OFF-POLICY RL FRAMEWORK FOR LEARNING FROM DIVERSE TEMPORAL PERSPECTIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning typically trains a policy with a fixed discount factor, assigning exponentially less weight to rewards further in the future. We introduce Temporal Perspective Learning (TPL), an off-policy actor–critic framework that jointly learns policies under different temporal weightings of the same rewards to improve learning for the standard discounted objective. TPL refers to these temporal weightings as temporal perspectives and jointly trains a shared actor and a shared critic conditioned on these perspectives. Unlike prior approaches that decompose a single policy's value across multiple horizons, TPL learns perspective-specific policies and evaluates the consequences of following each policy. This shared architecture allows learning across diverse temporal preferences to influence the policy associated with the standard discounted objective. Experiments on 24 continuous-control tasks across three benchmark suites demonstrate improved performance over the evaluated baselines. Ablations further support the benefits of jointly learning perspective-specific policies and values. The source code is included in the supplementary materials.

1 INTRODUCTION

Reinforcement learning typically trains policies to maximize exponentially discounted returns, assigning lower weights to rewards received further in the future. Although this objective accounts for both immediate and delayed consequences, it aggregates them according to a single temporal weighting. Changing this weighting can alter which actions are preferred, even when the environment and reward function remain unchanged. In this work, we refer to the standard exponentially discounted objective under which we seek to improve policy performance as the *reference objective*. We investigate whether jointly learning policies and value functions under different temporal weightings can improve the performance of the reference objective.

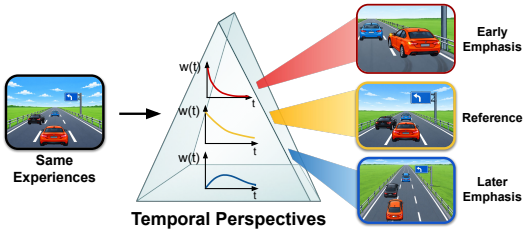

Figure 1: **Conceptual illustration of temporal perspectives.** The same stream of experience is interpreted through diverse temporal perspectives $w(t)$, each highlighting different consequences.

We define a **temporal perspective** as a kernel function for weighting future rewards according to when they are received. Prior work has leveraged multiple time horizons through auxiliary value prediction and value decomposition (Fedus et al., 2019; Romoff et al., 2019; Kalweit et al., 2022). These single-policy approaches learn values by applying different temporal weightings to the future rewards obtained by following the same policy. We instead treat each perspective as an auxiliary *objective* to maximize, jointly training a perspective-conditioned policy and a value function to evaluate the corresponding policy. Each value function therefore reflects the subsequent action choices of the policy associated with its perspective.

Such joint learning may benefit policy learning for the reference objective by enabling the model to learn advantageous actions and their consequences across different temporal perspectives. In the driving example in Figure 1, a perspective that emphasizes near-future rewards may favor accelerating

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.

or braking in response to surrounding traffic. By contrast, a perspective that assigns relatively greater weight to rewards later in the future may favor changing lanes in preparation for an upcoming turn. Jointly learning policies and values for these perspectives allows the model to learn both responses to immediate traffic conditions and action choices that account for the route ahead. We expect that learning these related action choices and their consequences within a shared network can help improve policy performance under the reference objective.

To instantiate this idea, we propose **Temporal Perspective Learning (TPL)**, an off-policy actor–critic framework. The actor takes a temporal perspective as input and selects an action according to that perspective. The critic takes the same perspective as input and estimates the value of subsequently following the corresponding policy. The training perspectives include the perspective corresponding to the reference objective, and we call the policy conditioned on this perspective the reference policy. The actor and critic each share parameters across perspectives, so updates under different perspectives also affect the reference policy and its value estimates. We consistently use the reference policy for both environment interaction and evaluation, and use the experience it collects to train all perspectives off-policy. Thus, perspective-specific policies and values can be jointly learned from shared experience without separately executing each perspective-specific policy to collect data.

A practical challenge is to create sufficiently distinct perspectives while preserving a structure that supports temporal-difference (TD) value learning. Under general temporal weighting, a value function may not admit a closed one-step Bellman recursion expressed solely in terms of itself. To address this challenge, we construct a **residual-basis mixture family** based on the residual decomposition of TD(Δ) (Romoff et al., 2019). The basis consists of a short-horizon exponential kernel and differences between adjacent exponential kernels. Varying the mixture weights yields perspectives that emphasize rewards at different future times, while an equal-weight mixture exactly recovers the exponential discounting of the reference objective. For each perspective, all residual value components evaluate the policy associated with that perspective and are learned through coupled one-step TD updates. We recombine these learned components using perspective-specific weights for actor updates, thereby linking value evaluation under diverse temporal perspectives to policy learning.

We evaluate TPL on 24 continuous-control tasks from AdroitHand, DMControl-Hard, and Humanoid-Bench. Our experiments show improvements in reference policy performance over existing temporal value decomposition baselines and recent actor–critic methods. Ablation studies on a diagnostic task set examine the effects of policy and critic conditioning, the construction of the temporal perspective family, and the diversity of training perspectives. The results support the benefits of jointly learning perspective-specific policies and values. Integrations with Simba and TD-MPC2 further demonstrate the applicability of TPL to other actor–critic architectures and model-based planning.

The contributions of this paper are as follows:

- **Joint policy–value learning across diverse temporal perspectives.** We propose TPL, which conditions both the actor and critic on the temporal weighting of rewards and jointly learns perspective-specific policies and values to improve reference policy performance.
- **A residual-mixture construction for temporal perspectives.** We construct a family of perspectives with diverse temporal emphases. This structure supports coupled one-step TD updates and exactly recovers the reference objective through an equal-weight mixture.
- **Empirical validation and extensions.** We validate the effectiveness of TPL across three continuous-control benchmarks, analyze the contributions of perspective conditioning and diversity, and demonstrate its applicability to different algorithms through integrations with Simba and TD-MPC2.

2 Related work

2.1 Multi-horizon reinforcement learning

Learning from returns over different temporal horizons has long been used in reinforcement learning to manage the bias–variance trade-off in value and policy-gradient estimation. Multi-step methods (Sutton, 1988; Schulman et al., 2018; Daley et al., 2024) use signals spanning different horizons while retaining a single underlying control objective. Other approaches use multi-horizon predictions as auxiliary learning signals or representations (Fedus et al., 2019; Nafi et al., 2022; Grigsby et al.,

2024). In particular, Fedus et al. (2019) further approximate hyperbolically discounted values by recombining values under multiple exponential discount factors.

A related line of work decomposes value estimates into components associated with different temporal scales or reward contributions. Methods such as Composite Q-Learning (Kalweit et al., 2022), TD(Δ) (Romoff et al., 2019), and Temporal Reward Decomposition (Towers et al., 2024) use such decompositions for purposes including improved learning stability and temporal credit attribution. TPL builds on the residual decomposition underlying TD(Δ), using the resulting components as a basis for recombining diverse perspectives.

TPL uses these temporal objectives for joint policy and value learning, with both the actor and the critic conditioned on temporal weighting. Each perspective pairs a conditioned policy with value components evaluated under that continuation policy. Auxiliary perspectives influence reference learning through shared-parameter updates, while only the reference-conditioned policy collects experience and is used for evaluation. A detailed comparison is provided in Appendix C.

2.2 Objective-conditioned reinforcement learning

TPL is also related to approaches that share learning across multiple objectives or conditions. Horde (Sutton et al., 2011) demonstrated how many predictive and goal-oriented questions can be learned in parallel from a shared stream of experience to construct factored, modular knowledge, while Universal Value Function Approximators (UVFA) (Schaul et al., 2015) condition shared value functions on task or goal descriptions. Related conditioning and parameter-sharing principles have since been used in goal-conditioned RL (Andrychowicz et al., 2017; Eysenbach et al., 2022), hierarchical RL (Nachum et al., 2018; Park et al., 2023), multi-objective RL (Abels et al., 2019; Alegre et al., 2023), and auxiliary learning (Jaderberg et al., 2017; Zheng et al., 2024).

TPL addresses an orthogonal *temporal* axis of conditioning. Rather than varying reward functions to specify external goals, subgoals, or different tasks, TPL conditions the actor and critic on the temporal weighting of future rewards within a fixed reward function. Thus, TPL varies not what is rewarded, but how rewards are weighted across future time, inducing auxiliary control objectives with different temporal emphases.

3 Problem formulation

3.1 Kernel-weighted objective

We consider trajectories $\tau = (s_0, a_0, r_0, s_1, \ldots)$ generated by a Markov decision process (Sutton et al., 1999), with initial state distribution μ. A *temporal perspective* specifies how strongly rewards at different future offsets contribute to the objective, without changing the environment or reward function. We represent it by a nonnegative kernel $w \in \mathcal{W}$ and assume bounded rewards and summable kernels. The corresponding objective and return are

$$J_w(\pi) := \mathbb{E}_{\tau\sim\pi}[G_w(\tau)]\,, \qquad G_w(\tau) := \sum_{t\geq 0} w(t) r_t. \tag{1}$$

The exponential kernel $w_{\mathrm{ref}}(t) = \gamma^t$, with $0 < \gamma < 1$, recovers the standard discounted-return objective and serves as our reference. For any state-action pair (s, a), the corresponding action-value is the expected kernel-weighted return starting from (s, a) and following π thereafter:

$$Q^\pi_w(s, a) := \mathbb{E}_\pi\left[G_w(\tau) \,|\, s_0 = s, a_0 = a\right]. \tag{2}$$

Here, temporal offsets are measured from the state-action pair being evaluated.

To build intuition for the role of the kernel-weighted objective in learning, we express $J_w(\pi)$ in terms of policy-induced transitions. Let P_π denote the transition operator under π and r_π the expected one-step reward vector. Using discrete notation, $[P_\pi]_{s,s'} = \sum_a \pi(a \mid s)p(s' \mid s, a)$ and $[r_\pi]_s = \sum_a \pi(a \mid s)\mathbb{E}[r_t \mid s_t = s, a_t = a]$, the objective can be written as

$$J_w(\pi) = \mu^\top \left(\sum_{t\geq 0} w(t) P^t_\pi\right) r_\pi. \tag{3}$$

Here, P_π^t captures the transition relationships obtained by composing the policy-induced one-step dynamics over t steps. As one-step transitions are composed over increasing numbers of steps, the dominant patterns in the resulting transition dynamics can vary across timescales. Then, the reward and temporal weights $w(t)$ determine which timescales of reward-relevant transitions TPL emphasizes during learning.

Importantly, the policy itself shapes the transition patterns represented by P_π^t. We use contrasting temporal perspectives as an auxiliary control objective to encourage each conditioned policy to reinforce transition patterns favored by its temporal weighting of rewards. Through these policy updates, we aim to make the associated reward-relevant transition signals more salient during policy and value learning.

3.2 Training objective under diverse temporal perspectives

Our aim is to improve learning for the fixed reference objective $J_{w_{\text{ref}}}$ through joint learning under auxiliary temporal criteria. Let $\boldsymbol{\alpha} \sim \mathbb{P}(\boldsymbol{\alpha})$ specify a kernel $w_{\boldsymbol{\alpha}}$, and let $\pi_\theta^{\boldsymbol{\alpha}}$ denote the corresponding conditioned policy, with parameters θ shared across perspectives. We express the joint training objective as

$$J_{\text{TPL}}(\theta) = \mathbb{E}_{\boldsymbol{\alpha}\sim\mathbb{P}}\left[J_{w_{\boldsymbol{\alpha}}}\left(\pi_\theta^{\boldsymbol{\alpha}}\right)\right]. \tag{4}$$

Each term pairs a temporal weighting with its corresponding conditioned policy. Varying $\boldsymbol{\alpha}$ therefore changes the reward weighting and can also change the continuation policy, rather than reweighting returns under a common policy.

TPL instantiates this criterion using a residual-basis family of temporal kernels (Section 4.1) and a perspective-conditioned actor and critic (Section 4.2). The kernel family provides contrasting temporal emphases, includes the reference kernel, and supports tractable off-policy value learning. Each network shares parameters across perspectives, and the critic evaluates the policy associated with its condition.

Let $\boldsymbol{\alpha}_{\text{ref}}$ denote the condition satisfying $w_{\boldsymbol{\alpha}_{\text{ref}}} = w_{\text{ref}}$. Environment interaction and evaluation use only $\pi_\theta^{\boldsymbol{\alpha}_{\text{ref}}}$. All perspectives learn off-policy from a shared replay buffer populated by this policy. Non-reference perspectives are used only for auxiliary updates (Section 4.3); their influence on reference learning therefore comes through shared-parameter updates, not through their own exploration.

4 Method

4.1 Residual-basis mixture family of temporal perspectives

A general kernel-weighted value Q_w^π need not admit a closed one-step Bellman recursion in terms of itself alone. We nevertheless seek a kernel family that provides contrasting temporal emphases beyond exponential discounts, while retaining a tractable structure for off-policy value learning. We therefore construct a basis of residual components that separate incremental contributions between successive discount horizons, and propose a *residual-basis mixture family* that mixes these components using $\boldsymbol{\alpha}$ to vary their relative emphasis.

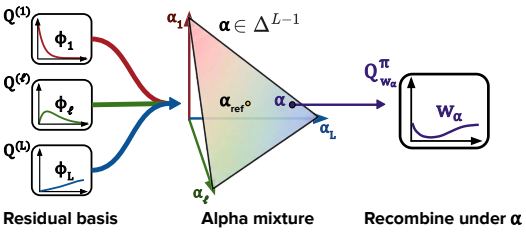

Figure 2: **Residual-basis mixture.** (Left) The critic predicts residual value components associated with successive discount-horizon increments. (Middle) A perspective is represented by $\boldsymbol{\alpha}$ on the simplex, where the centroid $\boldsymbol{\alpha}_{\text{ref}}$ recovers the reference perspective. (Right) The components are recombined to obtain a perspective-specific value.

4.1.1 Residual-basis components

Let $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_L = \gamma < 1$ denote a sequence of discount factors. We define the residual basis functions as

$$\phi_\ell(k) := \gamma_\ell^k - \gamma_{\ell-1}^k, \qquad \ell = 1, \ldots, L, \tag{5}$$

with the convention $\gamma_0^0 := 0$. For a continuation policy π, define the corresponding residual value component as

$$Q^{(\ell),\pi}(s,a) := \mathbb{E}_\pi\Big[\sum_{t\geq 0}\phi_\ell(t)r_t \;\Big|\; s_0 = s, a_0 = a\Big]. \tag{6}$$

$\phi_1(k) = \gamma_1^k$ retains the shortest exponential kernel, while subsequent components capture the incremental contribution between adjacent discount horizons. By smoothly decomposing the reference horizon, these residual components can represent relatively different temporal ranges.

4.1.2 Residual-basis mixture family

We parameterize each temporal perspective by $\boldsymbol{\alpha} \in \Delta^{L-1}$, with $\alpha_\ell \geq 0$ and $\sum_{\ell=1}^{L}\alpha_\ell = 1$. The corresponding temporal weighting function is

$$w_{\boldsymbol{\alpha}}(k) := L\sum_{\ell=1}^{L}\alpha_\ell\phi_\ell(k). \tag{7}$$

Varying $\boldsymbol{\alpha}$ over the simplex continuously changes the relative weights of the residual components and hence the temporal emphasis of the resulting kernel.

For any fixed policy π, linearity gives

$$Q^\pi_{w_{\boldsymbol{\alpha}}}(s,a) = L\sum_{\ell=1}^{L}\alpha_\ell Q^{(\ell),\pi}(s,a). \tag{8}$$

Thus, each perspective value is reconstructed from residual values evaluated under the same continuation policy.

The *reference condition* is the simplex centroid, $\boldsymbol{\alpha}_{\text{ref}} := \frac{1}{L}\mathbf{1}$. (center of Figure 2) Its equal-weight mixture recovers the standard exponential kernel:

$$w_{\boldsymbol{\alpha}_{\text{ref}}}(k) = L\sum_{\ell=1}^{L}\frac{1}{L}\phi_\ell(k) = \gamma^k = w_{\text{ref}}(k). \tag{9}$$

The family thus organizes temporal contrasts around an interior reference: departures from the centroid increase the relative weights of selected residual components while decreasing others.

4.2 Perspective-conditioned actor and critic

Different temporal criteria can favor different policies. TPL therefore conditions both the actor and critic on $\boldsymbol{\alpha}$, which represents $w_{\boldsymbol{\alpha}}$, rather than only evaluating a common policy under different kernels. Each model shares parameters across conditions, allowing auxiliary updates to influence the reference policy and its value estimates.

The actor represents a family of policies

$$\pi_\theta^{\boldsymbol{\alpha}}(a \mid s) := \pi_\theta(a \mid s, \boldsymbol{\alpha}), \tag{10}$$

and the critic predicts all residual component values under the corresponding conditioned policy:

$$\hat{Q}_\psi^{(\ell)}(s,a,\boldsymbol{\alpha}) \approx Q^{(\ell),\pi_\theta^{\boldsymbol{\alpha}}}(s,a), \qquad \ell = 1,\ldots,L. \tag{11}$$

Together, the actor and critic learn to represent changes in action preferences and the consequent values of the conditioned policies as continuous functions of temporal perspective.

The perspective-specific value is estimated by recombining the residual predictions:

$$\hat{Q}^{\pi_\theta^{\boldsymbol{\alpha}}}_{w_{\boldsymbol{\alpha}}}(s,a) := L\sum_{\ell=1}^{L}\alpha_\ell\hat{Q}_\psi^{(\ell)}(s,a,\boldsymbol{\alpha}). \tag{12}$$

Here, $\boldsymbol{\alpha}$ plays two coupled roles. As a conditioning input, it identifies the *continuation policy* $\pi_\theta^{\boldsymbol{\alpha}}$ evaluated by the critic. As a set of recombination weights α_ℓ, it specifies the *temporal kernel* $w_{\boldsymbol{\alpha}}$. All residual components at a given condition $\boldsymbol{\alpha}$ evaluate the same policy $\pi_\theta^{\boldsymbol{\alpha}}$, while their index ℓ specifies the residual basis component.

4.3 PERSPECTIVE-CONDITIONED ACTOR–CRITIC TRAINING AND EVALUATION

At each optimization step, we sample a replay transition $(s, a, r, s') \sim \mathcal{D}$ and a perspective $\boldsymbol{\alpha} \sim \mathbb{P}(\boldsymbol{\alpha})$. Our default distribution is $\mathbb{P} = \mathrm{Dirichlet}(\mathbf{1})$, which is uniform over the simplex.

Actor update. The actor minimizes a perspective-specific surrogate using the recombined value:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{a \sim \pi_\theta^{\boldsymbol{\alpha}}(\cdot|s)} \left[-\hat{Q}_{w_{\boldsymbol{\alpha}}}^{\pi_\theta^{\boldsymbol{\alpha}}}(s, a) \right]. \tag{13}$$

This surrogate guides each conditioned policy toward actions preferred under the temporal weighting $w_{\boldsymbol{\alpha}}$, with their consequences evaluated under its own continuation policy $\pi_\theta^{\boldsymbol{\alpha}}$. However, an exact prediction of $Q_{w_{\boldsymbol{\alpha}}}^{\pi_\theta^{\boldsymbol{\alpha}}}(s, a)$ does not by itself guarantee monotonic policy improvement when the update is used for non-exponential perspectives. Appendix I further discusses the theoretical scope of fixed-policy evaluation and the actor surrogate.

Critic update. For each sampled condition, the critic learns all residual components, regardless of their weights in the actor objective. Given $a' \sim \pi_\theta^{\boldsymbol{\alpha}}(\cdot \mid s')$, we form the residual TD targets Romoff et al. (2019):

$$y_{\boldsymbol{\alpha}}^{(\ell)} = \begin{cases} r + \gamma_1 \hat{Q}_{\bar{\psi}}^{(1)}(s', a', \boldsymbol{\alpha}), & \ell = 1, \\ (\gamma_\ell - \gamma_{\ell-1}) \sum_{i=1}^{\ell-1} \hat{Q}_{\bar{\psi}}^{(i)}(s', a', \boldsymbol{\alpha}) + \gamma_\ell \hat{Q}_{\bar{\psi}}^{(\ell)}(s', a', \boldsymbol{\alpha}), & \ell = 2, \ldots, L. \end{cases} \tag{14}$$

where $\bar{\psi}$ denotes the target critic parameters. The critic minimizes the summed squared TD errors:

$$\mathcal{L}_Q(\psi) = \mathbb{E}\left[\sum_{\ell=1}^{L} \left(\hat{Q}_\psi^{(\ell)}(s, a, \boldsymbol{\alpha}) - y_{\boldsymbol{\alpha}}^{(\ell)} \right)^2 \right]. \tag{15}$$

The same condition $\boldsymbol{\alpha}$ is used in prediction and bootstrapping to evaluate one continuation policy $\pi_\theta^{\boldsymbol{\alpha}}$. By varying the condition, we use the same replay transition to learn component values in a common residual basis for multiple perspective-conditioned policies. This allows the critic to capture the policy-dependent consequences associated with diverse temporal emphases.

Appendix A provides derivations of these TD targets and details the residual-learning design choices, including critic-head separation and an adaptive residual basis designed to balance policy-gradient.

Data collection and evaluation. Only the reference-conditioned policy $\pi_\theta^{\boldsymbol{\alpha}_{\mathrm{ref}}}$ is used for environment interaction and evaluation. Non-reference conditions enter only through auxiliary off-policy updates, excluding direct data collection by their policies.

Appendix F gives the complete training algorithm, and Appendix G provides the entropy-regularized SAC formulation of the updates described above.

5 EXPERIMENTS

We evaluate whether auxiliary temporal control learning improves a fixed reference policy. We then study how this improvement depends on perspective conditioning, α-weighted recombination, and the choice and range of temporal perspective family. Next, we examine whether the learned perspective family induces policy variation and whether the resulting policy differences are associated with instability in off-policy value learning. Finally, we assess sensitivity to residual granularity and applicability to other actor–critic backbones.

Experimental setup. Our primary implementation builds on SAC (Haarnoja et al., 2018), with $L = 4$ unless otherwise specified. We evaluate on AdroitHand (Rajeswaran et al., 2017) (3 tasks), DMControl-Hard (Tunyasuvunakool et al., 2020) (7 tasks), and HBench (Sferrazza et al., 2024) (14 tasks), using continuous actions and state-vector observations. For diagnostic experiments, we use HBench-Periodic (subset of HBench): `h1-hurdle`, `h1-slide`, and `h1-stair`, which involve

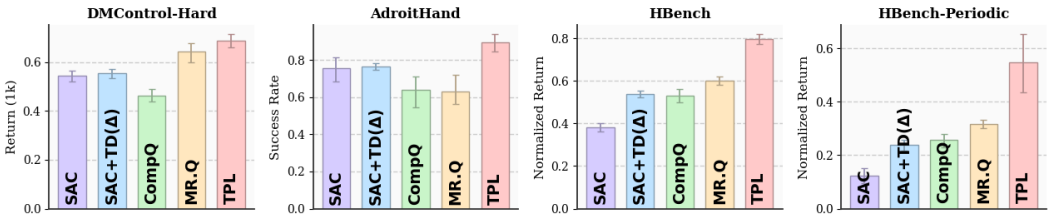

Figure 3: **Empirical performance of TPL and other baselines under diverse continuous control suites.** TPL achieves the highest suite-level mean among the evaluated methods.

Table 1: **Ablations on perspective conditioning and recombination on HBench-Periodic.** All variants use $L = 4$ and the same critic-network structure. O/X indicate the presence/absence of each design choice.

Variant	Residual basis	Actor cond.	Critic cond.	α-recombination	HBench-Periodic
No conditioning	O	X	X	X	0.30 [0.28, 0.31]
No critic conditioning	O	O	X	O	0.41 [0.35, 0.47]
Homogeneous horizon basis	X	O	O	O	0.24 [0.17, 0.30]
Full TPL	O	O	O	O	**0.55** [0.44, 0.66]

repeated obstacle or terrain transitions. All training runs use 1M environment steps and four seeds per task. We report mean normalized scores across tasks within each suite, with 95% stratified-bootstrap confidence intervals from 10,000 resamples. Task details and hyperparameters are provided in Appendices H and E.

5.1 Reference-policy performance

We compare TPL with SAC, SAC+TD(Δ) (Romoff et al., 2019), CompQ (Kalweit et al., 2022), and MR.Q (Fujimoto et al., 2025). SAC provides the base-learner comparison, while SAC+TD(Δ) and CompQ represent temporal value decomposition under a single control objective. MR.Q provides a comparison with an off-policy method using auxiliary model-based representation learning. Baseline details are provided in Appendix D.

Figure 3 reports suite-level performance. TPL achieves the highest suite-level mean across the three suites, showing benefits from auxiliary temporal objective learning in both manipulation and locomotion tasks. We also note strong performance in the HBench-Periodic subset, which we use for further diagnosis. Individual-task learning curves are provided in Appendix L. As TPL introduces extra parameters on its residual components, we additionally report comparisons against SAC and SAC+TD(Δ) baselines with parameter counts matched to TPL in Appendix B.

5.2 Perspective conditioning and α-recombination

We conducted an ablation study to distinguish the contribution of learning *temporal objectives* from residual architecture and training structure. Table 1 summarizes the results on HBench-Periodic.

No conditioning trains only the reference condition while retaining rest of the TPL design choices. **No critic conditioning** retains the conditioned actor but uses a critic that evaluates the reference continuation policy for every condition. Together, they help distinguish the contribution of auxiliary control objectives from gains attributable to the residual critic design. **Homogeneous horizon basis** retains the conditioned networks and sampled recombination weights, but all critic groups estimate the same reference horizon. This removes temporal distinctions among the basis components, serving to isolate effects arising from TPL's training architecture in the absence of temporal objectives.

Full TPL achieves the highest mean performance among these variants. These results indicate that constructing auxiliary control objectives through temporal perspectives—the central idea of TPL—contributes substantially to the observed performance gains.

5.3 PERSPECTIVE FAMILY DESIGN AND DIVERSITY

Comparing perspective families. We first examine how the choice of temporal family affects reference-policy performance. Figure 4(a) compares three constructions on HBench-Periodic. The **exponential-mixture family** combines exponentially discounted components, with the longest-horizon component serving as the reference. The **residual-discrete family** retains the residual basis but trains only on the reference centroid and one-hot perspectives. The **residual-mixture family**, used by TPL, trains on continuous mixtures of residual components.

The residual-mixture family achieves the highest mean score. The residual-discrete family also performs substantially better than the exponential-mixture family, while continuous mixture sampling provides a further improvement. These results support the residual basis as a useful construction for temporal control objectives and show the benefit of learning across mixtures of its components.

Varying perspective diversity. Within the residual-mixture family, we examine how training over narrower or broader ranges of perspectives affects performance. We vary the range of perspectives within the residual-mixture family by linearly contracting its sampling distribution toward the reference centroid:

$$\boldsymbol{\alpha}_\lambda = (1-\lambda)\boldsymbol{\alpha}_{\text{ref}} + \lambda\boldsymbol{\alpha} \tag{16}$$

At $\lambda = 0$, training uses only the reference condition, corresponding to *No conditioning*; $\lambda = 1$ recovers the full sampling distribution of TPL. At each update, the contraction varies the dispersion of temporal kernels while preserving their mean at the reference kernel.

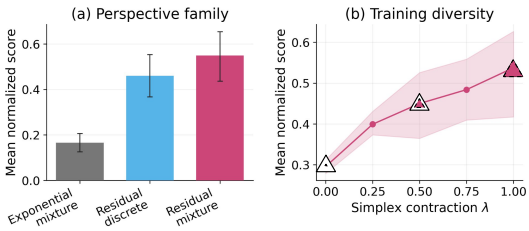

Figure 4: **Perspective families and diversity.** (a) Three perspective families. (b) Contraction toward the reference condition. The colored triangle represents the sampling distribution over the simplex.

Figure 4(b) shows a progressive increase in reference-policy performance as λ grows, with the full simplex achieving the highest mean score. These results suggest that learning across a broader range of temporal perspectives with stronger temporal contrasts benefits reference learning.

5.4 POLICY VARIATION AND OFF-POLICY VALUE ACCURACY

The policy differences induced by auxiliary temporal objectives in TPL may provide richer learning signals, but they may also introduce instability in off-policy value learning as the auxiliary policies do not collect their own interaction data. To quantify this potential trade-off, we analyze how action difference and value-estimation error vary with a perspective's distance from the reference condition.

Using fixed TPL checkpoints on HBench-Periodic, we sample random perspectives $\boldsymbol{\alpha} \sim \text{Dirichlet}(\mathbf{1})$ and group them by normalized distance from the centroid:

$$d(\boldsymbol{\alpha}) = \frac{\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\text{ref}}\|_2}{\sqrt{1 - 1/L}}. \tag{17}$$

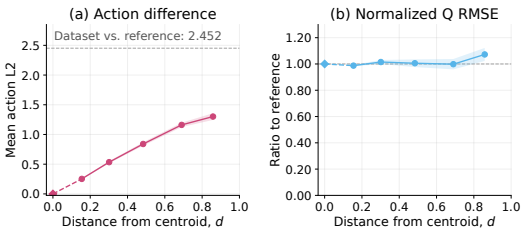

Figure 5: **Policy variation and value diagnostics.** Equal-weight averages over HBench-Periodic. (a) Mean action distance from the reference. (b) Q RMSE against Monte Carlo returns, normalized within each task and seed by the reference error.

Here, $d = 0$ corresponds to the reference condition and $d = 1$ to a simplex vertex. We divide the full distance range $[0.0, 1.0]$ into five equal-width bins and report the mean within each bin, while evaluating the reference condition separately at $d = 0$.

On identical held-out replay states, we measure **action difference**, the L_2 distance between conditioned and reference action means, and **reference-normalized Q error**, the RMSE between predicted Q values and Monte Carlo estimates obtained from rollouts of the conditioned policy, normalized by the error under the reference condition. The detailed procedure is provided in Appendix J.

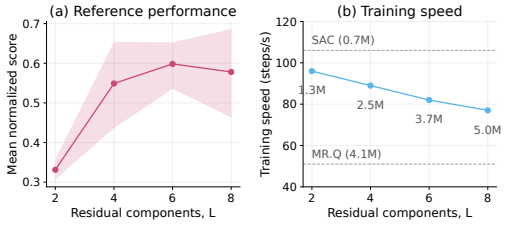

Figure 6: **Varying the number of residual basis components** L. (a) Performance on HBench-Periodic. (b) Training throughput with parameter-count annotations.

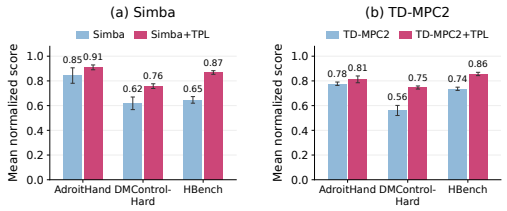

Figure 7: **Full suite +TPL variant results on (a) Simba and (b) TD-MPC2.** The +TPL variants achieve higher suite-level mean performance with both backbones.

Figure 5 shows that average action difference continues to increase with distance from the reference, while the mean MC Q error remains close to the reference-perspective error. Together with the contraction experiment, these results suggest that more distant perspectives induce greater action variation that helps improve reference-policy performance, while remaining stable to preserve the benefits of auxiliary temporal learning.

5.5 Number of residual basis components

We vary the number of residual basis components over $L \in \{2, 4, 6, 8\}$ on HBench-Periodic. Figure 6 shows a clear improvement from $L = 2$ to $L = 4$, followed by smaller changes as the decomposition becomes finer. The highest mean score is obtained with $L = 6$. Increasing L, however, also increases the number of parameters associated with the residual components and leads to a modest increase in training time. We use $L = 4$ as a common default balancing temporal granularity and critic size. The training-speed measurements were obtained using a single NVIDIA GeForce RTX 4090 GPU.

5.6 Generality across backbones

Finally, we evaluate whether the benefits of TPL extend to Simba (Lee et al., 2025) and TD-MPC2 (Hansen et al., 2024) by comparing each backbone with its +TPL variant. The integrations use perspective-conditioned policy and value learning with reference-aligned interaction and evaluation. Backbone-specific implementations and hyperparameters are provided in Appendix D and Appendix E respectively.

Figure 7 shows improved suite-level mean performance with both backbones. The Simba results extend the benefits to a different actor–critic architecture, and the TD-MPC2 results demonstrate applicability to a model-based planning framework. Individual-task learning curves are provided in Appendix L.

6 Discussion

TPL shows that diverse temporal perspectives can support off-policy actor–critic learning, while leaving several directions for future work.

First, only the reference policy collects experience. Sampling perspectives during interaction could enable more diverse exploration, but its benefits must be balanced against sample efficiency for reference-policy learning. Second, actor updates under non-exponential temporal weightings do not generally guarantee policy improvement. Appendix I.3 discusses a possible policy-switching extension, which is not part of the current algorithm. Future work could develop more explicit improvement guarantees by leveraging value comparison or action selection across temporal perspectives. Finally, temporal reweighting alone may provide limited learning signals in sparse-reward environments. Combining TPL with hierarchical policies could help address this limitation by applying temporal perspectives at multiple levels of temporal abstraction.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs, refine hypotheses, assist with translation, and edit the research paper to improve readability. We did not use generative AI tools to provide critical ingredients for proving mathematical claims, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide feedback on research methodology, design or provide feedback on experiments, support qualitative or thematic data analysis, interpret results, or implement methods. Generating synthetic datasets and cleaning or reformatting datasets were not applicable to this work. We have reviewed all AI-assisted work. We manually checked the assumptions, intermediate steps, and logical consistency of AI-assisted mathematical derivations and proofs. Suggestions for refining hypotheses were critically assessed before incorporation. AI-generated or AI-edited text was reviewed and revised for technical accuracy and consistency with the intended claims. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the source code for reproducing all experiments in the supplementary materials. Appendix F presents the complete TPL training algorithm, and Appendix G describes the entropy-regularized implementation used in our experiments. Appendix E reports the hyperparameters and baseline configurations, while Appendix H specifies the benchmark environments and evaluation protocols. Full per-task learning curves are provided in Appendix L. Appendix I states the assumptions and provides proofs for the theoretical results.

REFERENCES

Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 11–20, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

L. N. Alegre, Ana Lúcia Cetertich Bazzan, Diederik M. Roijers, Ann Now'e, and Bruno C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. *ArXiv*, abs/2301.07784, 2023. URL https://api.semanticscholar.org/CorpusID:256000043.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5055–5065, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nIAxjsniDzg.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4058–4068, Red Hook, NY, USA, 2017. Curran Associates Inc. doi: 10.5555/3294996.3295161.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations (ICLR)*, 2021.

Brett Daley, Martha White, and Marlos C. Machado. Averaging n-step returns reduces variance in reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G. Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=R1hIXdST22.

Jake Grigsby, Justin Sasek, Samyak Parajuli, Daniel Adebi, Amy Zhang, and Yuke Zhu. AMAGO-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=OSHaRf4TVU.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1861–1870, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications, 2019. URL https://arxiv.org/abs/1812.05905.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJ6yPD5xg.

Gabriel Kalweit, Maria Kalweit, and Joschka Boedecker. Robust and data-efficient q-learning by composite value-estimation. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=ak6Bds2DcI.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jXLiDKsuDo.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. URL https://api.semanticscholar.org/CorpusID:205242740.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3307–3317, Red Hook, NY, USA, 2018. Curran Associates Inc.

Nasik Muhammad Nafi, Raja Farrukh Ali, and William Hsu. Hyperbolically discounted advantage estimation for generalization in reinforcement learning. In *Decision Awareness in Reinforcement Learning Workshop at ICML 2022*, 2022. URL https://openreview.net/forum?id=UjPGKYtzuuk.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: offline goal-conditioned rl with latent states as actions. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Joshua Romoff, Peter Henderson, Ahmed Touati, Emma Brunskill, Joelle Pineau, and Yann Ollivier. Separating value functions across time-scales, 2019. URL https://arxiv.org/abs/1902.01883.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, Aug 1988. doi: 10.1007/BF00115009. URL https://doi.org/10.1007/BF00115009.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1–2):181–211, August 1999. ISSN 0004-3702. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pp. 761–768, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 0982657161.

Mark Towers, Yali Du, Christopher Freeman, and Timothy J. Norman. Explaining an agent's future beliefs through temporally decomposing future reward estimators, 2024. URL https://arxiv.org/abs/2408.08230.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.

org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning, 2024. URL https://arxiv.org/abs/2306.13229.

A Implementation Details

A.1 Off-policy Bootstrapping for Residual Components

We provide the following derivation for the residual Bellman equation, adopting the one-step TD(Δ).

For the residual components $\mathbf{Q} = \{Q^{(1)}, \dots, Q^{(L)}\}$, the following generalized Bellman equations hold:

$$\begin{aligned} Q^{(1)} &= r + \gamma_1 Q^{(1)\prime}, \\ Q^{(1)} + Q^{(2)} &= r + \gamma_2 \left(Q^{(1)\prime} + Q^{(2)\prime} \right), \\ &\vdots \\ \sum_{i=1}^{L} Q^{(i)} &= r + \gamma_L \sum_{i=1}^{L} Q^{(i)\prime}. \end{aligned}$$

By subtracting the $(\ell-1)$-th equation from the ℓ-th equation, we obtain the update rule for $Q^{(\ell)}$:

$$y^{(\ell)} = Q^{(\ell)} = (\gamma_\ell - \gamma_{\ell-1}) \sum_{i=1}^{\ell-1} Q^{(i)\prime} + \gamma_\ell Q^{(\ell)\prime}. \tag{18}$$

A.2 Design Choices for Learning Residual Components in Off-policy

Here, we introduce the design choices to maximize the temporal expressibility of critics and stabilize the training.

Separating Heads. In contrast to a critic network with linear (single-layer) multi-head outputs, we train *separate* critic head networks of L components, $Q^{(1)}_{\psi^1}, \dots, Q^{(L)}_{\psi^L}$, each dedicated to modeling the residual return R^ℓ. Using separate critic heads reduces interference and allows each critic to specialize in its temporal horizon. Then, the target y^ℓ of a given transition is formed as:

$$y^\ell = \underbrace{(\gamma_\ell - \gamma_{\ell-1}) \sum_{i=1}^{\ell-1} \hat{Q}^{(i)}(s', a')}_{\text{reward term}} + \gamma_\ell \underbrace{Q^{(\ell)}_{\bar{\psi}^\ell}(s', a')}_{\text{bootstrapped term}} \tag{19}$$

where $\bar{\psi}^\ell$ denotes the target network Mnih et al. (2015). The details of the equation 19 and $\hat{Q}^i$ are introduced in the next paragraph.

Accumulated Bias Correction. We use Clipped Double Q (CDQ) Haarnoja et al. (2018); Fujimoto et al. (2018) as the default to compute bootstrapped values, which aggregates the Q value by taking a minimum over two ensembles of networks trained by the same loss.

$$Q^{(\ell)}_{\bar{\psi}^\ell}(s', a') = \min_{j=1,2} (Q^{(\ell)}_{\bar{\psi}^\ell_j}(s', a')). \tag{20}$$

CDQ is known to effectively prevent overestimation at the cost of a small underestimation bias Fujimoto et al. (2018); Chen et al. (2021). However, this small bias can be repeatedly accumulated due to the recursive summation of Qs of the reward term. To mitigate the unintended bias, the Q values are aggregated as:

$$\hat{Q}^{(i)}(s', a') := \underbrace{\min_{j=1,2} Q^{(i)}_{\psi^i_j}}_{\text{CDQ of online}} + \underbrace{\frac{1}{1-\gamma_i} \left(\frac{1}{2} \sum_{j=1,2} Q^{(i)}_{\bar{\psi}^i_j} - \min_{j=1,2} Q^{(i)}_{\bar{\psi}^i_j} \right)}_{\text{estimated CDQ bias of accumulated (avg - min)}}. \tag{21}$$

Here, we omitted the Q inputs. The first term calculates the CDQ of the *online* network ψ^i_j, instead of the target network $\bar{\psi}^i_j$, mitigating the delayed value propagation to the upper-level components.

The second term estimates the *accumulated bias* from the bootstrapped procedure of CDQ. The single-step CDQ bias is calculated by subtracting the min from the average of the predictions. To estimate the infinite discounted sum of the accumulated CDQ bias, we calculate the geometric series sum of the ratio γ_i. The aggregation only corrects the extra accumulation of underestimation caused by the reward term.

A.3 Adaptive Residual Basis

To alleviate the burden of manually tuning the horizon grid $\{\gamma_1, \ldots, \gamma_L\}$, we introduce an auxiliary mechanism that automatically balances the learning signals across all residual components. The core idea is to dynamically adjust the grid boundaries with dual gradient descent Haarnoja et al. (2019) such that the magnitude of the weighted policy gradient for each component ℓ aligns with the global average:

$$\mathbb{E}\left[\|\alpha_\ell \nabla_a \hat{Q}_\psi^{(\ell)}\|\right] \approx \frac{1}{L}\sum_{i=1}^{L} \mathbb{E}\left[\|\alpha_i \nabla_a \hat{Q}_\psi^{(i)}\|\right]. \tag{22}$$

When the learning signal from a specific residual basis becomes strong, its corresponding grid interval narrows, effectively redistributing the learning energy to adjacent, less-informative scales. This behavior minimizes hyperparameter sensitivity and ensures that all temporal horizons contribute meaningfully to the final policy.

For each level, we train the residual horizon ratio h_ℓ. The relationship between the residual horizon ratios and the discount factors for each component follows. Let h'_ℓ be the recovered horizons by the softmax of horizon ratios; then the γ_ℓ can be reconstructed by:

$$h'_\ell := \frac{1}{1-\gamma}\frac{e^{h_\ell}}{\sum_{i=1}^{L} e^{h_i}} \qquad \gamma_\ell = 1 - \frac{1}{\sum_{i=1}^{\ell} h'_i}. \tag{23}$$

Changing the horizon ratio affects the gammas of $\gamma_{\ell<L}$, while the discount factor of the full horizon $\gamma = \gamma_L$ does not change. To train the horizon ratio, we use the following loss term. Let s_ℓ be the expected weighted policy gradient

$$s_\ell \doteq \mathbb{E}\left[\|\alpha_\ell \nabla_a \hat{Q}_\psi^{(\ell)}\|\right], \tag{24}$$

Then the loss for dual gradient descent is

$$\mathcal{L}_\gamma^\ell = h_\ell \cdot (s_\ell - \frac{1}{L}\sum_{i=1}^{L} s_i), \tag{25}$$

which increases the horizon ratio for smaller s_ℓ and decreases the horizon ratio of larger s_ℓ.

We initialize the horizon ratio to match the doubling horizon discount factors as in TD(Δ) (Romoff et al., 2019). The gradient steps for horizons are only done once per $G = 500$ actor-critic updates, waiting for the changed discount factors to affect the critics.

A.4 Ablation Study for Implementation Designs

To further analyze the contribution of bias correction and adaptive residual basis, we evaluate TPL after removing both components. As shown in Table 2, removing these mechanisms has a minor effect on the mean performance on HBench-Periodic. We maintained the designs for consistent plug-and-play across different environments and algorithms, rather than increasing average performance.

Table 2: Ablation study without bias correction and adaptive residual bases.

Algorithms	**HBench-Periodic**
Full TPL	$0.55\ [0.44, 0.66]$
w/o bias correction and adaptive residual bases	$0.54\ [0.40, 0.65]$

A.5 CRITIC NETWORK ARCHITECTURE

Here, we provide sketches of critic structures for TPL, SAC+TD(Δ), and CompQ shown in Figure 8. TPL uses a separate network head for each horizon level (with a shared encoder), whereas TD(Δ) and CompQ use a shared network with linear heads. We also note that we added LayerNorm (Ba et al., 2016) (after activation) and observation normalization (Andrychowicz et al., 2021) for fair comparison throughout the diverse suites of environments.

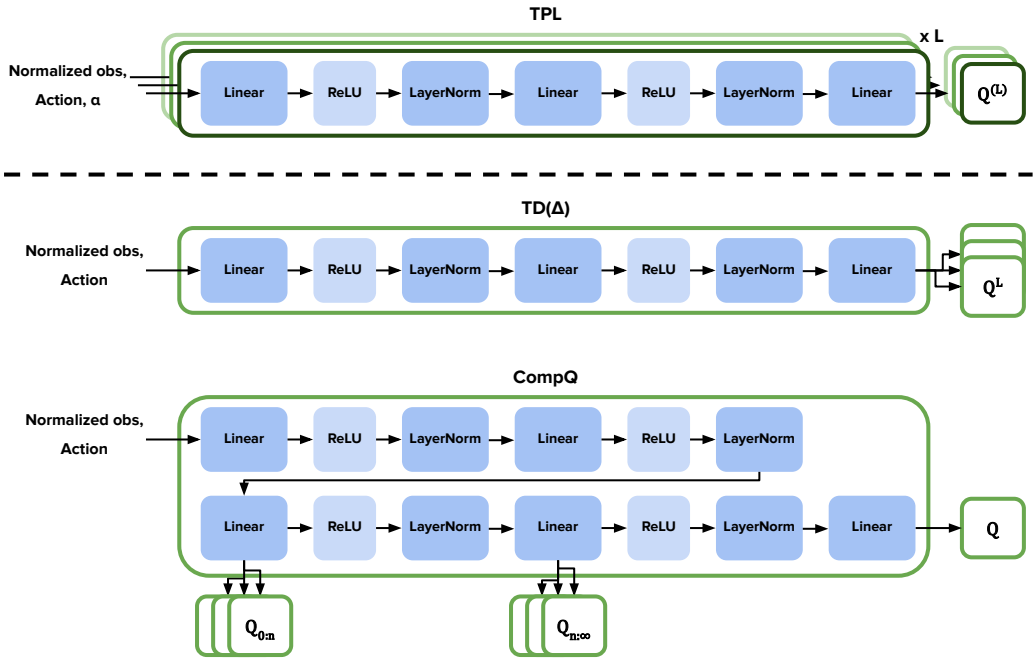

Figure 8: **Critic network structure of TPL, SAC+TD(Δ), and CompQ.**

B Comparisons with Larger Critics

TPL uses separate residual critic networks, which increases its parameter count relative to the default SAC and shared-head TD(Δ) implementations. We compare these methods with runs that increase the baseline critic width from 512 to 1,024 while retaining actor width 256. The larger baselines are close in size to TPL: their actor-plus-critic counts are approximately 2.35M, compared with 2.52M for TPL.

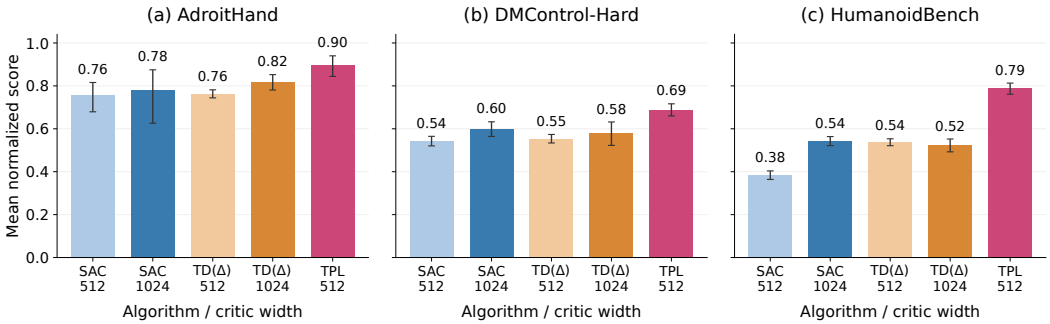

Figure 9: **Comparisons with larger critics.** Bars show equally weighted task means and error bars show 95% task-stratified bootstrap intervals from 10,000 seed resamples. TD(Δ) denotes SAC+TD(Δ); numbers below algorithm names denote critic hidden width. Larger baselines have approximately 2.35M actor-plus-critic parameters, versus 2.52M for TPL.

Results and scope. Increasing SAC critic width improves its mean in each reported suite. TPL retains the highest mean among the selected configurations in each suite. Increasing TD(Δ) width has mixed outcomes across suites. It supports the narrower observation that the wider baselines do not eliminate the mean performance gap to TPL.

C COMPARISON WITH TD(Δ)

Table 3: **Comparison table between SAC variant of TD(Δ) and TPL** on each aspect.

Aspect	TD(Δ)	TPL
Training Objective	optimize a single policy under a single objective	optimize conditioned policies under multiple perspectives
	$J_{\mathrm{ref}}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t\geq 0}\gamma^t r_t\right]$	$J_{\mathrm{TPL}}(\theta) = \mathbb{E}_{\alpha\sim\mathrm{Dir}(\mathbf{1}),\tau\sim\pi_\theta^\alpha}\left[\sum_{t\geq 0} w_\alpha(t) r_t\right]$
Data Collection	single policy under reference objective	single policy under reference objective
	$a_t \sim \pi_\theta(\cdot \mid s_t)$	$a_t \sim \pi_\theta^{\mathrm{ref}}(\cdot \mid s_t)$
Actor	learn a single policy	learn multiple conditioned policies
	$\pi_\theta(\cdot \mid \cdot)$	$\pi_\theta^\alpha = \pi_\theta(\cdot \mid \cdot, \alpha)$
Actor Update	update under the sum of decomposed values with $a \sim \pi_\theta$, $\max_\theta \mathbb{E}\left[\sum_{\ell=1}^{L}\hat{Q}^{(\ell)}(s,a)\right]$	update under distinct mixtures of conditioned components with $\alpha \sim \mathrm{Dir}(\mathbf{1}), a \sim \pi_\theta^\alpha$, $\max_\theta \mathbb{E}\left[L\sum_{\ell=1}^{L}\alpha_\ell \hat{Q}^{(\ell)}(s,a,\alpha)\right]$
Usage of decomposition	Decompose value components for evaluating a single policy	Recombine decomposed value components of conditioned policy to predict $\hat{Q}_{w_{\boldsymbol{\alpha}}}^{\pi_\theta^{\boldsymbol{\alpha}}}(s,a)$
Benefits	Stabilize the value estimation with the bias-variance tradeoff	Exploit multi-scale temporal emphasis with shared actor and critic

D BASELINE ALGORITHMS

TPL. Temporal Perspective Learning is the main algorithm of the paper, which trains the agent by conditioning on diverse temporal perspectives. We implemented our codebase from the open jax SAC implementation of https://github.com/vwxyzjn/cleanrl (Huang et al., 2022).

SAC (Haarnoja et al., 2019). Soft Actor-Critic is an off-policy actor-critic algorithm that maximizes the soft objective of both cumulative reward and entropy. We used the same code base of TPL for evaluating SAC.

SAC+TD(Δ) (Romoff et al., 2019). TD(Δ) is an originally on-policy algorithm that decomposes the full horizon return into residual returns and models each of them. In experiments, we used our implementation of an off-policy version based on SAC that uses one-step TD(Δ) for target calculation, sharing the same code base as TPL.

CompQ (Kalweit et al., 2022). CompQ decomposes the horizon by a truncated and shifted Q function that sums up to the full return. We re-implemented the algorithm in JAX, which can reproduce the official repository:
https://github.com/NrLabFreiburg/composite-q-learning. While the original paper suggests using a separate learning rate for each layer, we found that using layernorms with the same learning rate performed better and reported their scores. Other architectures or hyperparameters are kept the same as the original implementation.

MR.Q (Fujimoto et al., 2025). MR.Q is a model-free actor-critic algorithm with model-based representation learning that aims to address diverse problems with unified settings. For all benchmark suites, we ran 4 seeds with author-suggested hyperparameters from their original repository: https://github.com/facebookresearch/MRQ.

Simba (Lee et al., 2025). Simba proposes an architecture designed for RL, which minimizes the simplicity score of the network for better generalization. For all benchmark suites, we ran 4 seeds of the Simba implementation on our TPL code base. For Simba+TPL, we ran TPL with the Simba architecture by using the default hyperparameters of TPL.

TD-MPC2 (Hansen et al., 2024). TD-MPC2 is a model-based algorithm that trains an implicit world model to perform local trajectory optimization in the latent space. For running the experiments, we reproduced the JAX version of their original repository: https://github.com/nicklashansen/tdmpc2. For TD-MPC2+TPL, we used the same hyperparameters of TD-MPC2 unless noted in Table 5.

E HYPERPARAMETERS

Table 4 shows each hyperparameter of TPL, SAC, SAC+TD(Δ), and CompQ used throughout the environments. The original configurations did not transfer uniformly across all benchmark suites; we therefore use LayerNorm after each activation and observation normalization to obtain a common configuration across environments. Table 5 summarizes the changed hyperparameters for +TPL variants.

Table 4: **Hyperparameter settings for TPL, SAC, SAC+TD(Δ), CompQ.**

Category	Hyperparameter	Value
Common	Discount Factor (γ)	0.99
	Optimizer	AdamW Loshchilov & Hutter (2019)
	Learning rate	3e–4
	Weight decay	1e–4
	Update to Data (UTD) ratio	2
	Critic hidden dim	512
	Actor hidden dim	256
	MLP hidden layers	2
	Activation function	ReLU+LayerNorm Ba et al. (2016)
	Observation Normalization	True
	Clipped Double Q	True
	Target update momentum (τ)	5e–3
	Target update frequency	1
	Min action log std	-5
	Max action log std	2
SAC Common	Use alpha tuning	True
	Target Entropy	$-\lvert\mathcal{A}\rvert/2$
SAC+TD(Δ)	Number of decompositions (L)	4
CompQ	Separate learning rate	False
	Critic hidden dim	500
	Hidden layers	4
TPL	Number of decompositions (L)	4
	Target update momentum (τ)	1.0
	Target update frequency (H)	200
	Adaptive horizon update frequency (G)	500

Table 5: **Hyperparameter settings for +TPL variants.** Unless noted in the table, Simba+TPL uses the default settings of TPL, while TD-MPC2+TPL uses the default settings of TD-MPC2.

Category	Hyperparameter	Value
Simba+TPL	Actor network blocks	2
	Actor network hidden dim	256
TD-MPC2+TPL	Target update momentum (τ)	1.0
	Target update frequency	200

F FULL ALGORITHM OF TPL

Algorithm 1 Temporal Perspective Learning (TPL)

Initialize:
  Actor parameters θ;
  Residual critic parameters $\psi, \bar{\psi}$ (for L components);
  Horizon grid $\Gamma = \{\gamma_\ell\}_{\ell=1}^{L}$; Replay buffer $\mathcal{D}$;
  Reference perspective weights $\alpha_{\text{ref}} = [1/L, \ldots, 1/L]$

while training steps $t < T_{\text{max}}$ **do**
  // Interaction with Reference Perspective
  Sample action $a \sim \pi_\theta(\cdot \mid s, \alpha_{\text{ref}})$
  Take step $s', r \leftarrow \text{env.step}(a)$
  Store transition (s, a, r, s') in $\mathcal{D}$
  $s \leftarrow s'$

  // Off-Policy Training with Diverse Perspectives
  for $k \leftarrow 1$ **to** UTD **do**
    Sample mini-batch $B \sim \mathcal{D}$
    Sample perspective mixture $\alpha \sim \text{Dir}(\mathbf{1})$ ▷ Sample training lens

    // 1. Critic Update (Policy Evaluation)
    Sample next actions $a' \sim \pi_\theta(\cdot \mid s', \alpha)$ ▷ Condition on sampled w_α
    for $\ell \leftarrow 1$ **to** L **do**
      Compute target $y^{(\ell)}$ using $\bar{\psi}, \Gamma$ (equation 14) ▷ Recursive $TD(\Delta)$
    end for
    Update ψ by minimizing $\mathcal{L}_Q = \sum_{\ell=1}^{L} (Q_\psi^{(\ell)}(s, a, \alpha) - y^{(\ell)})^2$

    // 2. Actor Update (Policy Improvement)
    Update θ by minimizing $\mathcal{L}_\pi$ (equation 13)

    // 3. Adaptive Horizon Update (A.3)
    if $t \bmod G = 0$ **then**
      Update grid Γ via dual gradient descent (equation 25)
    end if

    // 4. Target Update
    if $t \bmod H = 0$ **then**
      $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
    end if
  end for
end while

G ENTROPY-REGULARIZED IMPLEMENTATION

The main text describes reward-only values to expose the temporal decomposition. The SAC implementation additionally uses entropy regularization. We denote its temperature by $\eta > 0$ to distinguish it from the perspective weights $\boldsymbol{\alpha}$. In the following derivation, the policy, temperature, and discount grid are fixed, and every value uses the same conditioned continuation policy $\pi^{\boldsymbol{\alpha}}$.

Soft residual values. Define the soft action value with entropy beginning at the next action:

$$Q^{\pi^{\boldsymbol{\alpha}}}_{\gamma,\eta}(s,a) = \mathbb{E}\left[r_0 + \sum_{t=1}^{\infty} \gamma^t \big(r_t - \eta \log \pi^{\boldsymbol{\alpha}}(a_t \mid s_t) \big) \,\middle|\, s_0 = s, a_0 = a \right]. \tag{26}$$

Let $q_1 = Q_{\gamma_1,\eta}$ and $q_\ell = Q_{\gamma_\ell,\eta} - Q_{\gamma_{\ell-1},\eta}$ for $\ell > 1$. Writing $h' = -\eta \log \pi^{\boldsymbol{\alpha}}(a' \mid s')$, subtraction of adjacent soft Bellman equations gives

$$\begin{aligned} y_1 &= r + \gamma_1(q'_1 + h'), \\ y_\ell &= \left[(\gamma_\ell - \gamma_{\ell-1}) \left(\sum_{i<\ell} q'_i + h' \right) + \gamma_\ell q'_\ell \right], \qquad \ell > 1. \end{aligned} \tag{27}$$

Thus entropy contributes with coefficient γ_1 to the first target and $\gamma_\ell - \gamma_{\ell-1}$ to each subsequent target. The sum telescopes to the reference soft target $r + \gamma_L(\sum_\ell q'_\ell + h')$.

Actor and temperature updates. The implemented actor surrogate is

$$\mathcal{L}^{\text{soft}}_\pi = \mathbb{E}_{s\sim\mathcal{D},\, \boldsymbol{\alpha},\, a\sim\pi^{\boldsymbol{\alpha}}_\theta} \left[\eta \log \pi^{\boldsymbol{\alpha}}_\theta(a \mid s) - L \sum_{\ell=1}^{L} \alpha_\ell \overline{q}_{\psi,\ell}(s,a,\boldsymbol{\alpha}) \right], \tag{28}$$

where $\overline{q}$ is the mean across critic ensemble members in the default TPL configuration. A shared scalar temperature is learned using

$$\mathcal{L}_\eta = -\mathbb{E}\left[\log \eta \;\, \mathrm{sg}\big(\log \pi^{\boldsymbol{\alpha}}_\theta(a \mid s) + \mathcal{H}_{\text{target}} \big) \right], \qquad \mathcal{H}_{\text{target}} = -|\mathcal{A}|/2, \tag{29}$$

with sg denoting stop-gradient. Log probabilities are joint action log probabilities, including the squashing correction.

H Environment Details

In this section, three suites of experiments are introduced in detail. Figure 10 shows the visualizations of example tasks in each suite. Table 6 shows the summary of suites. Table 7 shows all included environments of each suite and their state and action spaces.

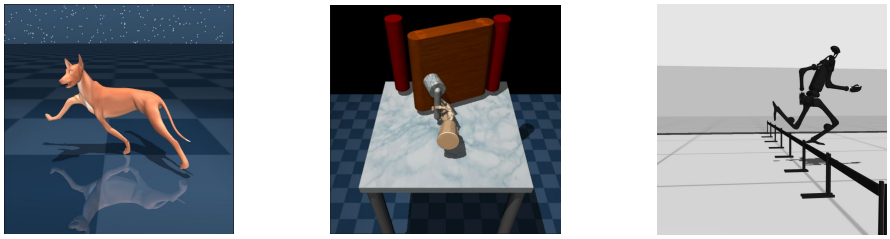

Figure 10: **Visual appearance of each suite.**

Table 6: **Environment details.** We list the episode length, action repeat, total environment steps, and performance metrics for each domain. Note that total interaction steps are equal to the total environment steps/action repeat.

	DMControl-Hard	**AdroitHand**	**HumanoidBench**
Episode length	1,000	400	1,000
Action repeat	2	2	1
Total env. steps	1M	1M	1M
Performance metric	Average Return	Average Success	Average Return

H.1 AdroitHand

We experiment on 3 tasks from the AdroitHand suite (Rajeswaran et al., 2017), involving complex control of a high-dimensional dexterous hand to solve manipulation tasks. The agent is trained with dense reward settings and evaluated by the success rate, where success is determined solely by the state at the end of a truncated episode.

H.2 DeepMind Control

In the DeepMind Control suite, we experiment on 7 tasks of DMControl-Hard as suggested in Simba (Lee et al., 2025), where high-dimensional scenarios of Dog and Humanoid are chosen. In contrast to the HBench suite, the environment is a non-episodic infinite-horizon environment with truncations, and the agent has to first learn to stand up before learning to move. The reported normalized scores are the return divided by the theoretical maximum score of 1k.

H.3 HumanoidBench

In the HumanoidBench suite (Sferrazza et al., 2024), the agent should learn to control a high-dimensional simulated robot to solve a diverse set of tasks, which spans tasks of fundamental locomotion to complex control that involves interaction with different obstacles. Specifically, we experiment on 14 tasks of locomotion without hand variants. The normalization process follows Simba (Lee et al., 2025), where each score is normalized by the pre-defined success score from the original suite description (Sferrazza et al., 2024) and the score of the random policy.

$$\text{HBench-Normalized}(x) := \frac{x \;-\; \text{random score}}{\text{Target success score} \;-\; \text{random score}}$$

Task	Random	Target Success
h1-balance_simple	9.391	800
h1-balance_hard	9.044	800
h1-crawl	272.658	700
h1-hurdle	2.214	700
h1-maze	106.441	1 200
h1-pole	20.090	700
h1-reach	260.302	12 000
h1-run	2.020	700
h1-sit_simple	9.393	750
h1-sit_hard	2.448	750
h1-slide	3.191	700
h1-stair	3.112	700
h1-stand	10.545	800
h1-walk	2.377	700

Table 7: **Environment descriptions of all suites.**

Suite	Environment name	State dim $\|\mathcal{S}\|$	Action dim $\|\mathcal{A}\|$
DMControl-Hard	dog-stand-v0	223	38
	dog-walk-v0	223	38
	dog-trot-v0	223	38
	dog-run-v0	223	38
	humanoid-stand-v0	67	24
	humanoid-walk-v0	67	24
	humanoid-run-v0	67	24
AdroitHand	AdroitHandPen-v1	45	24
	AdroitHandHammer-v1	46	26
	AdroitHandDoor-v1	39	28
HBench	h1-balance_hard-v0	77	19
	h1-balance_simple-v0	64	19
	h1-crawl-v0	51	19
	h1-maze-v0	51	19
	h1-pole-v0	51	19
	h1-reach-v0	57	19
	h1-run-v0	51	19
	h1-sit_simple-v0	51	19
	h1-sit_hard-v0	64	19
	h1-stand-v0	51	19
	h1-walk-v0	51	19
	h1-hurdle-v0	51	19
	h1-slide-v0	51	19
	h1-stair-v0	51	19

I Policy Evaluation and Reference Performance Bounds

We establish fixed-policy evaluation for an arbitrary temporal perspective, show that greedy policy improvement can fail, and derive a reference performance bound via cross-evaluation and policy switching. All returns are reward-only and start from a specified state $S_0 = s$.

I.1 Fixed-policy evaluation for one perspective

Fix an arbitrary perspective $\boldsymbol{\alpha} \in \Delta^{L-1}$, a stationary policy $\pi = \pi^{\boldsymbol{\alpha}}$, and a discount grid $0 \le \gamma_1 < \cdots < \gamma_L < 1$ throughout this subsection. Assume $|R_t| \le R_{\max}$ and define

$$J_{\boldsymbol{\alpha}}(\pi; s) = \mathbb{E}_\pi \left[\sum_{t \ge 0} w_{\boldsymbol{\alpha}}(t) R_t \,\middle|\, S_0 = s \right], \qquad w_{\boldsymbol{\alpha}}(t) = L \sum_{\ell=1}^{L} \alpha_\ell \phi_\ell(t), \tag{30}$$

where $\phi_1(t) = \gamma_1^t$ and $\phi_\ell(t) = \gamma_\ell^t - \gamma_{\ell-1}^t$ for $\ell \ge 2$, with $0^0 = 1$. Define

$$\begin{aligned} Q_\gamma^\pi(s,a) &= \mathbb{E}_\pi \left[\sum_{t \ge 0} \gamma^t R_t \,\middle|\, S_0 = s, A_0 = a \right], \\ q_1^\pi &= Q_{\gamma_1}^\pi, \qquad q_\ell^\pi = Q_{\gamma_\ell}^\pi - Q_{\gamma_{\ell-1}}^\pi \quad (\ell \ge 2). \end{aligned} \tag{31}$$

Here $q_\ell^\pi = Q^{(\ell),\pi}$ in the main text's notation. All components use the same continuation policy π, so the weighted action value satisfies

$$Q_{w_{\boldsymbol{\alpha}}}^\pi = L \sum_\ell \alpha_\ell q_\ell^\pi, \qquad J_{\boldsymbol{\alpha}}(\pi; s) = \mathbb{E}_{a \sim \pi(\cdot|s)} Q_{w_{\boldsymbol{\alpha}}}^\pi(s,a). \tag{32}$$

Let $r(s,a) = \mathbb{E}[R_0 \mid s, a]$ and $(P_\pi f)(s,a) = \mathbb{E}_{S' \sim p(\cdot|s,a),\, A' \sim \pi(\cdot|S')} f(S', A')$. For bounded residual functions $\mathbf{q} = (q_1, \ldots, q_L)$, the expected residual backup follows the TD(Δ) decomposition Romoff et al. (2019):

$$\begin{aligned} (\mathcal{T}_\pi \mathbf{q})_1 &= r + \gamma_1 P_\pi q_1, \\ (\mathcal{T}_\pi \mathbf{q})_\ell &= (\gamma_\ell - \gamma_{\ell-1}) P_\pi \sum_{i<\ell} q_i + \gamma_\ell P_\pi q_\ell, \quad \ell \ge 2. \end{aligned} \tag{33}$$

Theorem I.1 (Fixed-policy residual evaluation)**.** *Define the cumulative norm* $\|\mathbf{q}\|_{\mathrm{C}} = \max_\ell \|\sum_{i \le \ell} q_i\|_\infty$. *Then* $\mathcal{T}_\pi$ *is a* γ_L *contraction and has the unique bounded fixed point* $\mathbf{q}^\pi$ *in equation 31. For any bounded initialization, the exact expected iteration* $\mathbf{q}^{k+1} = \mathcal{T}_\pi \mathbf{q}^k$ *satisfies*

$$\|\mathbf{q}^k - \mathbf{q}^\pi\|_{\mathrm{C}} \le \gamma_L^k \|\mathbf{q}^0 - \mathbf{q}^\pi\|_{\mathrm{C}}. \tag{34}$$

Consequently, $L \sum_\ell \alpha_\ell q_\ell^k$ *converges uniformly to* $Q_{w_{\boldsymbol{\alpha}}}^\pi$*, and its initial-action expectation converges to* $J_{\boldsymbol{\alpha}}(\pi; s)$ *for every starting state* s*.*

Proof. Let $C_\ell = \sum_{i \le \ell} q_i$. This transformation is invertible: $q_1 = C_1$ and $q_\ell = C_\ell - C_{\ell-1}$. Summing the first ℓ components of equation 33 telescopes to

$$\sum_{i \le \ell} (\mathcal{T}_\pi \mathbf{q})_i = r + \gamma_\ell P_\pi C_\ell. \tag{35}$$

Hence the cumulative coordinates follow independent geometric policy-evaluation backups. Since $\|P_\pi f\|_\infty \le \|f\|_\infty$,

$$\|\mathcal{T}_\pi \mathbf{q} - \mathcal{T}_\pi \widetilde{\mathbf{q}}\|_{\mathrm{C}} \le \gamma_L \|\mathbf{q} - \widetilde{\mathbf{q}}\|_{\mathrm{C}}. \tag{36}$$

The cumulative fixed points $C_\ell = Q_{\gamma_\ell}^\pi$ and the inverse transformation identifies the unique residual fixed point equation 31, with rate equation 34. Since every residual is a difference of cumulative values, the fixed linear readout and its initial-action expectation in equation 32 also converge uniformly. □

I.2 Evaluation does not imply policy improvement

From the original starting state, future rewards at time $t+k$ receive weight $w(t+k)$, whereas local action values at time t use $w(k)$. Geometric kernels satisfy $w(t+k) = \gamma^t w(k)$; general residual mixtures need not. Consequently, greedy policy replacement need not improve $J_w(\pi; s)$, even with exact evaluation.

Example I.2 (Exact greedy improvement can fail)**.** *Let $L = 2$, $\boldsymbol{\alpha} = (0, 1)$, $\gamma_1 = 1/2$, and $\gamma_2 = 9/10$, so that $w(t) = 2((9/10)^t - (1/2)^t)$ belongs to the residual family. Consider the following deterministic continuing MDP:*

State	*a_0: reward, next state*	*a_1: reward, next state*
A	$1, A$	$1, B$
B	$2, A$	$0, B$

Let π_0 choose a_0 at both states and π_1 choose a_1 at both states. Their exact weighted action values are, for either $s = A$ or $s = B$,

Continuation	$Q^\pi_w(s, a_0)$	$Q^\pi_w(s, a_1)$	$J_w(\pi; s)$
π_0	16	$84/5$	16
π_1	$4/5$	0	0

To verify the entries, the geometric state values under π_0 are $(10, 11)$ at discount $9/10$ and $(2, 3)$ at discount $1/2$; under π_1 they are $(1, 0)$ at either discount. For a successor state x, the weighted action value is $2[(9/10)V^\pi_{9/10}(x) - (1/2)V^\pi_{1/2}(x)]$, because $w(0) = 0$.

Thus the unique greedy replacement of π_0 is π_1, but $J_w(\pi_1; s) = 0 < 16 = J_w(\pi_0; s)$. The reverse greedy replacement is $\pi_1 \to \pi_0$, yielding a two-cycle. By continuity, these strict inequalities persist for sufficiently small positive first mixture weight.

I.3 Reference evaluation and policy switching

For a fixed policy $\pi^{\boldsymbol{\beta}}$ and discount grid, equation 32 gives

$$Q^{\pi^{\boldsymbol{\beta}}}_{w_{\boldsymbol{\alpha}'}}(s, a) = L \sum_{\ell=1}^{L} \alpha'_\ell q^{\pi^{\boldsymbol{\beta}}}_\ell(s, a), \qquad \boldsymbol{\alpha}' \in \Delta^{L-1}. \tag{37}$$

Every residual retains the continuation policy $\pi^{\boldsymbol{\beta}}$. In particular, the uniform weights $\boldsymbol{\alpha}_{\mathrm{ref}} = \mathbf{1}/L$ recover $w_{\mathrm{ref}}(t) = \gamma_L^t$, so telescoping gives

$$Q^{\pi^{\boldsymbol{\beta}}}_{\mathrm{ref}}(s, a) = \sum_{\ell=1}^{L} q^{\pi^{\boldsymbol{\beta}}}_\ell(s, a) = Q^{\pi^{\boldsymbol{\beta}}}_{\gamma_L}(s, a). \tag{38}$$

Write $J_{\mathrm{ref}} = J_{\boldsymbol{\alpha}_{\mathrm{ref}}}$. Theorem I.1 applies separately to each fixed candidate evaluated under this common reference objective.

Consider a finite collection of stationary randomized policies

$$\mathcal{C} = \{\pi_0, \pi_1, \ldots, \pi_m\}, \qquad \pi_0 = \pi_{\mathrm{ref}},$$

in a finite-state, finite-action MDP with bounded rewards. The candidates and their common discount grid $0 \le \gamma_1 < \cdots < \gamma_L < 1$ are frozen throughout the construction and evaluation below. Define each candidate's reference state value by

$$V_i(s) = \mathbb{E}_{a \sim \pi_i(\cdot|s)} Q^{\pi_i}_{\mathrm{ref}}(s, a) = J_{\mathrm{ref}}(\pi_i; s).$$

Select a maximizing candidate at every state, and execute its action distribution:

$$i^*(s) \in \arg\max_{0 \le i \le m} V_i(s), \qquad \pi_{\mathrm{sw}}(\cdot \mid s) = \pi_{i^*(s)}(\cdot \mid s). \tag{39}$$

This is a GPI-style use of evaluated policies, but selects candidates by their own state values rather than maximizing $\max_i Q^{\pi_i}_{\mathrm{ref}}(s, a)$ over actions as in actionwise GPI (Barreto et al., 2017).

Theorem I.3 (Reference performance bound)**.** *If the candidate reference values are exact, then for every initial state s,*

$$J_{\mathrm{ref}}(\pi_{\mathrm{sw}}; s) \geq \max_{0 \leq i \leq m} J_{\mathrm{ref}}(\pi_i; s) \geq J_{\mathrm{ref}}(\pi_{\mathrm{ref}}; s). \tag{40}$$

Proof. Write $\gamma = \gamma_L$ and $M(s) = \max_i V_i(s)$. The reference Bellman evaluation operator is

$$(T_\pi v)(s) = \mathbb{E}_{a \sim \pi(\cdot|s),\, s' \sim p(\cdot|s,a)} \big[r(s,a) + \gamma v(s') \big].$$

Since $M \geq V_i$ pointwise and $V_i = T_{\pi_i} V_i$, at each state s,

$$(T_{\pi_{\mathrm{sw}}} M)(s) \geq (T_{\pi_{i^*(s)}} V_{i^*(s)})(s) = V_{i^*(s)}(s) = M(s).$$

Monotonicity gives $T^k_{\pi_{\mathrm{sw}}} M \geq M$ for every $k \geq 1$. The operator is a γ-contraction in the supremum norm, so the iterates converge to $V^{\pi_{\mathrm{sw}}}_{\mathrm{ref}}$, proving its dominance over M. The final inequality follows from $\pi_{\mathrm{ref}} \in \mathcal{C}$. □

We do not incorporate policy switching into the main TPL algorithm. Although recombining residual values is inexpensive, evaluating and comparing multiple candidate policies introduces additional computation. Our exploratory tests of explicit bounding did not show a clear performance gain to offset this overhead, motivating the simpler formulation used in our experiments. Reducing the cost of candidate evaluation would further help translate this theoretical construction into a practical extension of TPL.

J Policy Variation and Monte Carlo Value Diagnostics

Checkpoints, states, and perspectives. We analyze fixed full-coverage TPL checkpoints on HBench-Periodic using four training seeds per task. These are the full-coverage rerun checkpoints, distinct from the default TPL cohort used in the main benchmark comparison. For each checkpoint, we use the same 128 replay states for every condition. We draw a pool of 2,048 perspectives from $\mathrm{Dirichlet}(\mathbf{1})$ and compute the normalized distance defined in equation 17. We select 16 perspectives from each of the five bins $[0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0]$ and evaluate the centroid separately. This gives 81 conditions per checkpoint and 972 checkpoint–condition pairs. The same selected conditions are used across checkpoints. The mean distances of the five bins are $0.155, 0.301, 0.484, 0.690, 0.858$.

Action difference. For a squashed Gaussian actor, let $m_{\boldsymbol{\alpha}}(s) = \tanh \mu_\theta(s, \boldsymbol{\alpha})$. We measure

$$D(\boldsymbol{\alpha}) = \frac{1}{N} \sum_{i=1}^{N} \left\| m_{\boldsymbol{\alpha}}(s_i) - m_{\boldsymbol{\alpha}_{\mathrm{ref}}}(s_i) \right\|_2, \qquad N = 128. \tag{41}$$

The norm is taken over all 19 normalized action coordinates.

Common-reference critic readout. To compare values with a common temporal weighting, we evaluate

$$\widehat{Q}^{\pi^{\boldsymbol{\alpha}}}_{\mathrm{ref}}(s, a) = \sum_{\ell=1}^{L} \frac{1}{M} \sum_{j=1}^{M} Q^{(\ell)}_{\psi,j}(s, a, \boldsymbol{\alpha}), \qquad M = 2. \tag{42}$$

The critic condition and rollout continuation policy remain $\boldsymbol{\alpha}$; only the residual readout uses reference weights. We do not compare a perspective-weighted critic against a different horizon.

Monte Carlo target. At each state, we sample $K = 4$ first actions from $\pi^{\boldsymbol{\alpha}}(\cdot \mid s)$ and keep each action fixed across $R = 8$ independent continuation rollouts. Let $\gamma = \gamma_L$ and let η be the temperature stored in the checkpoint. For horizon $H = 200$, a rollout target is

$$G_H = r_0 + \sum_{t=1}^{H-1} \gamma^t \left[r_t - \eta \log \pi^{\boldsymbol{\alpha}}(a_t \mid s_t) \right]. \tag{43}$$

Contributions after termination are zero, and no tail bootstrap is added. There is no entropy term at $t = 0$, since the first action is fixed in the action-value comparison. The joint-action log probability has the same scale as in critic training. We compute

$$E_{e,z}(\boldsymbol{\alpha}) = \sqrt{\frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \left(\widehat{Q}^{\pi^{\boldsymbol{\alpha}}}_{\mathrm{ref}}(s_i, a_{ik}) - \frac{1}{R} \sum_{r=1}^{R} G_{H,ikr} \right)^2}, \tag{44}$$

where e indexes environments and z training seeds. We average these perspective-level RMSEs within each bin before normalizing by the same environment and seed's reference RMSE:

$$R_{e,z,b} = \frac{\mathrm{mean}_{\boldsymbol{\alpha} \in b} E_{e,z}(\boldsymbol{\alpha})}{E_{e,z}(\boldsymbol{\alpha}_{\mathrm{ref}})}, \qquad \overline{R}_b = \frac{1}{3} \sum_{e} \frac{1}{4} \sum_{z} R_{e,z,b}. \tag{45}$$

K PRELIMINARY RESULTS ON ATARI 100K

To observe the generality of TPL beyond state-based continuous control, we conduct a lightweight evaluation on the Atari 100k benchmark. This benchmark serves as a challenging testbed for sample efficiency in discrete action spaces with high-dimensional image observations.

K.1 EXPERIMENTAL RESULTS

Table 8 presents the full results. Following standard protocols, we report raw scores for individual games to allow direct comparison with reference scores, while the aggregate metrics (Mean, Median, IQM) are calculated using Human Normalized Scores (HNS).

$$\text{HNS} = \frac{\text{Score}_{\text{Agent}} - \text{Score}_{\text{Random}}}{\text{Score}_{\text{Human}} - \text{Score}_{\text{Random}}} \tag{46}$$

The TPL variant achieves higher aggregate scores than the SAC baseline, motivating further evaluation across modalities.

Table 8: **Performance on Atari 100k (26 games).** We compare SAC (Baseline) and SAC+TPL (Ours) against Random and Human reference scores across 4 seeds. Individual game scores are raw returns (higher is better). Aggregate metrics (Mean, Median, IQM) are calculated using Human Normalized Scores (HNS).

Game	Random	Human	SAC	SAC+TPL
Alien	227.8	7127.7	785.5	**933.5**
Amidar	5.8	1719.5	156.5	**169.9**
Assault	222.4	742.0	623.6	**763.0**
Asterix	210.0	8503.3	1226.3	**1522.5**
BankHeist	14.2	753.1	37.5	37.5
BattleZone	2360.0	37187.5	7700.0	**10175.0**
Boxing	0.1	12.1	74.1	**83.8**
Breakout	1.7	30.5	**10.1**	8.6
ChopperCommand	811.0	7387.8	**1500.0**	1362.5
CrazyClimber	10780.5	35829.4	16905.0	**37497.5**
DemonAttack	152.1	1971.0	365.4	**774.3**
Freeway	0.0	29.6	30.5	**30.9**
Frostbite	65.2	4334.7	817.3	**946.5**
Gopher	257.6	2412.5	265.5	**710.0**
Hero	1027.0	30826.4	3355.8	**3757.3**
Jamesbond	29.0	302.8	207.5	**422.5**
Kangaroo	52.0	3035.0	1192.5	**2742.5**
Krull	1598.0	2665.5	2705.8	**4265.5**
KungFuMaster	258.5	22736.3	**10980.0**	7937.5
MsPacman	307.3	6951.6	**1603.3**	1052.8
Pong	-20.7	14.6	0.7	**19.5**
PrivateEye	24.9	69571.3	**89.5**	-93.1
Qbert	163.9	13455.0	597.5	**797.5**
RoadRunner	11.5	7845.0	9860.0	**13900.0**
Seaquest	68.4	42054.7	354.5	**423.0**
UpNDown	533.4	11693.2	3626.3	**5424.5**
Mean HNS (↑)	0.000	1.000	0.553	**0.793**
Median HNS (↑)	0.000	1.000	0.186	**0.232**
IQM HNS (↑)	0.000	1.000	0.237	**0.390**

K.2 IMPLEMENTATION DETAILS

Since Soft Actor-Critic (SAC), the base algorithm of TPL, was originally designed for low-dimensional continuous control, we adapted it for pixel-based discrete control following the methodology of MR.Q Fujimoto et al. (2025).

- **Discrete Action Adaptation:** Instead of the Gaussian policy, we utilize a Gumbel-Softmax distribution. This enables differentiable sampling of discrete actions, allowing gradients to propagate through the actor network.
- **Image Encoder Architecture:** To handle high-dimensional image inputs, we employ a simple CNN encoder architecture shared by the actor and residual critics, consistent with the SAC baseline for Atari. The perspective mixture vector α is concatenated with the flattened feature vector after the CNN encoder.
- **Algorithm Settings:** We maintain the core TPL structure but omit extensive hyperparameter tuning for this auxiliary image-based domain, relying on the robust default settings of the baseline.

L FULL TRAINING CURVES

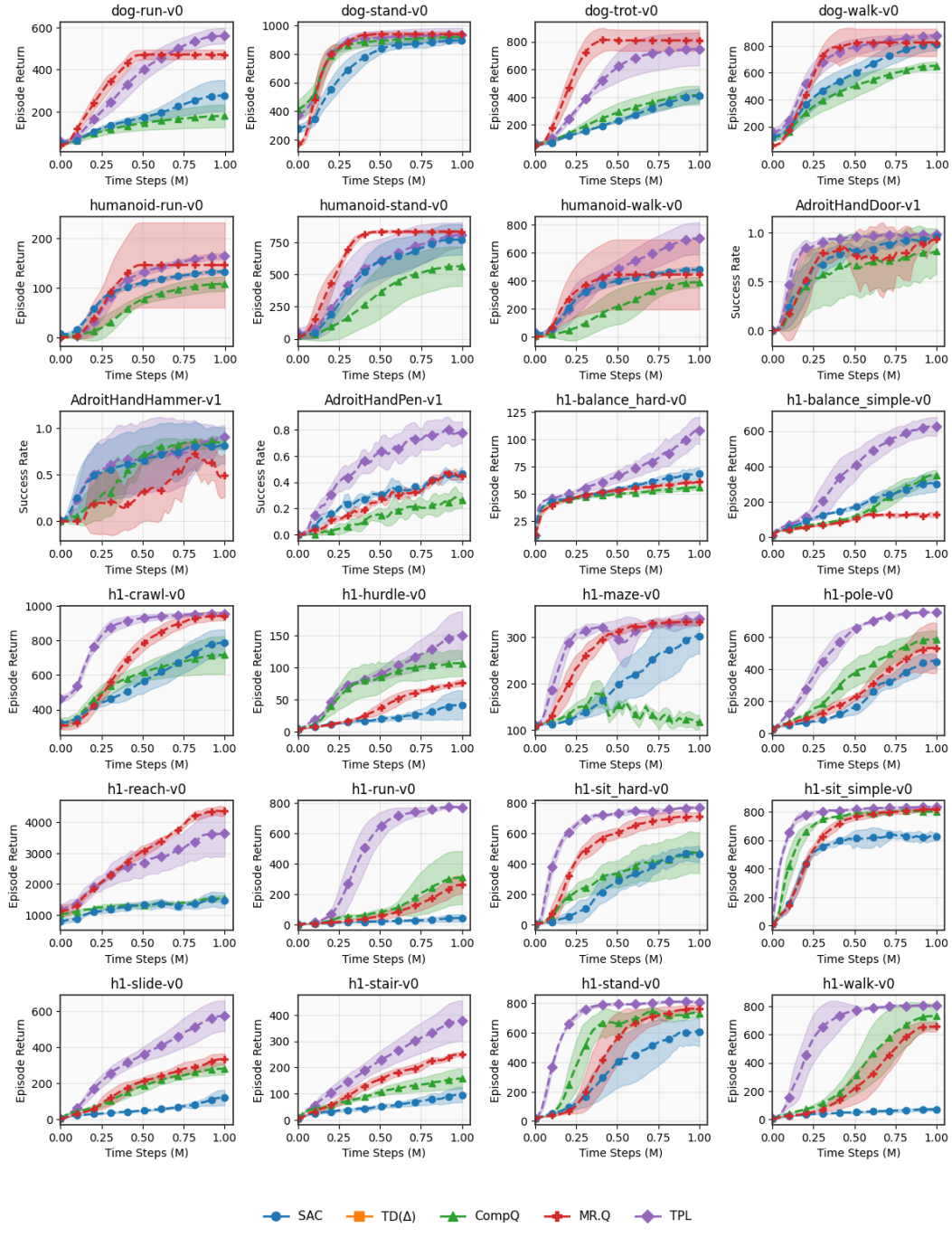

Figure 11: **Full learning curves of main baselines for all environments.**

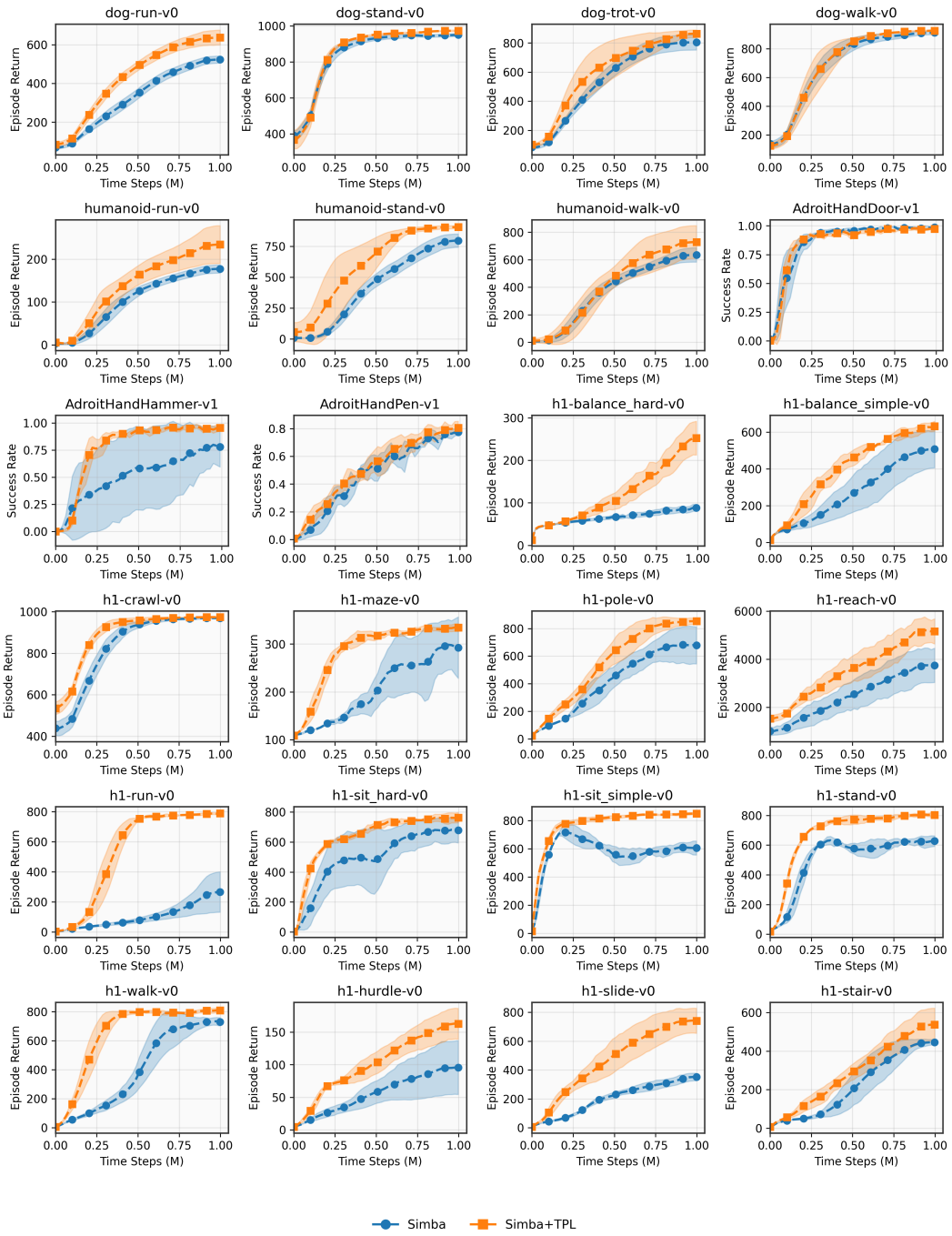

Figure 12: **Full learning curve comparisons of Simba for all environments.**

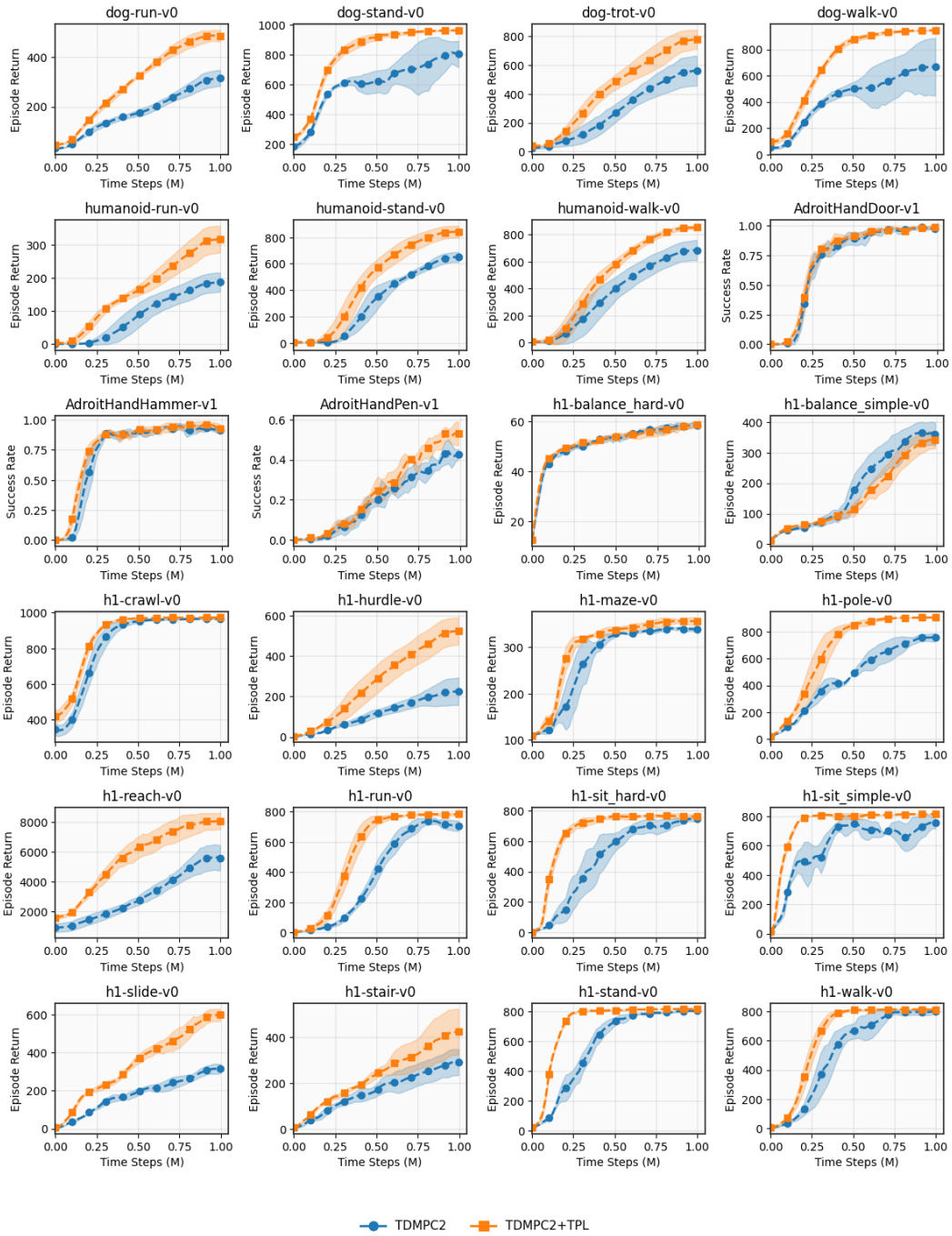

Figure 13: **Full learning curve comparisons of TD-MPC2 for all environments.**