of TAP: TAP-Self, TAP-Same, and TAP-Cross, corresponding to online self-shaping, same-task transfer, and cross-task transfer, respectively. We characterize their policy-invariance properties: the standard PBRS guarantee applies directly to the fixed potentials used by TAP-Same and TAP-Cross and conditionally to each fixed snapshot of TAP-Self's evolving target critic. 3) Empirically, TAP accelerates convergence and achieves higher cumulative returns than standard DRL baselines, including TD3 and D4PG. Across the evaluated benchmarks, TAP performs favorably against DPBRS and HuRL and remains effective in both same-task and cross-task transfer settings. Preliminary experiments combining TAP with MR.Q also yield improvements in both evaluated pixel-based visual-control environments, demonstrating its applicability to high-dimensional learned representations.

2 Related Work

Knowledge Transfer in RL aims to improve sample efficiency and learning performance on new tasks by leveraging prior experience. Existing approaches transfer source policies , value functions, learned representations, or environment models, demonstrating the diverse forms of knowledge that can facilitate learning across tasks (Taylor & Stone, 2009; Lazaric, 2012). These methods often require strong assumptions about task similarity or access to source-task information. Furthermore, theoretical analysis and empirical results show that learning performance depends critically on the representation of prior knowledge and that priors constructed using existing methods may not reliably yield improvements (Zhang et al., 2022).

Existing transfer methods can be broadly grouped into three categories. **(1) Value initialization and representation transfer.** MAXQINIT (Abel et al., 2018) accelerates learning through value-function initialization with theoretical guarantees, but is primarily effective in relatively simple lifelong RL settings. Successor Features (Barreto et al., 2017) enable transfer through generalized policy improvement by decoupling rewards from dynamics, but assume consistent dynamics across tasks and rely on suitable feature representations. **(2) Policy reuse and mixture-based transfer.** Contextual policy transfer (Gimelfarb et al., 2021) selects among source experts using a Bayesian mixture model, improving sample efficiency but requiring relevant source policies and accurate inference in complex environments. Similarly, the Policy Teaching Framework (PTF) (Yang et al., 2020) formulates transfer as an option learning problem that dynamically selects source policies, but assumes the availability of useful source policies and lacks a principled mechanism to quantify their relevance. **(3) Latent variable and inference-based transfer.** Single Episode Policy Transfer (Yang et al., 2019) infers latent environment variables from limited interaction to enable rapid adaptation, but depends on strong assumptions such as early detectability of dynamics and access to well-designed probe policies or simulators. Taken together, existing transfer methods often rely on strong assumptions, such as access to source policies, accurate environment models, or consistent task structures, which can limit their applicability in complex, high-dimensional, and non-stationary environments. In contrast, TAP

transfers prior knowledge through critic-derived shaping potentials, eliminating the need to explicitly reuse source policies, dynamics models, or learned reward functions.

Potential-Based Reward Shaping (PBRS). PBRS is a principled form of reward shaping that preserves policy invariance when the shaping reward is defined as a potential difference (Ng et al., 1999; Brys et al., 2015). Specifically, the shaping term is given by $F(s_k, s_{k+1}) = \gamma\Phi(s_{k+1}) - \Phi(s_k)$, where Φ is a potential function that encodes task-relevant structure. In practice, **specifying an effective potential** function is challenging. Many approaches construct potentials using domain knowledge, for example by solving simplified or relaxed versions of the original problem or by leveraging planning heuristics (Hoffmann & Nebel, 2001; Richter & Westphal, 2010; Hu et al., 2020). Extensions such as Dynamic PBRS (DPBRS) (Devlin & Kudenko, 2012; Badnava et al., 2023) derive time-varying potentials from learning progress or episode-level performance. While these methods often rely on domain-specific heuristics, auxiliary reward models, or separately learned potentials, TAP instead uses the target critic's value estimate directly as the shaping potential, avoiding separate reward modeling and reducing task-specific design.

Learned Reward Shaping. Because predefined potential functions are often difficult to specify, recent approaches instead learn auxiliary reward functions directly from data, treating them as representations of prior knowledge (Grzes & Kudenko, 2008; Harutyunyan et al., 2015; Sorg et al., 2010). Unlike PBRS-constrained shaping methods, these approaches typically optimize unconstrained auxiliary rewards learned directly from data. For example, Self-Tuning Networks (Stadie et al., 2020) adapt intrinsic reward parameters during training, while ROSA (Mguni et al., 2023) formulates reward shaping as a game to address sparse or uninformative rewards. ReLara (Ma et al., 2024) learns reward models directly from state transitions without explicit domain knowledge or human intervention. Although these methods are shown to improve learning efficiency, learned auxiliary rewards generally lack the policy invariance guarantees provided by classical PBRS formulations, and and may transfer unreliably when environment dynamics or state representations change. In contrast, TAP preserves the structured formulation of PBRS while avoiding separately learned reward models.

3 METHOD

Background. We consider a reinforcement learning agent that interacts with its environment in discrete time. At each time step k, the agent observes a state $s_k \in \mathbf{S}$, selects an action $a_k \in \mathbf{A}$ based on its policy $\pi : \mathbf{S} \to \mathbf{A}$, namely, $a_k = \pi(s_k)$, and receives a scalar reward $r(s_k, a_k) \in \mathbf{R}$ (use r_k as short hand notation).

Evaluation of a policy π is performed using the expected return obtained after taking an action a_k in state s_k and subsequently following the policy π:

$$Q^\pi(s_k, a_k) = \mathbb{E}[R_k \mid s_k, a_k] \tag{1}$$

where $R_k = \sum_{t=k}^{\infty} \gamma^{t-k} r_t$ with $0 < \gamma < 1$. $s_{k+1} \sim p(\cdot \mid s_k, a_k)$, and $a_t = \pi(s_t)$ for $t > k$. For actor-critic methods (Lillicrap et al., 2015; Fujimoto et al., 2018; Haarnoja et al., 2018), the policy π_ϕ is represented by a policy network (the actor) with parameters ϕ, and the state-action value function Q_θ is represented by a critic network with parameters θ. The corresponding target networks for the actor and critic are denoted by $\pi_{\phi'}$ and $Q_{\theta'}$, respectively.

Most actor-critic methods are based on temporal difference (TD) learning (Sutton & Barto, 2018; Si et al., 2004), which updates Q-value estimates by minimizing the TD error. The TD error is defined as the difference between a target value and the current critic estimate. For function approximation and learning, we consider samples drawn from the replay buffer, denoted as $(s, a) \sim \mathcal{D}$, and omit explicit time indices for notational simplicity. The target value y is given by:

$$y = r + \gamma Q_{\theta'}(s', \pi_{\phi'}(s')). \tag{2}$$

Thus, the critic Q_θ is updated by minimizing the loss function $L(\theta)$ with respect to the critic parameters θ:

$$L(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[(y - Q_\theta(s, a))^2\big], \tag{3}$$

where $\mathcal{D}$ denotes the replay buffer from which state-action pairs are sampled.

The actor parameters ϕ can be updated using deterministic policy gradient methods, with the gradient given by $\nabla_\phi J(\phi)$ (Silver et al., 2014):

$$\nabla_\phi J(\phi) = \mathbb{E}_{s\sim d_\mathcal{D}}\left[\nabla_a Q_\theta(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)\right]. \tag{4}$$

The TAP method. We propose to use the target critic network parameterized by θ' as the potential function. Let the target critic be $Q_{\theta'}$ and the target actor (policy) be $\pi_{\phi'}$. We formulate the potential function as

$$\Phi(s_k) = Q_{\theta'}(s_k, \pi_{\phi'}(s_k)). \tag{5}$$

Accordingly, our shaping reward signal $\bar{r}$ as in potential-based reward shaping (PBRS) becomes

$$\bar{r}_k = \gamma Q_{\theta'}(s_{k+1}, \pi_{\phi'}(s_{k+1})) - Q_{\theta'}(s_k, \pi_{\phi'}(s_k)). \tag{6}$$

With the original reward signal r_k and the auxiliary shaping signal $\bar{r}_k$, the shaped reward r'_k is

$$r'_k = r_k + \bar{r}_k. \tag{7}$$

Note, one benefit of TAP is requiring no additional hyperparameter, such as scaling coefficient, clipping, or decay schedule, because the potential is a value function learned under the same reward function as the task which the shaping signal matches the task's return scale.

Then the knowledge-shaped value function $\mathbb{Q}^\pi(s_k, a_k)$ for all state-action pairs and for policy π is

$$\mathbb{Q}^\pi(s_k, a_k) = \mathbb{E}[r'_k + \sum_{j=1}^{\infty} \gamma^j r'_{k+j}]. \tag{8}$$

From Equations (8) and (6) we have the following relationship between shaped Q value and the Q value without shaping as:

$$\mathbb{Q}^\pi(s_k, a_k) + \Phi(s_k) = Q^\pi(s_k, a_k). \tag{9}$$

Henceforth, the knowledge-shaped value function can be updated following the general policy gradient procedure as described in Equations (3) and (4).

Remark 1. TAP provides a unified reward-shaping formulation across all three realizations (TAP-Self, Same, and Cross in Section 5.1). In each case, the potential is derived from a critic using Equation (5), and the shaping signal follows the same PBRS formulation from Equation (6). They differ only in the source of the critic: the current run, a separate run on the same task, or a simpler source task. Consequently, the analysis in Section 4 below applies uniformly to all three realizations. As such, TAP requires neither domain-specific heuristics nor seperarately learned reward models.

4 Theoretical Analysis

We analyze TAP in two stages. Theorem 1 adapts classical PBRS guarantees to TAP's critic-derived potential under exact policy evaluation and improvement, establishing convergence, optimality, and policy invariance in the fixed-potential setting. Building on this foundation, Theorem 2 provides our main theoretical insight by characterizing when and how TAP improves convergence relative to learning without shaping. Proofs and additional discussion are provided in Appendix F.

Theorem 1 (Convergence, optimality, and policy invariance under a fixed potential). Consider TAP with a fixed, bounded potential function Φ under exact policy evaluation and policy improvement. Assume the standard conditions ensuring convergence of exact policy iteration. Let $\{\pi_i\}_{i\geq 0} \subseteq \Pi$ denote the resulting sequence of policies. Then $\{\pi_i\}_{i\geq 0}$ converges to an optimal policy π^* of the TAP-shaped MDP. The shaped and original MDPs have the same set of optimal policies; hence, π^* is also optimal for the original MDP. Moreover, $\mathbb{Q}^{\pi^*} = \mathbb{Q}^*$, where $\mathbb{Q}^*$ denotes the optimal action-value function under the TAP-shaped reward.

Proof. Details are provided in Appendix F. □

Next, we examine how TAP may improve learning efficiency relative to learning without reward shaping. Let π^* denote an optimal policy under the original reward formulation, and let $Q^{\pi^*} = Q^*$ be its corresponding optimal action-value function. The Bellman optimality equation gives

$$Q^*(s_k, \pi^*(s_k)) = \mathbb{E}[r_k + \gamma Q^*(s_{k+1}, \pi^*(s_{k+1})) \mid s_k,\ a_k = \pi^*(s_k)]\,. \tag{10}$$

Theorem 2. Assume exact Bellman updates with discount factor $0 < \gamma < 1$ and bounded one-step rewards, $|r_k| \le r_{\max}$. Let $\mathbb{Q}_k$ and Q_k denote the action-value iterates with and without TAP reward shaping, respectively. Assume that the critic-derived potential $\Phi(s) = Q_{\theta'}(s, \pi_{\phi'}(s))$, abbreviated as $Q_{\theta'}$, remains fixed throughout learning and is informative in the sense that

$$\|Q_{\theta'} - Q^*\|_\infty < \|Q^*\|_\infty\,. \tag{11}$$

Set $Q_0 = \mathbb{Q}_0 = 0$. Let $q \neq Q^*$ be an action-value estimate on the original reward scale such that

$$Q_{n_{ns}} = q \qquad \text{and} \qquad \mathbb{Q}_{n_s} + Q_{\theta'} = q,$$

where $n_{ns} \ge 1$ and $n_s \ge 1$ denote the numbers of Bellman updates required to reach q without and with TAP reward shaping, respectively. Define $\epsilon = \|q - Q^*\|_\infty$. Then the following results hold:

1. $n_{ns} \le \ln\left(\dfrac{\epsilon}{\|Q^*\|_\infty}\right) / \ln(\gamma)$.
2. $n_s \le \ln\left(\dfrac{\epsilon}{\|Q_{\theta'} - Q^*\|_\infty}\right) / \ln(\gamma)$.
3. Let $\bar{n}_{ns}$ and $\bar{n}_s$ be the upper bounds of n_{ns} and n_s, respectively. Then $\bar{n}_{ns} > \bar{n}_s$.

Proof. See Appendix F.

Remark 2. In Theorems 1 and 2 for **TAP-Same** and **TAP-Cross** (refer to Section 5.1), the fixed-potential assumption holds exactly by design as the prior is frozen throughout target-task training. For **TAP-Self**, Polyak averaging ($\tau = 0.005$) causes the target networks, and hence the potential, to evolve slowly, allowing the potential to be treated as approximately fixed in our analysis. For illustration, the target parameters retain a fraction $(1-\tau)^m$ of their previous values after m updates (roughly half after 138 updates).

Remark 3. 1) Theorem 2 suggests that TAP reward shaping can reduce the number of iterations required to reach the same value level compared to learning without shaping. 2) Next, consider a more realistic setting in which the prior knowledge is informative but imperfect. Additional insight can be obtained from the perspective of value decomposition. Specifically, we decompose the optimal action-value function Q^* around the prior target critic estimate $Q_{\theta'}$ as

$$Q^* = \underbrace{Q_{\theta'}}_{\text{(informative prior)}} + \underbrace{\left(Q^* - Q_{\theta'}\right)}_{\text{(residual to be learned)}}\,. \tag{12}$$

From Equation (9), the shaped value function satisfies $\mathbb{Q}^* = Q^* - Q_{\theta'}$. Under this interpretation, TAP can be viewed as learning a **residual** correction relative to a prior critic estimate, rather than learning the full value function from scratch. Consequently, more informative prior critics may reduce the amount of learning required during training.

Remark 4 (Insights from a simple maze problem)**.** To provide empirical intuition for Theorem 2 and Remark 3, we consider a simple maze problem; additional environment details are provided in Appendix A. Using this environment, we examine how the quality of prior target critic estimates affects learning performance. Specifically, prior target and thereafter in Remark 4 Q-tables extracted from different stages of baseline Q-learning are used to simulate varying levels of prior quality. For example, a relatively weak prior may be obtained after 10k training steps, whereas substantially more informative priors may be obtained after longer training horizons such as 250k steps. We use this example to investigate the following questions:
Q1**.** How does the quality of prior critic estimates affect learning speed and cumulative reward?
Q2**.** How does the quality of prior critic estimates influence exploration during learning?

Q1. **Higher-quality priors are associated with accelerated learning and higher success rates.** This evaluation compares standard Q-learning with and without TAP reward shaping. Specifically,

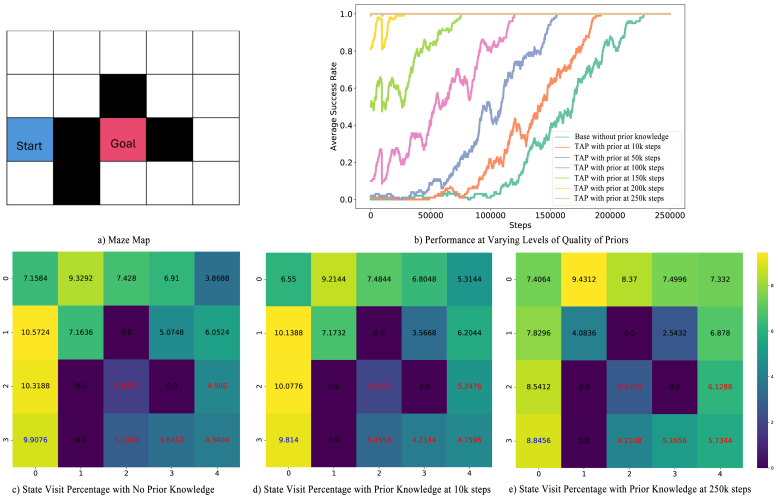

Figure 1: TAP improves learning performance in proportion to the quality of prior critic estimates obtained at different stages of learning the maze-solving task. a) Maze environment, where the blue block denotes the initial state, the red block denotes the terminal state, and the black blocks denote walls. Reaching the terminal state yields reward $r_k = 10$; otherwise $r_k = -1$. b) Learning success rate (averaged over 10 evaluation trials) of Q-learning with TAP under different prior qualities. Prior Q-tables extracted from different stages of baseline training are incorporated into TAP as prior critic estimates. c) Heat map of state visitation frequencies for baseline Q-learning without prior knowledge. The numbers inside the blocks represent visitation percentages over total training steps. d) Same as c), except using TAP with a prior Q-table obtained after 10k baseline training steps. e) Same as d), except using a prior Q-table obtained after 250k baseline training steps.

prior Q-tables obtained at different training stages (sampled every 50k steps) are incorporated into TAP to simulate priors of varying quality. Figure 1b illustrates the effect of prior quality on learning performance. Priors obtained from later training stages consistently lead to substantially faster convergence and higher success rates. In particular, stronger priors obtained after extended training produce near-immediate convergence with success rates approaching 1. These results are consistent with Theorem 2 and empirically support the interpretation in Remark 3 that informative prior critics reduce the amount of learning required during training. Similar trends are also observed in the DRL benchmark results presented in Q3 and Figure 2.

Similar trends hold in a moving-goal variant of the maze, where the goal is relocated every episode: TAP matches or exceeds the unshaped baseline at every prior-quality level (Figure 9, Appendix G). Even the weakest prior (10k steps) does not degrade performance, and gains increase with prior quality.

Q2. TAP biases exploration toward higher-value regions. Further analysis of Figures 1c–e shows that incorporating progressively stronger prior Q-tables reduces visitation frequency in low-value regions near the bottom-left corner of the maze while increasing visitation frequency in regions associated with larger value estimates, such as states near the goal. Compared to learning without shaping, TAP directs exploration toward more promising regions of the state space and reduces unnecessary exploration of low-value states. This effect becomes increasingly pronounced as the prior critic approaches the optimal value function.

5 Experiments and Results

5.1 Experimental Setup and Evaluation Overview

We evaluate TAP on 20 tasks across eight DMC domains, covering both **same-task** and **cross-task** transfer from easier to more challenging tasks. To assess robustness beyond fixed-goal settings, we

include goal-conditioned tasks (*fish swim* and *finger turn*) where targets are resampled each episode, preventing transfer from relying on fixed goal representations. In addition to proprioceptive control tasks, we further evaluate TAP on two pixel-based DMC benchmarks that use stacked 84×84 RGB observations instead of ground-truth states.

For proprioceptive DMC tasks, we augment two DRL baselines, TD3 and D4PG, with TAP and compare against HuRL (Cheng et al., 2021) and DPBRS (Badnava et al., 2023). For pixel-based DMC tasks, we integrate TAP with MR.Q (Fujimoto et al., 2025), a state-of-the-art visual RL method, and evaluate performance on visual control benchmarks. Additional implementation, training, and evaluation details are provided in Appendix A.

In the following, we present systematic results of using TD3 and D4PG results for proprioceptive environments in DMC, and expand TAP to a representation architecture MR.Q for pixel-based visual environments in DMC. The findings for D4PG are given in Appendix C due to space limitation. In reporting evaluation results below, we use the following short-form descriptions.

1) **Base**: the original DRL algorithms (TD3, D4PG).

2) **Base-Visual**: the original visual DRL algorithms, MR.Q.

3) **TAP-Self**: TAP instantiates the potential in Equation (5) with the target critic's value estimate while learning the **same task** within a single training run.

4) **TAP-Same**: TAP instantiates the potential in Equation (5) with the value estimate of a frozen target critic obtained from a separate training run on the **same task**.

5) **TAP-Cross**: TAP instantiates the potential in Equation (5) with the value estimate of a frozen target critic obtained from a simpler source task, thereby enabling **cross-task** transfer.

6) **HuRL**: HuRL-MC is used as it is considered the best performing variant (Cheng et al., 2021).

7) **DPBRS**: Dynamic Potential-Based Reward Shaping method (Badnava et al., 2023).

Our evaluations of TAP aim to quantitatively address the following questions:

Q3. How effectively TAP improves Base DRL methods for learning the same task?
Q4. How effectively TAP improves Base DRL methods for cross-task learning?
Q5. How does TAP encourage exploration?
Q6. Can TAP be extended to representation-based visual DRL methods on the same task?

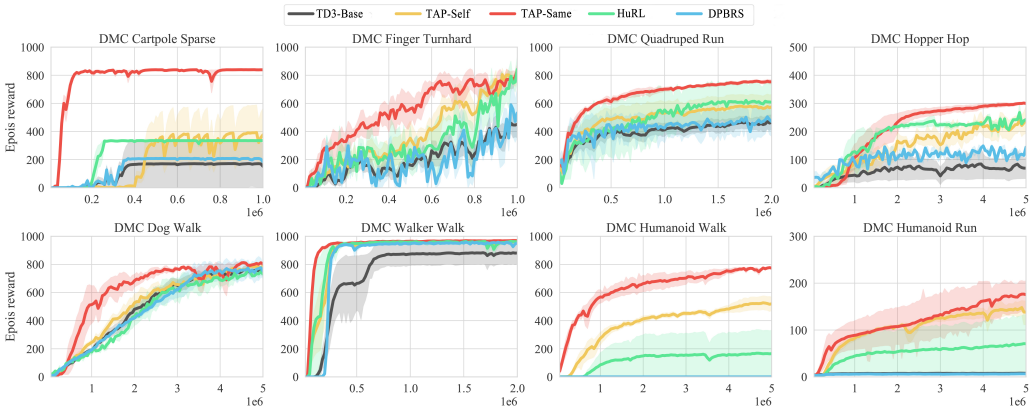

Figure 2: Systematic evaluation of TAP with TD3 as Base for **same-task** transfer in 7 DMC environments. The shaded regions represent the 95 % confidence range of evaluations over 10 seeds. The x-axis is the number of steps. The full set of results evaluated for 20 benchmark tasks in all 8 environments are in Appendix B. Corresponding results of D4PG are in Appendix C.

5.2 Main Results

Q3. TAP boosts Base method performance in learning the same task.

In learning the same task, TAP incorporates prior knowledge through two transfer strategies: **TAP-Self** and **TAP-Same**. Figure 2 presents representative results of TAP in DMC environments. Full benchmark results are provided in Appendix B, with the corresponding D4PG results in Appendix C.

The results show that TAP-Self (yellow curves) consistently improves learning speed, increases final returns, and reduces training variance **compared to the Base method** (black curves). In most environments, TAP-Same (red curves) further improves performance by leveraging prior knowledge transferred from other runs of the same task. The improvements are particularly pronounced in challenging environments such as *Hopper Hop* and *Humanoid Walk*, where the Base method struggles to learn effective policies. In these tasks, TAP achieves substantially higher converged returns than both the Base and HuRL methods. **Compared with HuRL,** TAP-Same achieves consistently stronger performance across the evaluated benchmark tasks. One possible reason is that HuRL depends on heuristics learned from offline datasets through behavior cloning and Monte Carlo regression (Cheng et al., 2021). In sparse-reward or high-dimensional continuous-control settings, such heuristics may generalize poorly due to inconsistencies between trajectories generated by different policies.

TAP-Same also outperforms a **warm-start** baseline that initializes the actor and critic from a source checkpoint (Figure 10, Appendix G), consistent with prior findings that parameter warm-starting can propagate approximation errors (Wang et al., 2023; Wexler et al., 2022). Furthermore, TAP-Self consistently **outperforms DPBRS** across the evaluated environments. Unlike DPBRS, which formulates shaping rewards primarily from episodic reward statistics, TAP-Self constructs the shaping signal using the current target critic and target policy. As a result, the shaping reward reflects state-dependent estimates of expected long-term return rather than coarse aggregated reward information. This distinction becomes increasingly important in continuous-control tasks with complex dynamics, where different states may produce similar immediate rewards while corresponding to substantially different future outcomes.

As training progresses, TAP continuously updates the shaping signal through target network updates, enabling exploration to be guided toward regions associated with higher expected return. This dynamic integration of prior value information improves both learning efficiency and training stability without requiring architectural modifications or additional optimization objectives.

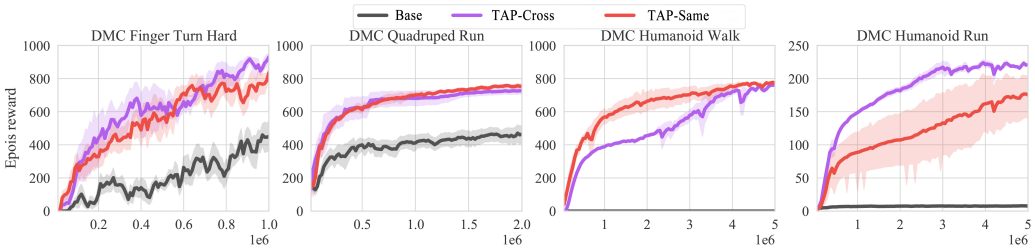

Figure 3: Systematic evaluation of TAP with TD3 as Base in DMC environments for **cross-task** transfer learning. The shaded regions represent the 95 % confidence range of evaluations over 10 seeds. The x-axis is the number of steps. Finger Turn hard environment uses transfer from Finger Turn easy; Quadruped Run from Quadruped walk; Humanoid Walk from Humanoid Stand; and Humanoid Run from Humanoid Walk.

Q4. TAP improves performance on challenging tasks through cross-task transfer from simpler tasks.

Prior work has shown that reward shaping can facilitate transfer across related tasks (Zhu et al., 2023; Marom & Rosman, 2018; Barreto et al., 2017). However, most existing evaluations focus on relatively simple domains such as grid worlds, pinball, or low-dimensional reaching tasks. Transfer in high-dimensional continuous-control environments, particularly **humanoid** locomotion, remains significantly more challenging. **To evaluate cross-task transfer, we consider the following task pairs:** 1) Finger Turn Easy $\rightarrow$ Finger Turn Hard, 2) Quadruped Walk $\rightarrow$ Quadruped Run, 3) Humanoid Stand $\rightarrow$ Humanoid Walk, and 4) Humanoid Walk $\rightarrow$ Humanoid Run.

Figure 3 presents the corresponding results. Across all four transfer settings, **TAP-Cross** consistently improves the performance of the Base methods by leveraging priors obtained from simpler source tasks. The gains are especially pronounced in the humanoid environments, where exploration and policy optimization are substantially more difficult. In the *Humanoid Walk* and *Humanoid Run* tasks (Figure 3, 3rd and 4th plots), the baseline TD3 agent exhibits little or no meaningful learning progress, whereas TAP-based transfer enables stable performance improvement and substantially higher final returns. Existing **TD3 benchmarks** report limited success on humanoid locomotion tasks without additional architectural enhancements or distributed training (Pardo, 2020; Hoffman et al., 2020). In contrast, TAP substantially improves the performance of standard TD3 in these environments. As shown in Figure 2, TAP-Self already enables successful learning on Humanoid Walk and Humanoid Run, tasks on which the baseline TD3 agent struggles. Building on these learned priors, TAP-Same and TAP-Cross further improve performance in the cross-task setting, as shown in Figure 3.

Notably, these gains are achieved without modifying the underlying TD3 architecture or introducing auxiliary optimization objectives. While advanced methods such as distributed D4PG (Barth-Maron et al., 2018) rely on distributional critics and large-scale parallel data collection, TAP instead improves learning through reward shaping derived from transferable value priors. These results suggest that TAP provides a complementary mechanism for improving sample efficiency and exploration in challenging continuous-control problems.

Q5. TAP encourages exploration toward regions associated with higher expected return.

As Figure 7 in Appendix D shows, With TAP-Self, exploration expands into additional regions while simultaneously exhibiting greater concentration around regions associated with higher rewards. The observed exploration patterns are consistent with the interpretation of TAP as learning a residual correction relative to a prior value estimate (Equation 12). By incorporating prior value information into the shaping reward, TAP reduces the burden of learning the full value function from scratch and instead guides exploration using dynamically updated estimates of expected long-term return. This mechanism encourages exploration toward more promising regions of the state space while reducing unnecessary exploration of regions associated with consistently poor outcomes.

Q6. TAP extends to representation-based visual DRL methods on the same task.

In Appendix E, Figure 8 presents representative results of TAP-Same applied to MR.Q in pixel-based DMC environments. Across the evaluated tasks, TAP-Same consistently improves learning stability and either accelerates convergence or improves final return compared to the Base method. These results indicate that the benefits of TAP are not limited to low-dimensional state-based settings, but also extend to representation-learning methods operating directly on high-dimensional visual observations.

6 Discussion and Conclusion

TAP is a simple reward-shaping transfer framework that can be integrated into existing off-policy RL algorithms with minimal overhead, requiring only a modification to the Q-target update and introducing no additional hyperparameters or auxiliary objectives. Applicable to both **same-task** and **cross-task** transfer, TAP consistently improves learning efficiency, stability, and final performance across a wide range of **continuous-control** benchmarks, including challenging humanoid locomotion tasks where TD3 often fails to learn effectively. Moreover, TAP remains effective in **visual-control** settings, accelerating convergence and improving final returns when combined with MR.Q, demonstrating that its benefits extend beyond low-dimensional state-based control to pixel-based representation learning.

Limitations. Despite its promising empirical and theoretical properties, TAP has several limitations. **First**, it assumes compatible state and action spaces between source and target tasks, restricting transfer across substantially different morphologies or control interfaces. **Second**, TAP-Self relies on target networks available in off-policy methods such as TD3 and D4PG, limiting its direct applicability to standard on-policy algorithms like PPO. Extending TAP to such settings would require alternative formulations of the shaping prior. **Finally**, while our visual experiments demonstrate TAP's feasibility under pixel observations, broader evaluations across diverse visual benchmarks and architectures are needed to fully assess its generality in visual reinforcement learning.

7 AI USE STATEMENT

Generative AI tools were used solely for grammar correction and language polishing of the manuscript. No generative AI tools were used for research design, methodology development, experiments, implementation, data analysis, result interpretation, or any other research-related tasks. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of this work.

REFERENCES

David Abel, Yuu Jinnai, Guo Sophie Yue, George Konidaris, and Michael L Littman. Policy and Value Transfer in Lifelong Reinforcement Learning. In *Int. Conf. Mach. Learn.*, pp. 20–29, 2018.

Jacob Adamczyk, Argenis Arriojas, Stas Tiomkin, and Rahul V Kulkarni. Utilizing prior solutions for reward shaping and composition in entropy-regularized reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6658–6665, 2023.

Babak Badnava, Mona Esmaeili, Nasser Mozayani, and Payman Zarkesh-Ha. A new potential-based reward shaping for reinforcement learning agent. In *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 01–06. IEEE, 2023.

Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/350db081a661525235354dd3e19b8c05-Paper.pdf.

Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Tim Brys, Anna Harutyunyan, Matthew E. Taylor, and Ann Nowé. Policy transfer using reward shaping. In *Proc. Int. Conf. Auton. Agent. Multi Agent. Syst.*, pp. 181–188, Richland, SC, 2015. ISBN 9781450334136.

Olivier Buffet and Jörg Hoffmann. All that glitters is not gold: Using landmarks for reward shaping in fpg. In *ICAPS-10 Workshop on Planning and Scheduling Under Uncertainty*, 2010.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Heuristic-guided reinforcement learning. In *Adv. Neural Inf. Process. Syst.*, volume 34, pp. 13550–13563, 2021.

Rati Devidze, Parameswaran Kamalaruban, and Adish Singla. Exploration-guided reward shaping for reinforcement learning under sparse rewards. *Advances in Neural Information Processing Systems*, 35:5829–5842, 2022.

Sam Devlin and Daniel Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 433–440. IFAAMAS, 2012.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. *arXiv preprint arXiv:2501.16142*, 2025.

Michael Gimelfarb, Scott Sanner, and Chi-Guhn Lee. Contextual policy transfer in reinforcement learning domains via deep mixtures-of-experts. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1787–1797. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/gimelfarb21a.html.

Marek Grzes and Daniel Kudenko. Learning potential for reward shaping in reinforcement learning with tile coding. In *Proceedings AAMAS 2008 Workshop on Adaptive and Learning Agents and Multi-Agent Systems (ALAMAS-ALAg 2008)*, pp. 17–23, 2008.

Marek Grzes and Daniel Kudenko. Learning shaping rewards in model-based reinforcement learning. In *Proc. AAMAS 2009 Workshop on Adaptive Learning Agents*, volume 115, pp. 30. Citeseer, 2009.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Matthew W Hoffman, Bobak Shahriari, John Aslanides, Gabriel Barth-Maron, Nikola Momchev, Danila Sinopalnikov, Piotr Stańczyk, Sabela Ramos, Anton Raichuk, Damien Vincent, et al. Acme: A research framework for distributed reinforcement learning. *arXiv preprint arXiv:2006.00979*, 2020.

Jörg Hoffmann and Bernhard Nebel. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. Learning to utilize shaping rewards: A new approach of reward shaping. *Advances in Neural Information Processing Systems*, 33:15931–15941, 2020.

Sven Koenig and Reid G. Simmons. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *Mach. Learn.*, 22(1-3):227–250, January 1996. ISSN 0885-6125. doi: 10.1007/BF00114729.

Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning: State-of-the-Art*, pp. 143–173. Springer, 2012.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Haozhe Ma, Kuankuan Sima, Thanh Vinh Vo, Di Fu, and Tze-Yun Leong. Reward shaping for reinforcement learning with an assistant reward agent. In *Forty-first international conference on machine learning*, 2024.

Ofir Marom and Benjamin Rosman. Belief reward shaping in reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Augustine Mavor-Parker, Kimberly Young, Caswell Barry, and Lewis Griffin. How to stay curious while avoiding noisy tvs using aleatoric uncertainty estimation. In *International Conference on Machine Learning*, pp. 15220–15240. PMLR, 2022.

David Mguni, Taher Jafferjee, Jianhong Wang, Nicolas Perez-Nieves, Wenbin Song, Feifei Tong, Matthew Taylor, Tianpei Yang, Zipeng Dai, Hui Chen, et al. Learning to shape rewards using a game of two partners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11604–11612, 2023.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations : Theory and application to reward shaping. In *Int. Conf. Mach. Learn.*, pp. 278–287, 1999. ISBN 1558606122. doi: 10.1.1.48.345.

Fabio Pardo. Tonic: A deep reinforcement learning library for fast prototyping and benchmarking. *arXiv preprint arXiv:2011.07537*, 2020.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Silvia Richter and Matthias Westphal. The lama planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.

Jennie Si, Andrew G Barto, Warren B Powell, and Don Wunsch. *Handbook of learning and approximate dynamic programming*. John Wiley & Sons, 2004.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Jonathan Sorg, Richard L. Lewis, and Satinder Singh. Reward design via online gradient ascent. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Bradly Stadie, Lunjun Zhang, and Jimmy Ba. Learning intrinsic rewards as a bi-level optimization problem. In *Conference on Uncertainty in Artificial Intelligence*, pp. 111–120. PMLR, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning : an introduction*. MIT Press, Cambridge, MA, 2 edition, 2018. ISBN 9780262039246.

Csaba Szepesvári. *Algorithms for reinforcement learning*. Springer nature, 2022.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Hang Wang, Sen Lin, and Junshan Zhang. Warm-start actor-critic: From approximation error to sub-optimality gap. In *International Conference on Machine Learning*, pp. 35989–36019. PMLR, 2023.

Benjamin Wexler, Elad Sarafian, and Sarit Kraus. Analyzing and overcoming degradation in warm-start reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4048–4055. IEEE, 2022.

Jiachen Yang, Brenden Petersen, Hongyuan Zha, and Daniel Faissol. Single episode policy transfer in reinforcement learning. *arXiv preprint arXiv:1910.07719*, 2019.

Tianpei Yang, Jianye Hao, Zhaopeng Meng, Zongzhang Zhang, Yujing Hu, Yingfeng Chen, Changjie Fan, Weixun Wang, Wulong Liu, Zhaodong Wang, and Jiajie Peng. Efficient deep reinforcement learning via adaptive policy transfer. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3094–3100. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/428. URL https://doi.org/10.24963/ijcai.2020/428. Main track.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

A IMPLEMENTATION DETAILS

We use PyTorch for all implementations. All results were obtained using our internal server consisting of AMD Ryzen Threadripper 3970X Processor, a desktop with Intel Core i7-9700K processor, and two desktops with Intel Core i9-12900K processor.

Simple Maze environment Setup: start state is the initial state and goal is the terminal state. Black cell is the walls. **Reward Setup:** Only receive reward 10 when reach terminal state as $r_k = 10$ if $s = Goal$ else -1. **Algorithms Detail:** The Q learning used in this problem is from sutton book (Sutton & Barto, 2018). All algorithms and environment codes will be provided after paper get accepted.

DRL Training Procedure.

An episode is initialized by resetting the environment, and terminated at max step $T = 1000$. A trial is a complete training process that contains a series of consecutive episodes. Each trial is run for a maximum of 5×10^6 time steps with evaluations at every 5×10^4 time steps for complex benchmarks e.g. humanoid, hopper, and dog. For medium complexity locomotion task such as quadruped and fish, each trial is run for a maximum of 2×10^6 time steps with evaluations at every 2×10^4 time steps. For simple task such as cartpole and finger, each trial is run for a maximum of 1×10^6 time steps with evaluations at every 1×10^4 time steps. Each task is reported over 10 trials where the environment and the network were initialized by 10 random seeds, $(0 - 9)$ in this study.

For each training trial, to remove the dependency on the initial parameters of a policy, we use a purely exploratory policy for the first 8000 time steps (start timesteps). Afterwards, we use an off-policy exploration strategy, adding Gaussian noise $\mathbb{N}(0, 0.1)$ to each action.

Evaluation Procedure.

Every 5×10^4, 2×10^4 and 1×10^4 time steps training depends on task complexity, we have an evaluation section and each evaluation reports the average reward over 5 evaluation episodes, with no exploration noise and with fixed policy weights. The random seeds for evaluation are different from those in training which each trial, evaluations were performed using seeds $(seeds + 100)$.

Network Structure and optimizer.

TD3.The actor-critic networks in TD3 are implemented by feedforward neural networks with three layers of weights. Each layer has 256 hidden nodes with rectified linear units (ReLU) for both the actor and critic. The input layer of actor has the same dimension as observation state. The output layer of the actor has the same dimension as action requirement with a tanh unit. Critic receives both state and action as input to THE first layer and the output layer of critic has 1 linear unit to produce Q value. Network parameters are updated using Adam optimizer with a learning rate of 10^{-3} for simple control problems. After each time step k, the networks are trained with a mini-batch of a 256 transitions (s, a, r, s').

D4PG. Same with the actor-critic networks in D4PG are implemented by feedforward neural networks with three layers of weights. Each layer has 256 hidden nodes with rectified linear units (ReLU) for both the actor and critic. The input layer of actor has the same dimension as observation state. The output layer of the actor has the same dimension as action requirement with a tanh unit. Critic receives both state and action as input to THE first layer and the output layer of critic has a distribution with hyperparameters for the number of atoms l, and the bounds on the support (V_{min}, V_{max}). Network parameters are updated using Adam optimizer with a learning rate of 10^{-3}. After each time step k, the networks are trained with a mini-batch of 256 transitions (s, a, r, s').

Hyperparameters. To keep comparisons in this work fair, we set all common hyperparameters (network layers, batch size, learning rate, discount factor, number of agents, etc) to be the same for comparison within the same methods and different methods.

For TD3, target policy smoothing is implemented by adding $\epsilon \sim \mathbb{N}(0, 0.2)$ to the actions chosen by the target actor-network, clipped to $(-0.5, 0.5)$, delayed policy updates consist of only updating the actor and target critic network every d iterations, with $d = 2$. While a larger d would result in a larger benefit with respect to accumulating errors, for fair comparison, the critics are only trained once per time step, and training the actor for too few iterations would cripple learning. Both target networks are updated with $\tau = 0.005$.

The TD3 used in this study are based on the paper (Fujimoto et al., 2018) and the code from the authors (https://github.com/sfujim/TD3).

Table 1: TD3 hyper parameters used for DMC benchmark tasks

Hyperparameter TD3	Value
Start timesteps	8000 steps
Evaluation frequency	1e4, 2e4 or 5e4 steps
Max timesteps	1e6, 2e6 or 5e6 steps
Exploration noise	$\mathbb{N}(0, 0.1)$
Policy noise	$\mathbb{N}(0, 0.2)$
Noise clip	± 0.5
Policy update frequency	2
Batch size	256
Buffer size	1e6
γ	0.99
τ	0.005
Number of parallel actor	1
Adam Learning rate	1e-3
regularization factor	0.7

The D4PG used in this study is based on paper (Barth-Maron et al., 2018) and the code is modified from TD3. The hyperparameter is from Table 2.

Table 2: D4PG hyper parameters used for the DMC benchmark tasks

Hyperparameter D4PG	Value
Start timesteps	8000 steps
Evaluation frequency	1e4, 2e4 or 5e4 steps
Max timesteps	1e6, 2e6 or 5e6 steps
Exploration noise	$\mathbb{N}(0, 0.1)$
Noise clip	± 0.5
Batch size	256
Buffer size	1e6
γ	0.99
τ	0.005
Number of parallel actor	1
Adam Learning rate	1e-3
V_{max}	100
V_{min}	0
l	51
regularization factor	0.7

MR.Q. The MR.Q base used in this study extends base methods to high-dimensional pixel observations via a learned encoder that maps stacked RGB frames to a compact latent state. The encoder is trained with auxiliary dynamics, reward, and uses a separate multi-step horizon from the Q-learning update. The MR.Q used in this study is based on paper (Fujimoto et al., 2025) with notable changes to updating of the target networks using a soft update as opposed to a hard update. The network structure is from Figure 4 and The hyperparameters are from Table 3. Note that due to action repeat, 500k time steps correspond to 1M frames in the original environment.

Table 3: MR.Q hyper parameters used for the pixel input DMC benchmark tasks

Hyperparameter MR.Q	Value
Start timesteps	10000 steps
Evaluation frequency	5e3 steps
Evaluation episodes	10
Max timesteps	5e5 steps
Observation	84×84 RGB
Frame stack	3
Action repeat	2
Exploration noise	0.2
Target policy noise	0.2
Noise clip	± 0.3
Batch size	256
Buffer size	1e6
γ	0.99
τ (soft target update)	0.005
Encoder dynamics loss weight	1.0
Encoder reward loss weight	0.1
Encoder done loss weight	0.1
Encoder multi-step horizon	5
Q-learning multi-step horizon	3
LAP α	0.4
LAP min priority	1
Pixel augmentations	Enabled
Encoder z_s	512
Encoder z_{sa}	512
Encoder z_a	256
Encoder hidden dim	512
Encoder activation	ELU
Encoder learning rate	1e-4
Encoder weight decay	1e-4
Value network hidden dim	512
Value network activation	ELU
Value network learning rate	3e-4
Value network weight decay	1e-4
Value network gradient clip	20
Policy network hidden dim	512
Policy network activation	ReLU
Policy network learning rate	3e-4
Policy network weight decay	1e-4
Policy pre-activation weight	1e-5

Algorithm 1 TAP integration with MR.Q. (Fujimoto et al., 2025) Only the Q-target reward is modified; all other components remain unchanged.

1: Sample transition (s_k, a_k, r_k, s_{k+1}) from replay buffer
2: Compute potentials using prior networks (Eq. 5):
$\Phi(s_k) = Q_{\theta'}(s_k, \pi_{\phi'}(s_k))$
$\Phi(s_{k+1}) = Q_{\theta'}(s_{k+1}, \pi_{\phi'}(s_{k+1}))$
3: Form shaping term (Eq. 6): $\bar{r}_k = \gamma\, \Phi(s_{k+1}) - \Phi(s_k)$
4: Form shaped reward (Eq. 7): $r'_k = r_k + \bar{r}_k$
5: Use r'_k in place of r_k when computing the Q-target y_k (Eq. 2)
6: If the base algorithm normalizes the Q-target (e.g., via running reward statistics), compute the normalization using r'_k rather than r_k to maintain a consistent target scale
7: Encoder loss, policy update, and replay priorities continue to use the original environment reward r_k

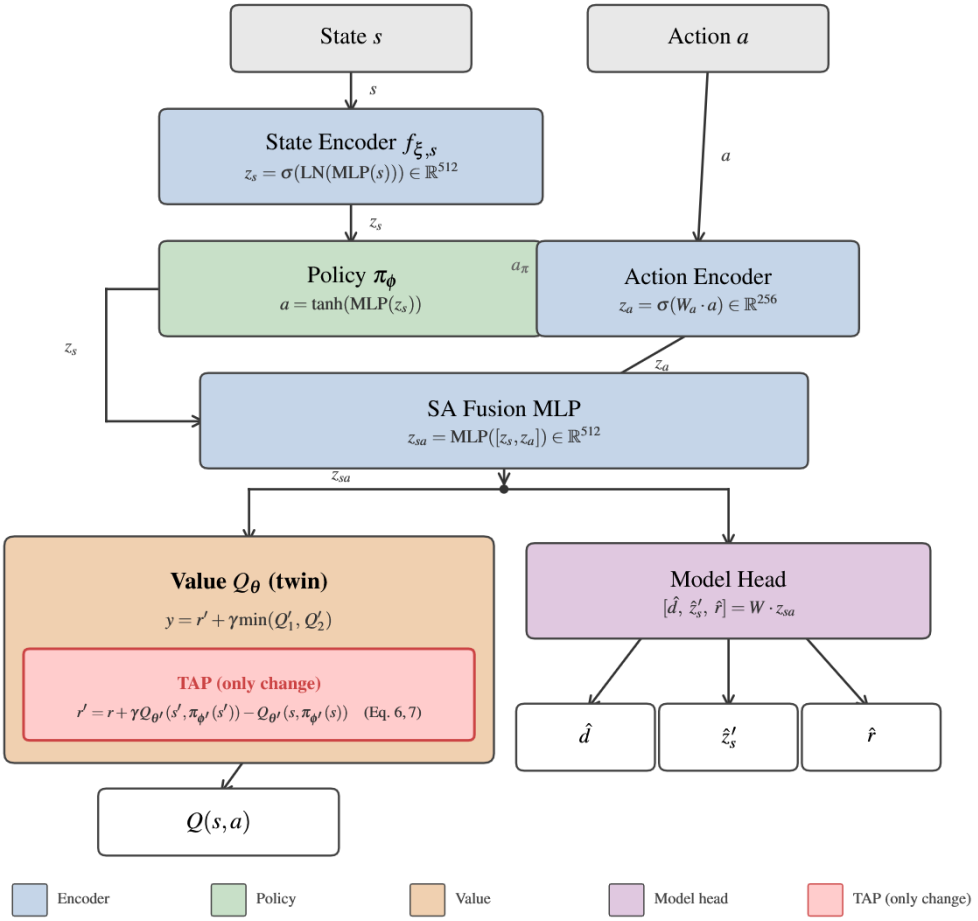

Figure 4: Architecture of MR.Q augmented with TAP. The red block denotes the only modification: the environment reward r is replaced by the shaped reward r' (Eq. 6, 7) when constructing the Q-target. All other components remain unchanged. This highlights that TAP integrates into representation-learning DRL systems with minimal implementation overhead and without architectural modifications.

B FULLSET BENCHMARK RESULTS ON TD3

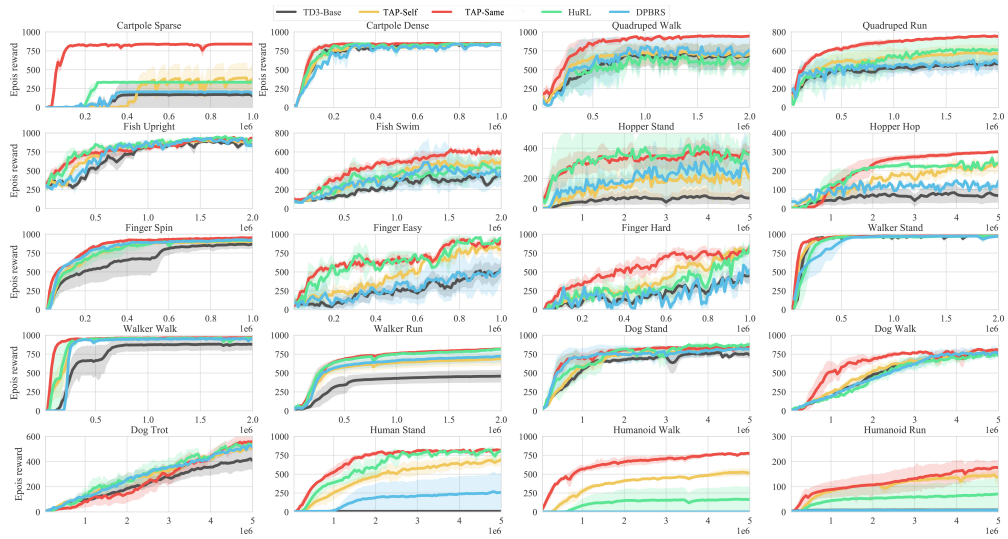

Figure 5: Systematic evaluation of TAP with TD3 base method in DMC environments under **same** task. The shaded regions represent the 95 % confidence range of the evaluations over 10 seeds. The x-axis is the number of steps.

C D4PG BENCHMARK RESULTS

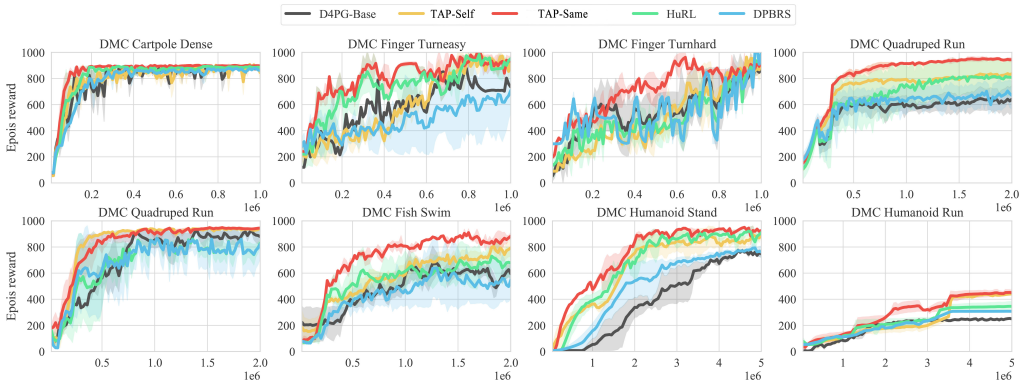

Figure 6: Systematic evaluation of TAP with D4PG base method in DMC environments under **same** task. The shaded regions represent the 95 % confidence range of the evaluations over 10 seeds. The x-axis is the number of steps.

D $Q5$. TAP ENCOURAGES EXPLORATION TOWARD REGIONS ASSOCIATED WITH HIGHER EXPECTED RETURN.

Figure 7 presents a qualitative comparison of state-action-reward samples collected with and without TAP-Self using one million transitions from a simple cartpole environment (Total 2 Millions transition table, first 1M represent with TAP, second 1M represent without TAP) and a more complex humanoid environment (Total 2 Millions transition table). To enable visualization, the state-action representations are projected into two dimensions using t-SNE (Van der Maaten & Hinton, 2008),

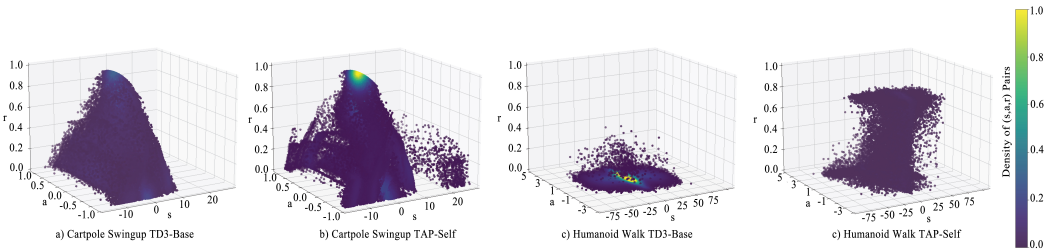

Figure 7: Scatter plot of state-action-reward data to illustrate how TAP enables effective exploration.

implemented through the sklearn package (Pedregosa et al., 2011). Higher-density regions are shown using warmer (yellow) colors.

In the cartpole environment, the baseline TD3 agent explores a relatively concentrated region of the state-action space (Figure 7a). With TAP-Self, exploration expands into additional regions while simultaneously exhibiting greater concentration around regions associated with higher rewards (Figure 7b). In the humanoid environment, the difference becomes substantially more pronounced: the baseline agent remains concentrated in limited low-reward regions (Figure 7c), whereas TAP enables broader exploration and increased visitation of regions associated with higher reward values (Figure 7d).

Although the t-SNE visualization is qualitative, the observed exploration patterns are consistent with the interpretation of TAP as learning a residual correction relative to a prior value estimate (Equation 12). By incorporating prior value information into the shaping reward, TAP reduces the burden of learning the full value function from scratch and instead guides exploration using dynamically updated estimates of expected long-term return. This mechanism encourages exploration toward more promising regions of the state space while reducing unnecessary exploration of regions associated with consistently poor outcomes.

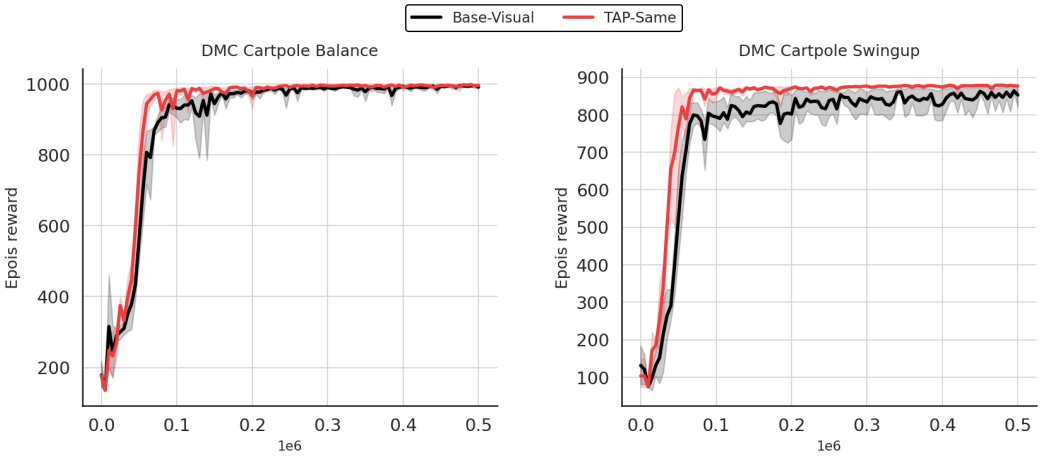

Figure 8: Initial evaluation of TAP with MR.Q as Base in DMC environments for **same-task** transfer learning. The shaded regions represent the 95% confidence interval over 10 seeds. The x-axis denotes the number of environment steps. TAP improves learning stability and achieves faster convergence or higher final return across the evaluated visual-control tasks.

E $Q6$. TAP EXTENDS TO REPRESENTATION-BASED VISUAL DRL METHODS ON THE **SAME TASK**.

To evaluate whether TAP generalizes beyond state-based control, we apply TAP to pixel-based reinforcement learning tasks using MR.Q (Fujimoto et al., 2025) as the base method. This evaluation

examines whether TAP remains effective when the value function and policy are learned from high-dimensional visual observations rather than ground-truth states. The purpose of these experiments is to test whether TAP's construction applies when the critic must learn from high-dimensional visual observations, rather than to claim comprehensive coverage of visual reinforcement learning. We evaluate this by integrating TAP with a recent representation-learning DRL method on two pixel-based DMC tasks.

Figure 8 presents representative results of TAP-Same applied to MR.Q in pixel-based DMC environments. Across the evaluated tasks, TAP-Same consistently improves learning stability and either accelerates convergence or improves final return compared to the Base method. These results indicate that the benefits of TAP are not limited to low-dimensional state-based settings, but also extend to representation-learning methods operating directly on high-dimensional visual observations.

Importantly, refer to Figure 4 in the Appendix A, TAP augments MR.Q without modifying the encoder architecture, representation objectives, policy optimization procedure, or replay mechanism. The prior consists of a frozen encoder, policy, and value network loaded from a previously trained checkpoint. During training, the shaping reward signal is computed by Equation 7 and replace the r when constructing the Q-target as Equation 2. All remaining components of the algorithm remain unchanged. These results show that TAP's construction integrates with learned latent states input in visual RL frameworks such as MR.Q without architectural changes. We report them as initial results rather than a full characterization of TAP in visual reinforcement learning.

F Theoretical Analysis

Lemma 1 (TAP Policy Evaluation). Let r_k, the one-step reward of the target task, be bounded. Given policy π, let the shaped value function $\mathbb{Q}_i$ be defined as in Equation (9). Then the sequence $\{\mathbb{Q}\}_{i\geq 0}$ of shaped value functions converges to $\mathbb{Q}^\pi$ as $i \to \infty$.

Proof. Define the shaping reward r'_k as in Equation (7) and consider the Bellman backup operator as

$$\mathbb{Q}_{i+1}(s_k, a_k) = r'_k + \gamma \mathbb{E}_{s_{k+1}\sim p_\pi, a_{k+1}\sim\pi} \mathbb{Q}_i(s_{k+1}, a_{k+1}). \tag{13}$$

where $s_k \sim p_\pi$ is the state probability induced by the policy π. Since TAP only directly affects the stage reward of the target task, the standard convergence property of policy evaluation (Sutton & Barto, 2018; Haarnoja et al., 2018) leads to the convergence of the sequence $\mathbb{Q}_i$. □

Lemma 2 (TAP Policy Improvement). Let $\pi_i \in \Pi$ be the policy at iteration i. Then the reward shaped value function $\mathbb{Q}^{\pi_i}$ as defined in Equation (9) have that $\mathbb{Q}^{\pi_{i+1}}(s_k, a_k) \geq \mathbb{Q}^{\pi_i}(s_k, a_k)$ for all $(s_k, a_k) \in \mathbf{S} \times \mathbf{A}$.

Proof. The result follows from the standard policy improvement theorem in Section 4.2 of (Sutton & Barto, 2018). □

Proposition 1 (Policy Invariance of TAP). Let the prior knowledge be represented in a potential function $\Phi(s)$ as in Equation (5) and denote $\Phi(s) = Q_{\theta'}(s, \pi'(s))$ which is assumed by be fixed and bounded with respect to policy π. Let $\mathbb{Q}^*$ and Q^* be the optimal value functions with and without reward shaping, respectively. Then the two optimal policies are the same, namely,

$$\pi^* = argmax_{a\in\mathbf{A}} Q^*(s_k, a) \quad = argmax_{a\in\mathbf{A}} \mathbb{Q}^*(s_k, a) \tag{14}$$

Proof. The result is based on (Ng et al., 1999), from Lemma 2, the policy improvement property, and by applying Equation (9),

$$\begin{aligned} \pi^* &= argmax_{a\in\mathbf{A}} \mathbb{Q}^*(s_k, a) \\ &= argmax_{a\in\mathbf{A}}((Q^*(s_k, a) - Q_{\theta'}(s_k, \pi'(s_k))) \\ &= argmax_{a\in\mathbf{A}}(Q^*(s_k, a)) \end{aligned} \tag{15}$$

Proposition 1 thus holds. □

With the above Lemmas, we have the follow convergence and optimality result of TAP.

Theorem 1 (Convergence, optimality, and policy invariance under a fixed potential). Consider TAP with a fixed, bounded potential function Φ under exact policy evaluation and policy improvement. Assume the standard conditions ensuring convergence of exact policy iteration. Let $\{\pi_i\}_{i\geq 0} \subseteq \Pi$ denote the resulting sequence of policies. Then $\{\pi_i\}_{i\geq 0}$ converges to an optimal policy π^* of the TAP-shaped MDP. The shaped and original MDPs have the same set of optimal policies; hence, π^* is also optimal for the original MDP. Moreover, $\mathbb{Q}^{\pi^*} = \mathbb{Q}^*$, where $\mathbb{Q}^*$ denotes the optimal action-value function under the TAP-shaped reward.

Proof. Follow the steps in (Sutton & Barto, 2018; Haarnoja et al., 2018), let π_i be the policy at iteration i. By Lemma 2, for any feasible (s, a), the sequence $\{\mathbb{Q}^{\pi_i}(s, a)\}\}_{i\geq 0}$ is non-decreasing. Also, it is bounded from above by $\mathbb{Q}^{\pi^*}(s, a)$. Hence, it converges pointwise to some limit $\bar{\mathbb{Q}}(s, a) \leq \mathbb{Q}^{\pi^*}(s, a)$. As π_{i+1} is chosen from $\pi_{i+1}(s) \in \arg\max_a \mathbb{Q}^{\pi_i}(s, a)$, any limit policy π^* of $\{\pi_i\}$ satisfies the inequality $\bar{\mathbb{Q}}(s, \pi^*(s)) \geq \bar{\mathbb{Q}}(s, a)$ for all a. That is to say that π^* is greedy w.r.t. $\bar{\mathbb{Q}}$, which is the unique fix point of the Bellman optimality operator, i.e., $\bar{\mathbb{Q}} = \mathbb{Q}^*$ and π^* is optimal. We thus have $\pi_i \to \pi^*$ and $\mathbb{Q}^{\pi_i} \to \mathbb{Q}^*$. □

To establish policy invariance, for any policy π and state-action pair (s, a), the TAP-shaped action-value function satisfies

$$\mathbb{Q}^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \gamma\Phi(s_{t+1}) - \Phi(s_t)\right) \,\middle|\, s_0 = s,\ a_0 = a\right] \tag{16}$$

$$= Q^{\pi}(s, a) - \Phi(s) + \lim_{T\to\infty} \gamma^{T+1}\mathbb{E}_{\pi}[\Phi(s_{T+1}) \,|\, s_0 = s,\ a_0 = a] \tag{17}$$

$$= Q^{\pi}(s, a) - \Phi(s), \tag{18}$$

where the final equality follows because Φ is bounded and $0 < \gamma < 1$. Consequently,

$$\mathbb{Q}^*(s, a) = \max_{\pi\in\Pi} \mathbb{Q}^{\pi}(s, a) = Q^*(s, a) - \Phi(s). \tag{19}$$

Since the offset $-\Phi(s)$ is independent of the action,

$$\arg\max_a \mathbb{Q}^*(s, a) = \arg\max_a Q^*(s, a). \tag{20}$$

Thus, the shaped and original MDPs have the same optimal actions at every state and therefore the same set of optimal policies.

Next, we examine how TAP may improve learning efficiency relative to learning without reward shaping. Let π^* denote an optimal policy under the original reward formulation, and let $Q^{\pi^*} = Q^*$ be its corresponding optimal action-value function. The Bellman optimality equation gives

$$Q^*(s_k, \pi^*(s_k)) = \mathbb{E}[r_k + \gamma Q^*(s_{k+1}, \pi^*(s_{k+1})) \,|\, s_k,\ a_k = \pi^*(s_k)]\,. \tag{21}$$

Theorem 2. Assume exact Bellman updates with discount factor $0 < \gamma < 1$ and bounded one-step rewards, $|r_k| \leq r_{\max}$. Let $\mathbb{Q}_k$ and Q_k denote the action-value iterates with and without TAP reward shaping, respectively. Assume that the critic-derived potential $\Phi(s) = Q_{\theta'}(s, \pi_{\phi'}(s))$, abbreviated as $Q_{\theta'}$, remains fixed throughout learning and is informative in the sense that

$$\|Q_{\theta'} - Q^*\|_\infty < \|Q^*\|_\infty\,. \tag{22}$$

Set $Q_0 = \mathbb{Q}_0 = 0$. Let $q \neq Q^*$ be an action-value estimate on the original reward scale such that

$$Q_{n_{ns}} = q \qquad \text{and} \qquad \mathbb{Q}_{n_s} + Q_{\theta'} = q,$$

where $n_{ns} \geq 1$ and $n_s \geq 1$ denote the numbers of Bellman updates required to reach q without and with TAP reward shaping, respectively. Define $\epsilon = \|q - Q^*\|_\infty$. Then the following results hold:

1. $n_{ns} \leq \ln\left(\dfrac{\epsilon}{\|Q^*\|_\infty}\right) / \ln(\gamma)$.

2. $n_s \leq \ln\left(\frac{\epsilon}{\|Q_{\theta'} - Q^*\|_\infty}\right) / \ln(\gamma)$.
3. Let $\bar{n}_{ns}$ and $\bar{n}_s$ be the upper bounds of n_{ns} and n_s, respectively. Then $\bar{n}_{ns} > \bar{n}_s$.

Proof. Consider that it takes n_{ns} updates for Q value to reach $q = Q_{n_{ns}}$ from Q_0 without shaping. Under the assumption of prior knowledge $Q_{\theta'}$ with fixed values, and from Equation 9, we can write $q = Q_{n_{ns}} = \mathbb{Q}_{n_s} + Q_{\theta'}$. Define the approximation error of q relative to Q^* as $\epsilon = \|q - Q^*\|_\infty$.

Then according to Banach's fixed-point theorem (Szepesvári, 2022), the convergence of the Q value function without using shaping is:

$$\epsilon \leq \gamma^{n_{ns}} \|Q_0 - Q^*\|_\infty. \tag{23}$$

Since $Q_0 = 0$, we can rewrite the inequality as:

$$\frac{\epsilon}{\|0 - Q^*\|_\infty} \leq \gamma^{n_{ns}}. \tag{24}$$

Taking the natural logs of both sides,

$$ln\left(\frac{\epsilon}{\|Q^*\|_\infty}\right) \leq n_{ns} ln(\gamma). \tag{25}$$

As $0 < \gamma < 1$, $ln(\gamma) < 0$, the inequality becomes

$$n_{ns} \leq ln\left(\frac{\epsilon}{\|Q^*\|_\infty}\right) / ln(\gamma). \tag{26}$$

After n_{ns} updates, $\epsilon < \|Q_0 - Q^*\|_\infty$, from which we have that $ln\left(\frac{\epsilon}{\|Q^*\|_\infty}\right) < 0$. Therefore, $0 \leq n_{ns} \leq ln(\frac{\epsilon}{\|Q^*\|_\infty}) / ln(\gamma)$.

With shaping we have the relationship between $\mathbb{Q}$ and Q as described in Equation 9. Therefore, $\epsilon \leq \gamma^{n_s} \|Q_{\theta'} - Q^*\|_\infty$, and n_s obeys the following inequality,

$$n_s \leq ln\left(\frac{\epsilon}{\|Q_{\theta'} - Q^*\|_\infty}\right) / ln(\gamma). \tag{27}$$

Because of $0 < Q_{\theta'} \leq Q^*$, we have

$$\|Q^*\|_\infty > \|Q_{\theta'} - Q^*\|_\infty. \tag{28}$$

As ln is a strictly increasing operator, we have

$$ln\left(\frac{\epsilon}{\|Q^*\|_\infty}\right) / ln(\gamma) > ln\left(\frac{\epsilon}{\|Q_{\theta'} - Q^*\|_\infty}\right) / ln(\gamma). \tag{29}$$

□

While Theorem 2 assumes a fixed potential, this assumption holds exactly for TAP-Same and TAP-Cross, where the prior critic is frozen throughout target-task training. For TAP-Self, the target critic evolves on a much slower timescale than the online critic ($\tau = 0.005$), so the fixed-potential analysis applies as a quasi-static approximation.

G Additional Comparisons

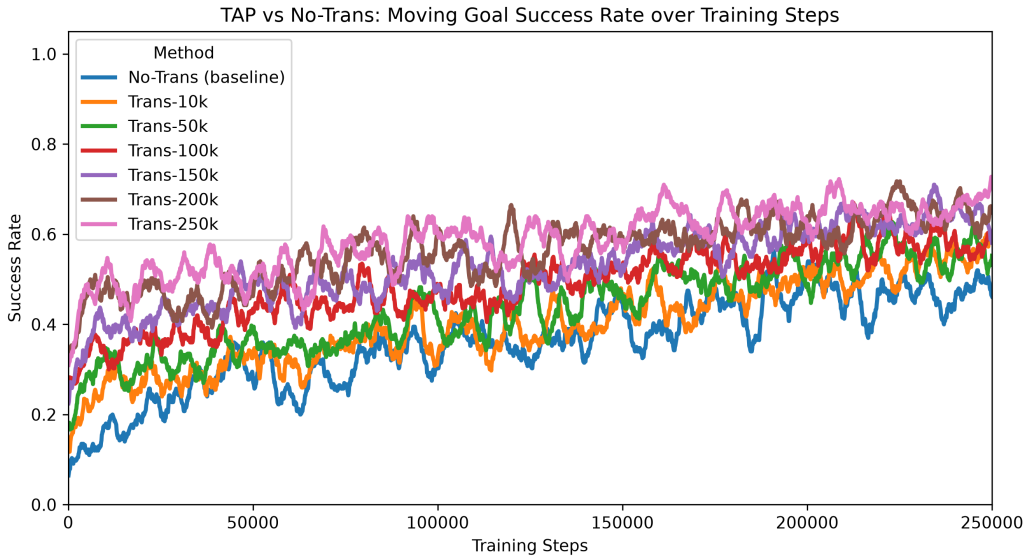

Figure 9: Maze with a moving goal, where the goal is relocated every episode and included in the state. Prior critics trained for 10k–250k steps are transferred via TAP to a fresh learner. TAP matches or exceeds the unshaped baseline at every prior-quality level.

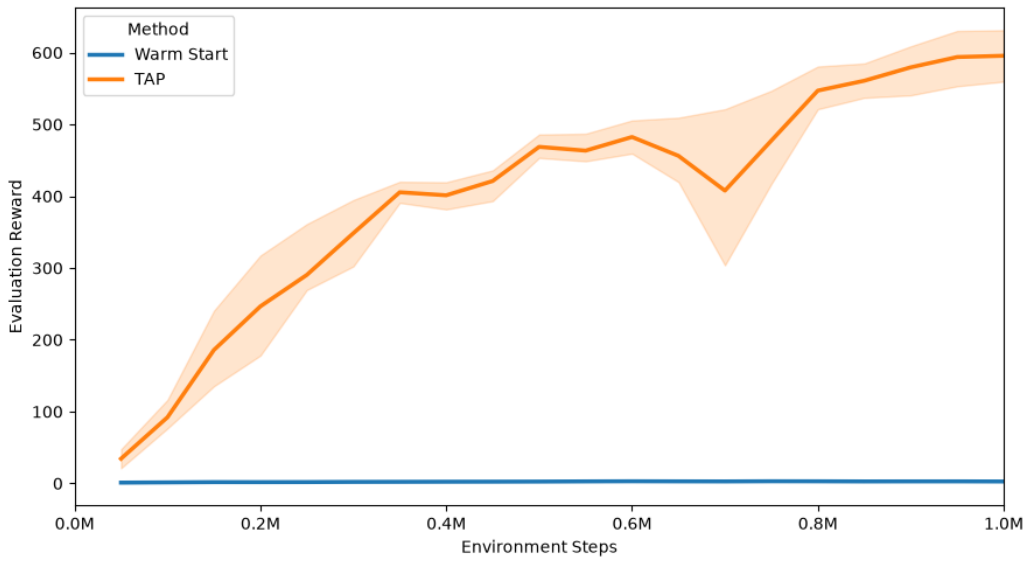

Figure 10: TAP-Same versus warm-start initialization on Humanoid Walk, where the actor and critic are initialized from a 5M-step source checkpoint. Warm-start yields limited improvement, whereas TAP-Same improves both learning speed and final return (Wang et al., 2023; Wexler et al., 2022).