SHARV: A SIMPLE MULTI-HORIZON APPROACH TO REPRESENTATION AND VALUE LEARNING IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective use of collected experience is essential for sample-efficient reinforcement learning. However, existing methods often rely on fixed horizon configurations, limiting their ability to exploit learning signals across environments with different temporal characteristics. We show that short- and long-horizon prediction have environment-dependent effects and that exhaustively combining prediction targets does not consistently improve performance. We present SHARV, a simple approach that exploits multiple horizons for representation and value learning. Our representation learning combines independently learned short- and long-horizon representations to support policy and value learning across diverse environments. For value learning, we characterize internal TD residual components not determined by the first-start target error, motivating Multi-Start Bootstrapping to control these residuals through supervision across multiple starting points. SHARV achieves the highest mean aggregate score among strong baselines on each of four benchmark suites. Ablation studies support the contribution of both components, and observed reductions in internal TD residuals provide empirical support for our theoretical analysis.

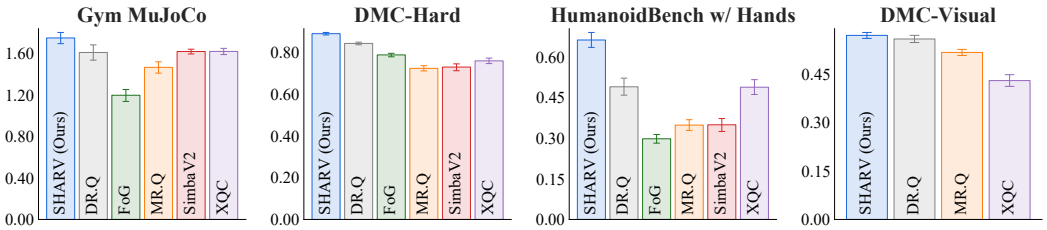

Figure 1: **Aggregate Performance.** SHARV achieves the highest mean normalized score in every benchmark suite. Bars show means, and error bars indicate 95% confidence intervals.

1 INTRODUCTION

Improving sample efficiency is a central challenge in reinforcement learning because collecting experience requires time and resources. Existing methods reuse collected experience or learn environment models to generate synthetic transitions. In model-free reinforcement learning, DQN (Mnih et al., 2015) samples transitions from a replay buffer to reuse experience and reduce temporal correlations. In model-based reinforcement learning, Dyna-style methods, such as Dyna-Q (Sutton, 1990) and MBPO (Janner et al., 2019), learn environment models from real experience and generate transitions for additional value updates. Both approaches enable more learning from existing experience without additional environment interactions.

Learned models also support action selection through planning. TD-MPC (Hansen et al., 2022) combines short-horizon planning in a learned latent space with a terminal value function that estimates long-term returns. Alongside these planning-based approaches, predictive representation methods use latent prediction objectives to learn features that support policy and value learning. In particular, MR.Q (Fujimoto et al., 2025) incorporates multi-step latent prediction and achieves competitive or superior performance across diverse environments. It uses dynamics and reward prediction to learn

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.

representations for a model-free actor-critic, without requiring online planning or synthetic transitions for optimization. DR.Q (Lyu et al., 2026) further improves this approach through contrastive representation learning and faded prioritized experience replay. These developments highlight the importance of both reusing experience and extracting useful supervision from it.

However, these advances leave room for more effective use of supervision across multiple horizons. In representation learning, one-step (teacher forcing) prediction and recursive multi-step (autoregressive) prediction impose different learning objectives: the former predicts from representations of observed states at each step, whereas the latter uses its own predicted representations as inputs to subsequent steps. SPR (Schwarzer et al., 2021), MR.Q (Fujimoto et al., 2025), and DR.Q (Lyu et al., 2026) supervise multiple rollout depths along a recursive prediction path with a fixed maximum horizon. Yet which prediction scheme works better may depend on the environment.

A second limitation concerns the use of multi-step experience for value estimation. MR.Q and its variants (Chang et al., 2026; Lyu et al., 2026) sample consecutive transitions but compress each sequence into a single multi-step target for the initial state-action pair. Although intermediate rewards contribute to this target, intermediate state-action pairs receive no direct critic supervision in that update. Meanwhile, other recent state-of-the-art methods, such as XQC (Palenicek et al., 2026) and FoG (Kang et al., 2025), rely on one-step backups in their state-based implementations. These approaches leave opportunities for richer supervision across multiple backup horizons. In particular, a sampled sequence can provide additional value targets by backing up returns from a common endpoint to intermediate starting points, enabling more learning from the same experience without additional environment interactions.

To address these limitations, we propose SHARV, a **S**imple multi-**H**orizon **A**pproach to **R**epresentation and **V**alue learning. For representation learning, SHARV uses two prediction streams with independent parameters for one-step and recursive multi-step supervision, combining their representations for policy and value estimation to reduce reliance on either regime alone. For value learning, SHARV supervises multiple starting points within a sampled sequence using a common bootstrap endpoint, providing critic targets across different backup horizons. Both components obtain more varied temporal supervision from collected experience.

Our experiments show that SHARV achieves the highest aggregate score on each of the four benchmark suites, outperforming strong baselines including DR.Q and XQC. Ablation studies support the benefits of combining short- and long-horizon representations and show that supervising multiple starting points further improves sample efficiency. These results demonstrate the effectiveness of a simple approach that exploits multiple horizons for both representations and value estimation.

Our main contributions are as follows:

- We introduce Decoupled Dual Representation learning, which captures both short- and long-horizon predictive information to provide consistently effective representations for policy and value learning across diverse environments.
- We leverage Multi-Start Bootstrapping to obtain richer value-learning supervision from sampled sequences, and theoretically show that controlling multi-start target errors bounds within-sequence residuals that are not bounded by the first-start target error alone.
- We empirically demonstrate that SHARV achieves the highest mean aggregate score among strong baselines on each of four benchmark suites. We further provide empirical evidence supporting our theoretical analysis of within-sequence residual control.

2 Related Works

Self-Predictive Representation Learning. Self-predictive representation learning aims to predict the latent representations of future observations from the current representation and action. Providing these learned representations to the actor and critic allows policy and value learning to exploit information about environment dynamics, potentially improving sample efficiency. TD7 (Fujimoto et al., 2023) supplies the actor and critic with state and state–action representations learned through one-step prediction. Combined with other improvements to TD3 (Fujimoto et al., 2018), this approach yields substantial performance gains. MR.Q (Fujimoto et al., 2025) learns representations through recursive multi-step prediction and reward prediction, achieving strong performance across

diverse environments. Related studies have also examined and refined self-predictive learning: Minimalist ϕ (Ni et al., 2024) and Tang et al. (2023) analyze the theoretical properties of self-predictive learning and its optimization schemes, while R2R2 (Song et al., 2026) introduces regularization to mitigate representation overfitting under intensive experience reuse. In particular, DR.Q (Lyu et al., 2026) augments the dynamics prediction objective with contrastive learning and incorporates faded prioritized experience replay, improving performance over MR.Q across multiple benchmarks.

Value Estimation. Many off-policy value-based and actor-critic methods (Mnih et al., 2015; Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018; Fujimoto et al., 2023; 2025; Lee et al., 2025b; Palenicek et al., 2026) estimate value functions through temporal-difference (TD) learning based on the Bellman equation. They update predictions toward bootstrap targets that combine observed rewards with value estimates at subsequent states. Extending one-step TD, n-step returns combine cumulative rewards with an endpoint bootstrap value, enabling faster reward propagation. Rainbow (Hessel et al., 2018) incorporates such multi-step learning to improve sample efficiency. However, updating only the initial state-action pair leaves intermediate pairs without direct supervision in that update. Off-policy discrepancies can also bias uncorrected multi-step targets, motivating correction methods such as Retrace (Munos et al., 2016). Furthermore, a small Bellman error on observed data does not necessarily imply accurate value estimation. Fujimoto et al. (2022) show that value errors can partially cancel within the Bellman equation and that incomplete data coverage can admit solutions with zero empirical Bellman error but inaccurate values. These findings highlight the importance of generalization beyond fitting bootstrap targets on the available data.

3 Preliminaries

Problem Setup. We model the environment as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition distribution, R is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At time t, the agent takes action a_t in state s_t and receives reward r_t and next state s_{t+1}. We seek a policy π that maximizes expected discounted return, with its action-value function Q^π defined in equation 1:

$$Q^\pi(s, a) = \mathbb{E}_{P,\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \,\middle|\, s_t = s,\ a_t = a\right]. \tag{1}$$

In off-policy learning, behavior-policy transitions are stored in a replay buffer $\mathcal{D}$ and reused for policy and value learning. We sample sequences of H consecutive transitions, denoted by $\tau_{t:t+H}$:

$$\tau_{t:t+H} = \big(s_t, a_t, r_t, s_{t+1}, \ldots, a_{t+H-1}, r_{t+H-1}, s_{t+H}\big). \tag{2}$$

Actions in equation 2 follow the behavior policies used during collection and may differ from the current policy. Episode boundaries are handled using stored termination and truncation indicators.

3.1 Self-Predictive Representation Learning

Self-predictive learning (SPL) learns representations by predicting the next state's latent representation from the current representation and action, as illustrated in Figure 2. Let $\phi : \mathcal{S} \to \mathbb{R}^d$ denote a state encoder, where d is the representation dimension, and let $\mathcal{T}$ denote a latent transition predictor (latent model). For a transition (s_t, a_t, s_{t+1}), define the current representation z_t and the predicted next representation $\hat{z}_{t+1}$ as

$$z_t = \phi(s_t), \qquad \hat{z}_{t+1} = \mathcal{T}(z_t, a_t). \tag{3}$$

A target encoder $\bar{\phi}$ provides the next-state target for the prediction in equation 3. For transitions sampled from the replay buffer $\mathcal{D}$, a one-step SPL objective is

$$\mathcal{L}_{\text{SPL}} = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}}\left[\left\|\hat{z}_{t+1} - \text{sg}\big(\bar{\phi}(s_{t+1})\big)\right\|_2^2\right], \tag{4}$$

where sg denotes the stop-gradient operation. The objective in equation 4 updates ϕ and $\mathcal{T}$ through the prediction branch, while $\bar{\phi}$ is updated separately. The learned representations can then provide the actor and critic with information about environment dynamics for policy and value learning.

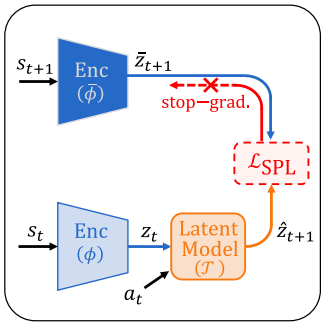

Figure 2: Typical 1-step self-predictive learning (SPL).

3.2 Multi-Step Value Learning

Critic targets combine observed rewards with a bootstrap value. For a backup horizon $n \geq 1$, let v_{t+n} denote the bootstrap value at the endpoint. Assuming no termination within the sequence, the n-step target, denoted by $y_t^{(n)}$, is

$$y_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n v_{t+n}. \tag{5}$$

The target in equation 5 supervises the critic prediction for the starting state-action pair (s_t, a_t). For example, in a basic actor-critic setting, the bootstrap value can be constructed using a target critic $Q_{\bar{\theta}}$, with parameters $\bar{\theta}$, and a target policy π:

$$v_{t+n} = \mathbb{E}_{a \sim \pi(\cdot|s_{t+n})} \left[Q_{\bar{\theta}}(s_{t+n}, a) \right]. \tag{6}$$

For a deterministic policy, equation 6 reduces to the single-action evaluation in equation 7:

$$v_{t+n} = Q_{\bar{\theta}}\big(s_{t+n}, \pi(s_{t+n})\big). \tag{7}$$

4 Method

Motivation Methods such as MR.Q (Fujimoto et al., 2025) and DR.Q (Lyu et al., 2026) sample sequences of H consecutive transitions for training. Once these sequences are available, a natural goal is to make fuller use of the supervision they provide. From this perspective, we explore denser supervision for both representation and value learning, making fuller use of the consecutive transitions for each learning objective without additional environment interactions.

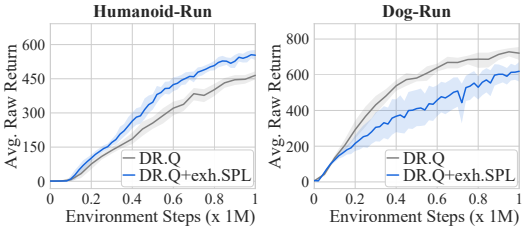

Figure 3: Contrasting effects of exhaustive pairwise SPL supervision on two DMC-Hard tasks.

For representation learning, we first evaluate an exhaustive pairwise variant of SPL (exh. SPL) that predicts between every start-end pair within a sequence and aggregates the losses to update encoder and prediction model. We test this variant on `humanoid-run` and `dog-run`, both belonging to DMC-Hard (Tassa et al., 2020; Lee et al., 2025a). As shown in Figure 3, it improves performance on the former but degrades on the latter. These contrasting results show that simply adding more prediction targets does not consistently improve performance, motivating a design that distinguishes how predictive supervision is provided.

4.1 Decoupled Dual Representation Learning

The exhaustive supervision examined in previous motivation changes the weighting of prediction horizons. A sequence of H transitions contains $H - h + 1$ pairs with horizon h, so equally weighting all pairs emphasizes shorter predictions. In contrast, a single recursive rollout contributes one prediction per horizon. The contrasting results in Figure 3 suggest environment-dependent effects, although exhaustive supervision also changes the prediction starting points.

Non-Equivalence of Short- and Long-Horizon Objectives. Short-horizon prediction starts from observed-state at each step, whereas long-horizon prediction recursively uses predicted representations, corresponding to teacher forcing and autoregressive prediction, respectively. We first compare their dynamics objectives using the same encoder ϕ, target encoder $\bar{\phi}$, and latent transition predictor $\mathcal{T}$. Let d denote the representation dimension. Each short-horizon prediction starts from the representation of an observed state. The corresponding dynamics objective $\mathcal{L}_{\text{dyn,S}}$ in equation 8 is

$$\mathcal{L}_{\text{dyn,S}} = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{k=0}^{H-1} \frac{1}{d} \left\| \mathcal{T}\big(\phi(s_{t+k}), a_{t+k}\big) - \text{sg}\big(\bar{\phi}(s_{t+k+1})\big) \right\|_2^2 \right]. \tag{8}$$

For long-horizon prediction, let $\mathcal{T}_a(z) = \mathcal{T}(z, a)$ and define the h-step composition $\mathcal{T}^{(h)}$ and predicted representation $\hat{z}_{t+h}$ as

$$\mathcal{T}^{(h)}(z, a_{t:t+h-1}) = \big(\mathcal{T}_{a_{t+h-1}} \circ \cdots \circ \mathcal{T}_{a_t}\big)(z), \quad \hat{z}_{t+h} = \mathcal{T}^{(h)}\big(\phi(s_t), a_{t:t+h-1}\big), \tag{9}$$

where $a_{t:t+h-1} = (a_t, \ldots, a_{t+h-1})$ and (h) denotes recursive composition over h steps. Using equation 9, the long-horizon dynamics objective $\mathcal{L}_{\mathrm{dyn,L}}$ is

$$\mathcal{L}_{\mathrm{dyn,L}} = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{h=1}^{H}\frac{1}{d}\left\|\mathcal{T}^{(h)}\big(\phi(s_t), a_{t:t+h-1}\big) - \mathrm{sg}\big(\bar{\phi}(s_{t+h})\big)\right\|_2^2\right]. \tag{10}$$

Although equation 10 includes a one-step term, it is generally not equivalent to equation 8. Beyond the first step, its inputs are predicted representations, and gradients propagate through earlier predictions and the initial encoding. Therefore, optimizing one dynamics objective does not necessarily minimize the other dynamics objective.

Combining Representations from Both Prediction Regimes. Figure 4(a) illustrates Decoupled Dual Representation (DDR), with independent encoders, target encoders, and prediction models for the two streams. Both streams use the same sampled sequences but are optimized independently under their respective SPL regimes, using the full objective described below. The short stream restarts from each observed state, while the long stream retains gradients through successive predictions.

Let $z_{\mathrm{S},t} = \phi_{\mathrm{S}}(s_t)$ and $z_{\mathrm{L},t} = \phi_{\mathrm{L}}(s_t)$ denote the learned state representations. We concatenate them into the combined representation $z_{\mathrm{SL},t}$:

$$z_{\mathrm{SL},t} = [z_{\mathrm{S},t}; z_{\mathrm{L},t}] = [\phi_{\mathrm{S}}(s_t); \phi_{\mathrm{L}}(s_t)], \tag{11}$$

where $[\,\cdot\,;\,\cdot\,]$ denotes concatenation. The combination in equation 11 allows downstream networks to learn how to use both streams without prescribed weights. We preserve the baseline representation dimension d_{base} by imposing $d_{\mathrm{S}} + d_{\mathrm{L}} = d_{\mathrm{base}}$, where d_{S} and d_{L} are the stream dimensions.

Full Representation Learning Objective. The dynamics objectives above isolate the distinction between prediction regimes. For training, we retain DR.Q's reward prediction and InfoNCE losses (Lyu et al., 2026). Each loss block $\mathcal{L}^{(h)}$ in Figure 4(a) supervises the prediction at step h. Let $\hat{z}_{t+h}$ denote the predicted representation and $\bar{z}_{t+h} = \mathrm{sg}(\bar{\phi}(s_{t+h}))$ the target representation. Let $\hat{r}_{t+h-1}$ denote the predicted reward distribution and r_{t+h-1} the observed reward. With loss weights λ_{dyn}, λ_{rew}, and λ_{NCE}, the full prediction loss is

$$\mathcal{L}_{\mathrm{rep}}^{(h)} = \lambda_{\mathrm{dyn}}\mathcal{L}_{\mathrm{dyn}}\big(\hat{z}_{t+h}, \bar{z}_{t+h}\big) + \lambda_{\mathrm{rew}}\mathcal{L}_{\mathrm{rew}}\big(\hat{r}_{t+h-1}, r_{t+h-1}\big) + \lambda_{\mathrm{NCE}}\mathcal{L}_{\mathrm{NCE}}\big(\hat{z}_{t+h}, \bar{z}_{t+h}\big). \tag{12}$$

In equation 12, $\mathcal{L}_{\mathrm{dyn}}$ is MSE over representation dimensions, $\mathcal{L}_{\mathrm{rew}}$ is cross-entropy against a two-hot reward target, and $\mathcal{L}_{\mathrm{NCE}}$ uses cosine similarity with other target representations in the minibatch as negatives. The minibatch dependence of InfoNCE is omitted for brevity. Each stream averages these losses over samples and sums over prediction steps, excluding positions after episode boundaries. The superscript (h) indexes the loss position: its latent prediction has horizon one in the short stream and horizon h in the long stream.

4.2 Multi-Start Bootstrapping and Error Analysis

As shown in Figure 4(b), we supervise every starting point in a sampled replay sequence using a common endpoint bootstrap value, exploiting multiple backup horizons. This structure is already used in on-policy actor-critic methods and asynchronous n-step Q-learning (Mnih et al., 2016); we call it Multi-Start Bootstrapping (MSB) for convenience. Although intermediate pairs could be sampled in later updates, supervising them together provides more critic supervision per window. We analyze which residual components the first-start target error captures and how multi-start supervision constrains the rest. The complete training procedure is detailed in Appendix B.

Multi-Step Targets and Error Definitions. Critic sequences are sampled independently of representation learning sequences, with length $K \geq 2$ that may differ from the representation horizon H. Consider a sampled sequence starting at time t without episode boundaries. At each position $i = 0, \ldots, K-1$, we use the n-step target in equation 5 with $n = K - i$, writing $y_{t+i} \equiv y_{t+i}^{(K-i)}$. These targets share the endpoint bootstrap value v_{t+K} and span backup horizons from K steps to one step, as illustrated in Figure 4(b). All targets are held fixed during the update. Denoting the critic loss at position i by $\ell_{\mathrm{critic},i}$, we define the overall critic objective $\mathcal{L}_{\mathrm{critic}}$ as

$$\mathcal{L}_{\mathrm{critic}} = \mathbb{E}_{\tau\sim\mathcal{D}}\left[\frac{1}{K}\sum_{i=0}^{K-1}\ell_{\mathrm{critic},i}\right]. \tag{13}$$

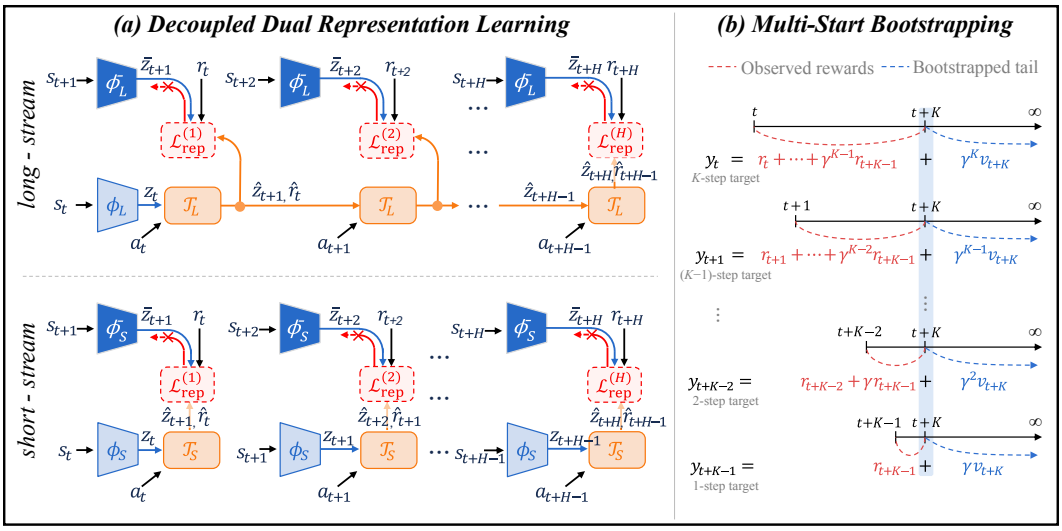

Figure 4: Overview of SHARV. (a) Decoupled Dual Representation (DDR) learning independently learns short- and long-stream representations and concatenates them for policy and value learning. (b) Multi-Start Bootstrapping (MSB) supervises multiple starting points within a sampled sequence using a common endpoint bootstrap value.

The averaging in equation 13 combines supervision across starting points into a single optimizer update. The critic parameterization and specific loss are described in § 4.3.

For the analysis, fix the sequence and its endpoint bootstrap value. Let θ denote the critic parameters and Q_θ its scalar value prediction. At positions $i = 0, \ldots, K-1$, abbreviate the rewards, targets, and predictions as $r_i = r_{t+i}$, $y_i = y_{t+i}$, and $Q_i = Q_\theta(s_{t+i}, a_{t+i})$. At the endpoint, define $Q_K = y_K = v_{t+K}$, with Q_K serving as a fixed boundary value. Define the multi-step target error e_i and the one-step residual δ_i by

$$e_i = y_i - Q_i, \qquad \delta_i = r_i + \gamma Q_{i+1} - Q_i. \tag{14}$$

In equation 14, e_i measures the discrepancy from the remaining-horizon target, whereas δ_i measures consistency between adjacent predictions along the recorded sequence. Let $\boldsymbol{\delta} = (\delta_0, \ldots, \delta_{K-1})^\top$ denote the residual vector.

Limitations of First-Start Supervision. A small target error at the first starting point need not imply small residuals throughout the sequence. The following proposition formalizes this limitation when intermediate predictions can vary independently.

Proposition 1 (First-start target error does not bound internal residuals)**.** *Fix a sequence of $K \geq 2$ transitions, its endpoint bootstrap value, and the first-start target error e_0. If intermediate predictions are treated as independently variable, unrestricted scalars, then for any threshold $M > 0$, there exist predictions preserving e_0 such that*

$$\frac{1}{K}\|\boldsymbol{\delta}\|_2^2 > M. \tag{15}$$

To interpret equation 15, let $w = (1, \gamma, \ldots, \gamma^{K-1})^\top$ denote the discount-weight vector and $\boldsymbol{\delta}_\perp$ the residual component orthogonal to w. Orthogonal decomposition gives

$$\frac{1}{K}\|\boldsymbol{\delta}\|_2^2 = \underbrace{\frac{1}{K}\frac{e_0^2}{\|w\|_2^2}}_{\text{determined by the first-start error}} + \underbrace{\frac{1}{K}\|\boldsymbol{\delta}_\perp\|_2^2}_{\text{not bounded by the first-start error alone}}. \tag{16}$$

The proof and decomposition are derived in Appendix A.1. In equation 16, reducing e_0^2 reduces only the first term. Under the unrestricted predictions of Proposition 1, the second can grow arbitrarily for fixed e_0, even when $e_0 = 0$.

In contrast, target errors across all starting points sharing the same endpoint bound the second term:

$$\frac{1}{K}\|\boldsymbol{\delta}_{\perp}\|_2^2 \leq \frac{1}{K}\|\boldsymbol{\delta}\|_2^2 \leq (1+\gamma)^2 \frac{1}{K}\sum_{i=0}^{K-1} e_i^2. \tag{17}$$

Thus, equation 17 shows that controlling target errors across starts also controls the residual term not bounded by the first-start error. Single-start learning may supervise intermediate pairs in later updates, whereas multi-start supervision uses them directly within the current sampled sequence.

4.3 Actor-Critic Architecture

Inspired by XQC (Palenicek et al., 2026), our actor-critic backbone uses batch normalization (BN) (Ioffe & Szegedy, 2015) and weight normalization (WN) (Salimans & Kingma, 2016) in both networks, with categorical critics trained by cross-entropy. Each critic parameterized by θ predicts probabilities $p_{\theta,m}(s,a)$ over fixed atoms $c_1,\ldots,c_M$. Let $p_{i,m}$ denote probabilities obtained by projecting the multi-step target distribution at starting point i onto this support. We average the per-position losses across starting points (§ 4.2), with the scalar value and loss given in equation 18:

$$Q_\theta(s,a) = \sum_{m=1}^{M} c_m p_{\theta,m}(s,a), \quad \ell_{\text{critic},i} = -\sum_{m=1}^{M} p_{i,m} \log p_{\theta,m}(s_{t+i}, a_{t+i}). \tag{18}$$

Unlike XQC's stochastic SAC policy (Haarnoja et al., 2018), our deterministic actor maximizes the critics' expected return estimates without entropy regularization. Network configurations and hyperparameters are in Appendix C.1.

5 Experiments

5.1 Performance Comparison

Environments. We evaluate SHARV on both proprioceptive and visual continuous-control benchmarks. For proprioceptive control, we use five Gym MuJoCo (Todorov et al., 2012) tasks with the v4 environment versions, seven tasks from DMC-Hard (Lee et al., 2025a), a subset of the DeepMind Control (DMC) Suite (Tassa et al., 2018; 2020), and 12 tasks from the with-hands configuration of HumanoidBench (Sferrazza et al., 2024). For visual control, we evaluate 10 pixel-based tasks from the DMC Suite. The complete task lists and environment settings are provided in Appendix C.2.

Baselines. We focus on strong recent baselines, including DR.Q (Lyu et al., 2026), XQC (Palenicek et al., 2026), SimbaV2 (Lee et al., 2025b), FoG (Kang et al., 2025), and MR.Q (Fujimoto et al., 2025). Following their primary reported settings, we use results at an update-to-data (UTD) ratio of 2 for XQC and SimbaV2, and 1 for the remaining baselines. We also considered TD-MPC2 (Hansen et al., 2024) and DreamerV3 (Hafner et al., 2023), but omit them because prior comparisons, particularly those reported by DR.Q, show that they are outperformed by the selected baselines both in aggregate performance and on most individual tasks.

Implementation Details. We follow DR.Q's representation dimensions and XQC's actor and critic widths and depths, keeping the combined dual-stream output dimension equal to the baseline's. We train SHARV with an update-to-data (UTD) ratio of 1. Results are averaged over 10 random seeds, with baseline results obtained from public releases or reproduced when unavailable. Full details are provided in Appendices C.1 and C.3.

Results. Table 1 shows that SHARV achieves the highest mean aggregate score on all four benchmark suites, covering both proprioceptive and visual observations. The advantage is particularly pronounced on HumanoidBench, while the improvement over DR.Q on DMC-Visual is more modest. The per-task results in the Appendix D show that gains vary across environments. Figure 5 presents learning curves for selected tasks with notable gains. One possible interpretation is that SHARV makes better use of information in collected experience that existing methods may leave underused. Gains may be larger where additional supervision across horizons offers more room for improvement, and modest where existing methods already use this information effectively or other bottlenecks limit performance.

Table 1: Aggregate normalized scores across benchmark suites (10-seed means; 95% CIs in brackets). Bold marks the highest mean per suite. See the Appendix C for evaluation details.

	Proprioceptive			Visual
Method	Gym MuJoCo	DMC-Hard	HumanoidBench	DMC-Visual
SHARV (Ours)	**1.748** [1.693, 1.801]	**0.889** [0.883, 0.895]	**0.662** [0.635, 0.690]	**0.571** [0.561, 0.580]
DR.Q	1.608 [1.534, 1.681]	0.842 [0.836, 0.848]	0.489 [0.458, 0.521]	0.559 [0.548, 0.571]
XQC	1.618 [1.588, 1.647]	0.759 [0.746, 0.772]	0.488 [0.461, 0.516]	0.430 [0.413, 0.449]
FoG	1.196 [1.137, 1.251]	0.787 [0.778, 0.795]	0.298 [0.281, 0.313]	—
SimbaV2	1.617 [1.594, 1.639]	0.729 [0.712, 0.745]	0.349 [0.325, 0.372]	—
MR.Q	1.465 [1.409, 1.518]	0.723 [0.711, 0.736]	0.348 [0.328, 0.368]	0.518 [0.508, 0.527]

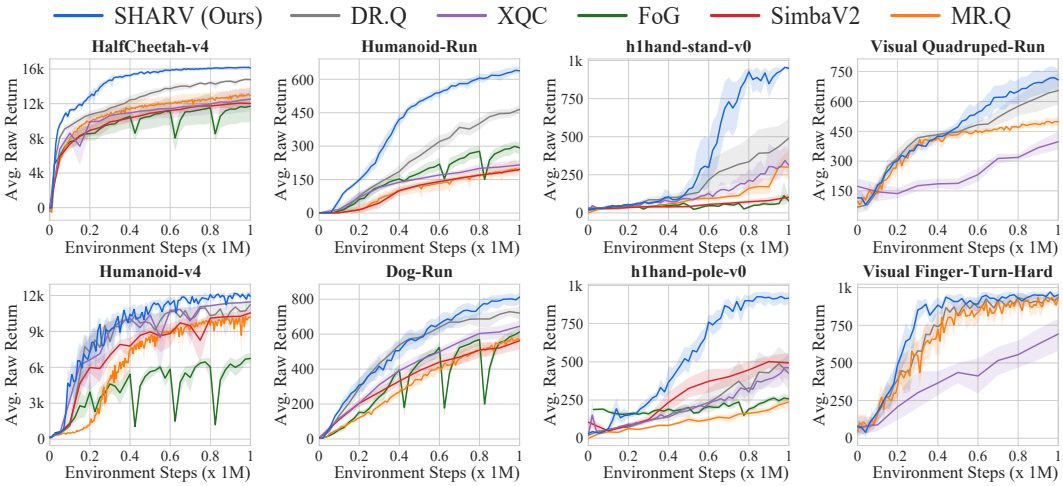

Figure 5: Selected learning curves from each suite. Lines and shading show mean returns and 95% confidence intervals across 10 seeds. Full results are in Appendix D.

5.2 Ablation Study and Error Analysis

Effect of Representation Configurations. Motivated by contrasting effects in § 4, we compare short-short (SS), long-long (LL), and short-long (SL) configurations over 10 seeds. All configurations use independently trained streams with the same total representation dimension and critic updates. Figure 6 shows that SS performs similarly to SL on `humanoid-run`, while LL performs worse; the pattern reverses on `dog-run`. SL matches or exceeds the better of SS and LL on both tasks, supporting the use of both prediction regimes across environments.

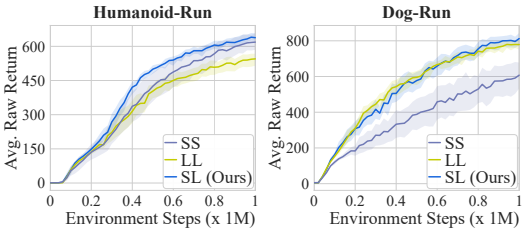

Figure 6: Comparison of short-short (SS), long-long (LL), and short-long (SL) streams. SS performs better on `humanoid-run` and LL on `dog-run`, while SL remains competitive with the better configuration on both tasks.

Contribution of Individual Components. We evaluate variants without DDR (§ 4.1) or MSB (§ 4.2) across all proprioceptive tasks over 10 random seeds. Without DDR, we use a single recursive prediction stream following DR.Q while preserving the total representation dimension and MSB updates. Without MSB, we supervise only the first state-action pair in each critic sequence, retaining DDR, the sequence length, and the endpoint bootstrap. As Table 2 shows, removing either component lowers the mean aggregate score on all three suites, supporting the contribution of each component to the combined method. Full results are provided in Appendix F.

Effect of MSB on Critic Residuals. On `humanoid-run`, we compare SHARV with and without MSB using the mean squared first-start target error E_{first} and one-step residual E_{TD}:

$$E_{\text{first}} = \mathbb{E}[e_0^2], \quad E_{\text{TD}} = \mathbb{E}\left[\frac{1}{K}\sum_{i=0}^{K-1}\delta_i^2\right]. \tag{19}$$

Both metrics in equation 19 use identical samples and averaging weights within each agent. From equation 16, the mean squared orthogonal residual component, denoted by $E_\perp$, is

$$E_\perp = \mathbb{E}\left[\frac{1}{K}\|\boldsymbol{\delta}_\perp\|_2^2\right] = E_{\text{TD}} - \frac{E_{\text{first}}}{K\sum_{i=0}^{K-1}\gamma^{2i}}. \tag{20}$$

We evaluate equation 20 with $K = 3$ and $\gamma = 0.99$. Measurements use each agent's own replay data and evolving policy, assessing online consistency.

Table 3 shows that during 200k–400k environment steps, the interval-averaged first-start errors differ by only 0.4%, while SHARV has a 57.6% lower $E_\perp$. During 800k–1,000k steps, E_{first} and $E_\perp$ decrease by 15.2% and 70.3% relative to the variant, respectively. These results support the motivation for MSB: similar first-start errors can coexist with substantially different internal residuals, and multi-start supervision is associated with a smaller component not determined by the first-start error.

Table 2: Component ablations on proprioceptive benchmarks: mean normalized scores with 95% CIs. DDR: § 4.1; MSB: § 4.2.

Method	MuJoCo	DMC-Hard	Hum. Bench
SHARV	**1.748** [1.693, 1.801]	**0.889** [0.883, 0.895]	**0.662** [0.635, 0.690]
SHARV w/o DDR	1.639 [1.562, 1.704]	0.871 [0.866, 0.877]	0.614 [0.587, 0.642]
SHARV w/o MSB	1.648 [1.579, 1.706]	0.847 [0.833, 0.859]	0.568 [0.544, 0.594]

Table 3: Critic errors on `humanoid-run` (DMC-Hard), averaged over 10 seeds and each interval. Values are scaled by 10^3; lower is better. Reductions are relative to SHARV w/o MSB.

Env. steps	Method	$E_{\text{first}} \downarrow$	$E_\perp \downarrow$
200k–400k	SHARV	**2.61**	**1.34**
	w/o MSB	2.62	3.16
	Reduction (%)	0.4	57.6
800k–1,000k	SHARV	**1.34**	**0.71**
	w/o MSB	1.58	2.39
	Reduction (%)	15.2	70.3

6 Conclusion

We introduced SHARV, a multi-horizon approach to representation and value learning that makes fuller use of sampled transition sequences. Our motivation stems from the observation that exhaustively adding predictive supervision does not consistently improve performance: its benefits vary across environments. We showed that teacher-forced one-step prediction and autoregressive multi-step prediction impose different learning objectives, and addressed their environment-dependent effectiveness through independently learned representation streams combined for policy and value learning. For critic learning, we employed supervision at multiple starting points and derived a bound relating within-sequence one-step TD residuals to multi-step target errors across those starting points. Experiments across proprioceptive and visual continuous-control benchmarks demonstrated improvements over strong baselines, while ablations supported both components and residual measurements provided empirical support for our theoretical analysis. Together, these findings establish SHARV as a simple yet powerful approach with a clear design for exploiting multiple horizons in representation and value learning.

Limitations and Future Work. Although SHARV improves sample efficiency, its independent representation streams and additional critic supervision increase training time (see Appendix E). We also combine the two representations through simple concatenation without extensively exploring alternatives that could reduce redundant information. Future work could investigate more efficient integration of the two streams. Our replay scheme favors recent experience, which may reduce discrepancies between the behavior and target policies. However, our multi-step critic targets do not explicitly correct for these discrepancies, leaving off-policy correction as another direction for improvement. Finally, our evaluation is limited to continuous-control tasks, and extending SHARV to discrete-action domains such as Atari (Bellemare et al., 2013) remains future work.

AI USE STATEMENT

We used generative AI to draft, translate, and refine the manuscript, help develop and write proofs of mathematical claims, and assist with method implementation. We did not use generative AI to develop theoretical models or conceptual frameworks, design or evaluate research methodology or experiments, generate or process datasets, conduct qualitative or thematic analyses, or interpret results. The authors reviewed and revised all AI-assisted text, verified translation accuracy, and independently checked the mathematical assumptions, derivations, proofs, and code for correctness. We take full responsibility for the final content of this work, including all AI-assisted text, claims, and artifacts.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29304–29320, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.*, 47:253–279, 2013. doi: 10.1613/JAIR.3912. URL https://doi.org/10.1613/jair.3912.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. *arXiv preprint arXiv:2601.21306*, 2026.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should I trust you, bellman? the bellman error is a poor replacement for value error. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 6918–6943. PMLR, 2022. URL https://proceedings.mlr.press/v162/fujimoto22a.html.

Scott Fujimoto, Wei-Di Chang, Edward J. Smith, Shixiang Gu, Doina Precup, and David Meger. For SALE: State-action representation learning for deep reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/c20ac0df6c213db6d3a930fe9c7296c8-Abstract-Conference.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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=R1hIXdST22.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023. doi: 10.48550/ARXIV.2301.04104. URL https://doi.org/10.48550/arXiv.2301.04104.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3215–3222. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11796. URL https://doi.org/10.1609/aaai.v32i1.11796.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ioffe15.html.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 12498–12509, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Zilin Kang, Chenyuan Hu, Yu Luo, Zhecheng Yuan, Ruijie Zheng, and Huazhe Xu. A forget-and-grow strategy for deep reinforcement learning scaling in continuous control. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/kang25c.html.

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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=jXLiDKsuDo.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=kfYxyvCYQ4.

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. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Jiafei Lyu, Zichuan Lin, Scott Fujimoto, Kai Yang, Yangkun Chen, Saiyong Yang, Zongqing Lu, and Deheng Ye. Debiased model-based representations for sample-efficient continuous control. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ZP1p8k106p.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. doi: 10.1038/NATURE14236. URL https://doi.org/10.1038/nature14236.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1928–1937. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1046–1054, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c3992e9a68c5ae12bd18488bc579b30d-Abstract.html.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive RL. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ms0VgzSGF2.

Daniel Palenicek, Florian Vogt, Joe Watson, Ingmar Posner, and Jan Peters. XQC: Well-conditioned optimization accelerates deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 901, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/ed265bc903a5a097f61d3ec064d96d2e-Abstract.html.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv Preprint arxiv:2403.10506*, 2024.

Sanghyeob Song, Donghyeok Lee, Jinsik Kim, and Sungroh Yoon. R2R2: Robust representation for intensive experience reuse via redundancy reduction in self-predictive learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.14026.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Bruce W. Porter and Raymond J. Mooney (eds.), *Machine Learning, Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990*, pp. 216–224. Morgan Kaufmann, 1990. doi: 10.1016/B978-1-55860-141-3.50030-4. URL https://doi.org/10.1016/b978-1-55860-141-3.50030-4.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle,

András György, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, and Michal Valko. Understanding self-predictive learning for reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33632–33656. PMLR, 2023. URL https://proceedings.mlr.press/v202/tang23d.html.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. DeepMind Control Suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy P. Lillicrap, and Nicolas Heess. dm_control: Software and tasks for continuous control. *CoRR*, abs/2006.12983, 2020. URL https://arxiv.org/abs/2006.12983.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. URL https://doi.org/10.1109/IROS.2012.6386109.

A PROOF

A.1 ANALYSIS OF MULTI-START CRITIC SUPERVISION

We use the notation introduced in § 4.2. The sampled sequence, discount factor γ, and endpoint bootstrap value v_{t+K} are fixed throughout the analysis. The identities and bounds below apply to any scalar predictions on this sequence. Proposition 1 additionally treats intermediate predictions as independently variable, unrestricted scalars; it does not assume that a particular critic architecture can realize every constructed prediction vector.

Relationship Between Target Errors and Residuals. Recall the target errors $e_i = y_i - Q_i$ and residuals $\delta_i = r_i + \gamma Q_{i+1} - Q_i$. The endpoint convention $Q_K = y_K = v_{t+K}$ gives $e_K = 0$. Using the target recursion $y_i = r_i + \gamma y_{i+1}$, we obtain

$$\begin{aligned} e_i - \gamma e_{i+1} &= (y_i - Q_i) - \gamma(y_{i+1} - Q_{i+1}) \\ &= r_i + \gamma Q_{i+1} - Q_i \\ &= \delta_i. \end{aligned} \tag{21}$$

Multiplying equation 21 by γ^i and summing over the sequence yields a telescoping sum. With the residual vector $\boldsymbol{\delta} = (\delta_0, \ldots, \delta_{K-1})^\top$ and discount-weight vector $w = (1, \gamma, \ldots, \gamma^{K-1})^\top$,

$$\begin{aligned} w^\top \boldsymbol{\delta} &= \sum_{i=0}^{K-1} \gamma^i \delta_i \\ &= \sum_{i=0}^{K-1} \gamma^i e_i - \sum_{i=0}^{K-1} \gamma^{i+1} e_{i+1} \\ &= e_0 - \gamma^K e_K = e_0. \end{aligned} \tag{22}$$

Thus, equation 22 shows that the first-start target error captures a single weighted combination of the internal residuals.

Derivation of the Orthogonal Decomposition. Write $\boldsymbol{\delta} = \alpha w + \boldsymbol{\delta}_\perp$, where α is a scalar coefficient and $\boldsymbol{\delta}_\perp$ is required to be orthogonal to w. The condition $w^\top(\boldsymbol{\delta} - \alpha w) = 0$, together with equation 22, gives $\alpha = e_0/\|w\|_2^2$. Since w has first entry one, its norm is nonzero, and the decomposition is uniquely defined:

$$\boldsymbol{\delta} = \frac{e_0}{\|w\|_2^2} w + \boldsymbol{\delta}_\perp, \qquad w^\top \boldsymbol{\delta}_\perp = 0. \tag{23}$$

The two components in equation 23 are orthogonal, so expanding the squared norm gives

$$\begin{aligned} \|\boldsymbol{\delta}\|_2^2 &= \left\| \frac{e_0}{\|w\|_2^2} w + \boldsymbol{\delta}_\perp \right\|_2^2 \\ &= \frac{e_0^2}{\|w\|_2^2} + \frac{2e_0}{\|w\|_2^2} w^\top \boldsymbol{\delta}_\perp + \|\boldsymbol{\delta}_\perp\|_2^2 \\ &= \frac{e_0^2}{\|w\|_2^2} + \|\boldsymbol{\delta}_\perp\|_2^2. \end{aligned} \tag{24}$$

Dividing equation 24 by K establishes equation 16 in the main text.

Proof of Proposition 1. Fix the first-start target error e_0 and an arbitrary threshold $M > 0$. Because $K \geq 2$, there exists a unit vector u orthogonal to w. For example, choose $u = (-\gamma, 1, 0, \ldots, 0)^\top / \sqrt{1+\gamma^2}$, which satisfies $\|u\|_2 = 1$ and $w^\top u = 0$. For a scalar $c > \sqrt{KM}$, construct the residual vector

$$\boldsymbol{\delta}^{(c)} = \frac{e_0}{\|w\|_2^2} w + cu. \tag{25}$$

Here, the superscript (c) indexes the constructed vectors. By orthogonality, equation 25 satisfies

$$w^\top \boldsymbol{\delta}^{(c)} = e_0, \qquad \frac{1}{K}\|\boldsymbol{\delta}^{(c)}\|_2^2 = \frac{e_0^2}{K\|w\|_2^2} + \frac{c^2}{K} > M. \tag{26}$$

The construction in equation 26 therefore preserves the discounted residual sum while making the mean squared residual exceed M.

To show that this residual vector corresponds to scalar predictions with the prescribed endpoint, denote the constructed predictions by $Q_i^{(c)}$ and define them recursively backward:

$$Q_K^{(c)} = v_{t+K}, \qquad Q_i^{(c)} = r_i + \gamma Q_{i+1}^{(c)} - \delta_i^{(c)}, \quad i = K-1, \ldots, 0. \tag{27}$$

By construction, equation 27 produces exactly the residuals $\delta_i^{(c)}$. Applying equation 22, the resulting first-start target error $y_0 - Q_0^{(c)}$ equals the fixed value e_0. Hence $Q_0^{(c)} = y_0 - e_0$ remains unchanged as c varies, while the intermediate predictions produce arbitrarily large internal residuals. This proves equation 15 and Proposition 1.

Control Through Target Errors Across Starting Points. Let $\boldsymbol{e} = (e_0, \ldots, e_{K-1})^\top$ denote the target-error vector. Define the shift operator S by $S\boldsymbol{e} = (e_1, \ldots, e_{K-1}, 0)^\top$. From equation 21, $\boldsymbol{\delta} = \boldsymbol{e} - \gamma S\boldsymbol{e}$. Since $\|S\boldsymbol{e}\|_2 \le \|\boldsymbol{e}\|_2$, the triangle inequality gives

$$\begin{aligned} \|\boldsymbol{\delta}\|_2 &= \|\boldsymbol{e} - \gamma S\boldsymbol{e}\|_2 \\ &\le \|\boldsymbol{e}\|_2 + \gamma \|S\boldsymbol{e}\|_2 \\ &\le (1+\gamma)\|\boldsymbol{e}\|_2. \end{aligned} \tag{28}$$

The decomposition in equation 24 also implies $\|\boldsymbol{\delta}_\perp\|_2^2 \le \|\boldsymbol{\delta}\|_2^2$. Squaring equation 28 and dividing by K therefore proves the complete inequality in equation 17. This bound holds regardless of whether the predictions can vary independently.

Connection to Empirical Measurements. For fixed K and γ, taking expectations in equation 16 gives

$$E_{\mathrm{TD}} = \frac{E_{\mathrm{first}}}{K \sum_{i=0}^{K-1} \gamma^{2i}} + E_\perp, \tag{29}$$

where E_{TD}, E_{first}, and $E_\perp$ are defined in the main text. Rearranging equation 29 establishes equation 20. The identity also holds for empirical averages when the same samples and averaging weights are used for all terms.

The unboundedness construction assumes unrestricted scalar predictions. For a bounded critic, residual magnitudes on a fixed sequence are finite. Nevertheless, the orthogonal decomposition and the multi-start error bound remain valid, and an exact first-start target fit does not generally eliminate the orthogonal residual component.

B TRAINING PROCEDURE

Algorithm 1 summarizes the training procedure of SHARV. Representation and critic updates use independently sampled sequences, with local indices used for readability. Gradients propagate through successive predictions in the long-horizon stream, while target representations and critic targets are held fixed during each update. The actor and critic use features from both streams, whose representation parameters are trained through their own objectives. Episode boundaries are respected when constructing predictions and targets, and terminal states omit bootstrapping.

Algorithm 1 Training Procedure of SHARV

Require: Batch size N, representation horizon H, critic horizon K, discount factor γ
Initialize independent encoders $\phi_{\mathrm{S}}, \phi_{\mathrm{L}}$ and prediction models $\mathcal{T}_{\mathrm{S}}, \mathcal{T}_{\mathrm{L}}$
Initialize actor π, critic Q, target networks, and replay buffer $\mathcal{D}$
for each environment step t **do**
    Observe s_t and encode $z_{\mathrm{SL}}(s_t) \leftarrow [\phi_{\mathrm{S}}(s_t); \phi_{\mathrm{L}}(s_t)]$
    Select a_t using $\pi(z_{\mathrm{SL}}(s_t))$ with exploration noise
    Execute a_t and observe r_t, s_{t+1}, and the termination indicator
    Store the transition in $\mathcal{D}$
    if a learning update is scheduled **then**
1. Decoupled Dual Representation Learning (Encoder and Transition model) (§ 4.1)
        Sample N sequences of H-horizon transitions from $\mathcal{D}$
        Short stream: predict one step from each observed-state representation
        Long stream: recursively predict from the initial representation over depths $1, \ldots, H$
        Compute target representations using $\bar{\phi}_{\mathrm{S}}$ and $\bar{\phi}_{\mathrm{L}}$ with stop-gradient
        Compute each stream's dynamics, reward, and InfoNCE losses
        Update both streams independently, retaining gradients through long-horizon predictions
2. Multi-Start Bootstrapping (Critic) (§ 4.2)
        Independently sample N sequences of K-horizon transitions from $\mathcal{D}$
        Compute the common endpoint bootstrap using the target networks
        Construct categorical targets at all starting points with backup horizons $K, \ldots, 1$
        Update Q using cross-entropy averaged over sequences and starting points
3. Policy and Target Updates (Actor)
        Update π using both streams' features to maximize estimated return
        Update target networks according to their configured schedules
    end if
end for

C Experimental Setup

C.1 Implementation Details and Hyperparameters

We follow the hyperparameter settings of DR.Q (Lyu et al., 2026), except for the modifications described in our method and the settings listed in Table 4. Each representation stream retains DR.Q's encoder hidden width, loss coefficients, optimizer settings, and update schedule. We halve the state and state-action representation dimensions within each stream so that their concatenated dimensions match those of DR.Q. The actor and critic adopt XQC (Palenicek et al., 2026)-inspired network configurations and normalization. Rewards used for critic learning are normalized by the running standard deviation of discounted reward accumulations, while representation learning uses unscaled rewards.

Batch-normalization running statistics are updated as $m \leftarrow 0.99m + 0.01m_{\text{batch}}$. Weight normalization projects each dense weight vector associated with an output unit to unit Euclidean norm after initialization and each optimizer update. The critic support is defined in normalized return units. The representation prediction horizon and maximum critic backup horizon remain unchanged from DR.Q; multi-start supervision additionally uses the intermediate starting points within each critic-learning sequence.

Table 4: Settings modified or introduced relative to DR.Q. All other hyperparameters follow DR.Q.

Setting	Value
Representation learning	
State representation dimension per stream	256
State-action representation dimension per stream	256
Sequence Length of Sampling (H)	5
Actor and critic architecture	
Actor hidden dim	256
Critic hidden dim	512
Critic hidden activation	ReLU
Actor and critic normalization	BN and WN
BN running-statistic update coefficient	0.01
Dense weight initialization	Orthogonal, gain $\sqrt{2}$
Critic output	Categorical distribution
Number of critic atoms	101
Critic support	$[-5, 5]$
Actor and critic optimization	
Learning rate schedule	Linear: $3 \times 10^{-4} \rightarrow 3 \times 10^{-5}$
Weight decay	0
Critic loss	Cross-entropy
Sequence Length of Sampling (K)	3

C.2 Environments

We evaluate SHARV on 34 continuous-control tasks across four benchmark suites: Gym MuJoCo (Todorov et al., 2012), DMC-Hard (Tassa et al., 2018; 2020; Lee et al., 2025a), HumanoidBench (Sferrazza et al., 2024), and DMC-Visual. The first three suites use state-vector observations, whereas DMC-Visual uses image observations. Table 5 lists all tasks and their observation and action dimensions. We use an action repeat of 2 for DMC-Hard, HumanoidBench, and DMC-Visual.

Gym MuJoCo. We consider five locomotion tasks from Gym MuJoCo: Ant, HalfCheetah, Hopper, Humanoid, and Walker2d, using the `v4` versions. These tasks cover different robot morphologies and require coordinated control for forward locomotion. Observations contain task-dependent physical state information, and actions control the robot actuators.

DMC-Hard. We consider seven challenging tasks from the DeepMind Control Suite, following the DMC-Hard selection in SimBa (Lee et al., 2025a). These comprise four dog tasks and three humanoid tasks, covering standing and locomotion at different speeds. Dog tasks have 223-

dimensional observations and 38-dimensional actions, while humanoid tasks have 67-dimensional observations and 21-dimensional actions.

HumanoidBench. We consider 12 tasks from HumanoidBench, using the `h1hand` configuration with two dexterous hands. These tasks cover locomotion, posture control, reaching, and object manipulation. All tasks share a 61-dimensional action space. Observations include the 151-dimensional robot state and, where applicable, additional object or target information.

DMC-Visual. We consider 10 image-based tasks from the DeepMind Control Suite. Each observation consists of three consecutive RGB frames of resolution 84×84, stacked along the channel dimension. The resulting input shape is $9 \times 84 \times 84$ in channel-first order. Actions retain the dimensions of the corresponding DMC tasks.

Table 5: Observation and action dimensions across the four benchmark suites. Visual observations are reported as channels $\times$ height $\times$ width after frame stacking.

Suite	Environment	Observation dimension	Action dimension
Gym MuJoCo	`Ant-v4`	27	8
	`HalfCheetah-v4`	17	6
	`Hopper-v4`	11	3
	`Humanoid-v4`	376	17
	`Walker2d-v4`	17	6
DMC-Hard	`dog-run`	223	38
	`dog-stand`	223	38
	`dog-trot`	223	38
	`dog-walk`	223	38
	`humanoid-run`	67	21
	`humanoid-stand`	67	21
	`humanoid-walk`	67	21
HumanoidBench	`h1hand-basketball-v0`	164	61
	`h1hand-bookshelf_hard-v0`	308	61
	`h1hand-crawl-v0`	151	61
	`h1hand-door-v0`	155	61
	`h1hand-pole-v0`	151	61
	`h1hand-reach-v0`	157	61
	`h1hand-run-v0`	151	61
	`h1hand-sit_hard-v0`	164	61
	`h1hand-slide-v0`	151	61
	`h1hand-stair-v0`	151	61
	`h1hand-stand-v0`	151	61
	`h1hand-walk-v0`	151	61
DMC-Visual	`acrobot-swingup`	$9 \times 84 \times 84$	1
	`cheetah-run`	$9 \times 84 \times 84$	6
	`finger-turn_easy`	$9 \times 84 \times 84$	2
	`finger-turn_hard`	$9 \times 84 \times 84$	2
	`hopper-hop`	$9 \times 84 \times 84$	4
	`humanoid-run`	$9 \times 84 \times 84$	21
	`humanoid-stand`	$9 \times 84 \times 84$	21
	`quadruped-run`	$9 \times 84 \times 84$	12
	`quadruped-walk`	$9 \times 84 \times 84$	12
	`walker-run`	$9 \times 84 \times 84$	6

C.3 Baselines

We use the publicly released results for DR.Q (Lyu et al., 2026), FoG (Kang et al., 2025), SimbaV2 (Lee et al., 2025b), and MR.Q (Fujimoto et al., 2025) from the DR.Q repository[1] and those for XQC (Palenicek et al., 2026) from the XQC repository[2]. For DR.Q and XQC evaluations included in our experiments but unavailable in these releases, we reproduce the results ourselves.

[1] `https://github.com/dmksjfl/DR.Q`

[2] `https://github.com/danielpalenicek/xqc`

C.4 Evaluation Metrics

We evaluate performance using undiscounted episodic returns. To aggregate results across tasks with different reward scales, we follow the normalization procedures in DR.Q (Lyu et al., 2026).

Gym MuJoCo. For each task e, we normalize the return R_e using the random-policy return R_e^{random} and the TD3 reference return R_e^{TD3}:

$$S_e = \frac{R_e - R_e^{\text{random}}}{R_e^{\text{TD3}} - R_e^{\text{random}}}. \tag{30}$$

A normalized score of 0 corresponds to random-policy performance, while 1 corresponds to the TD3 reference performance. The task-specific reference returns are listed in Table 6.

HumanoidBench. We normalize returns relative to the random-policy return and the task success reference return R_e^{success}:

$$S_e = \frac{R_e - R_e^{\text{random}}}{R_e^{\text{success}} - R_e^{\text{random}}}. \tag{31}$$

We follow the success normalization used in SimbaV2 (Lee et al., 2025b), using the `h1hand` random-policy returns and task success references listed in DR.Q (Lyu et al., 2026). Success references are shared between configurations with and without hands. A normalized score of 1 denotes the success reference return, not an empirical success rate. Table 6 lists the values for all 12 tasks.

DMC-Hard and DMC-Visual. For both DMC suites, we normalize episodic returns by 1000:

$$S_e = R_e/1000. \tag{32}$$

Aggregate Performance. For each benchmark suite $\mathcal{E}$, we report the mean and interquartile mean (IQM) of normalized scores using rliable (Agarwal et al., 2021). The aggregate mean assigns equal weight to each task:

$$S_{\text{aggregate}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \overline{S}_e, \tag{33}$$

where $\overline{S}_e$ is the mean normalized score over 10 seeds. IQM averages the middle 50% of scores across tasks and seeds, limiting the influence of extremes. MuJoCo and HumanoidBench scores can exceed 1 when returns exceed their references.

Confidence Intervals. We report 95% confidence intervals (CIs). For individual tasks, CIs are computed as $\mu \pm 1.96\,\hat{\sigma}/\sqrt{n}$, where μ and $\hat{\sigma}$ are the sample mean and sample standard deviation across $n = 10$ random seeds. For aggregate metrics, we use rliable (Agarwal et al., 2021) to compute stratified percentile bootstrap CIs with 50,000 resamples, resampling runs within each task. The lower and upper bounds are the 2.5th and 97.5th percentiles of the resulting bootstrap distribution.

Table 6: Normalization returns. Reference denotes TD3 returns for Gym MuJoCo and success returns for HumanoidBench; the latter's random returns use the with-hands configuration.

Suite	Environment	Random	Reference
Gym MuJoCo TD3 normalization	`Ant-v4`	−70.288	3942
	`HalfCheetah-v4`	−289.415	10574
	`Hopper-v4`	18.791	3226
	`Humanoid-v4`	120.423	5165
	`Walker2d-v4`	2.791	3946
HumanoidBench Success normalization	`h1hand-basketball-v0`	8.979	1200
	`h1hand-bookshelf_hard-v0`	14.848	2000
	`h1hand-crawl-v0`	278.868	700
	`h1hand-door-v0`	2.771	600
	`h1hand-pole-v0`	19.721	700
	`h1hand-reach-v0`	−50.024	12000
	`h1hand-run-v0`	1.927	700
	`h1hand-sit_hard-v0`	2.477	750
	`h1hand-slide-v0`	3.142	700
	`h1hand-stair-v0`	3.161	700
	`h1hand-stand-v0`	11.973	800
	`h1hand-walk-v0`	2.505	700

D FULL RESULTS

D.1 GYM MUJOCO

Table 7: **Full raw-return results on Gym MuJoCo.** Mean returns with 95% confidence intervals for 5 tasks at 1M time steps (action repeat 1).

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
Ant-v4	**8950.5** [8360.1, 9540.8]	8137.5 [7763.9, 8511.2]	7612.0 [7445.9, 7778.2]	6761.2 [6161.6, 7360.7]	7429.4 [7209.1, 7649.6]	6901.8 [6280.8, 7522.8]
HalfCheetah-v4	**16119.8** [15996.7, 16242.8]	14775.2 [14637.9, 14912.5]	12527.9 [11171.4, 13884.5]	11709.4 [9928.2, 13490.6]	12022.4 [11640.5, 12404.3]	12939.6 [11731.3, 14147.8]
Hopper-v4	3068.3 [2606.0, 3530.6]	2503.9 [1930.9, 3077.0]	3745.0 [3711.6, 3778.5]	1821.5 [1316.1, 2327.0]	**4053.5** [3928.3, 4178.7]	2692.6 [2104.9, 3280.3]
Humanoid-v4	**11922.8** [11547.9, 12297.6]	11239.4 [11052.6, 11426.1]	11460.8 [11328.1, 11593.4]	6736.9 [6318.9, 7154.9]	10545.8 [10195.1, 10896.5]	10223.7 [9919.2, 10528.1]
Walker2d-v4	6664.6 [5944.2, 7385.0]	6422.1 [5123.0, 7721.2]	6256.3 [5930.9, 6581.7]	5124.3 [4719.3, 5529.2]	**6938.4** [6691.3, 7185.5]	6039.5 [5652.0, 6427.0]

Table 8: **Normalized results on Gym MuJoCo.** TD3-normalized scores from Table 7 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
Ant-v4	**2.25** [2.10, 2.40]	2.05 [1.95, 2.14]	1.91 [1.87, 1.96]	1.70 [1.55, 1.85]	1.87 [1.81, 1.92]	1.74 [1.58, 1.89]
HalfCheetah-v4	**1.51** [1.50, 1.52]	1.39 [1.37, 1.40]	1.18 [1.05, 1.30]	1.10 [0.94, 1.27]	1.13 [1.10, 1.17]	1.22 [1.11, 1.33]
Hopper-v4	0.95 [0.81, 1.09]	0.77 [0.60, 0.95]	1.16 [1.15, 1.17]	0.56 [0.40, 0.72]	**1.26** [1.22, 1.30]	0.83 [0.65, 1.02]
Humanoid-v4	**2.34** [2.27, 2.41]	2.20 [2.17, 2.24]	2.25 [2.22, 2.27]	1.31 [1.23, 1.39]	2.07 [2.00, 2.14]	2.00 [1.94, 2.06]
Walker2d-v4	1.69 [1.51, 1.87]	1.63 [1.30, 1.96]	1.59 [1.50, 1.67]	1.30 [1.20, 1.40]	**1.76** [1.70, 1.82]	1.53 [1.43, 1.63]
IQM	**1.79** [1.74, 1.84]	1.69 [1.60, 1.78]	1.58 [1.54, 1.62]	1.24 [1.17, 1.31]	1.64 [1.60, 1.67]	1.50 [1.43, 1.56]
Mean	**1.75** [1.69, 1.80]	1.61 [1.53, 1.68]	1.62 [1.59, 1.65]	1.20 [1.14, 1.25]	1.62 [1.59, 1.64]	1.46 [1.41, 1.52]

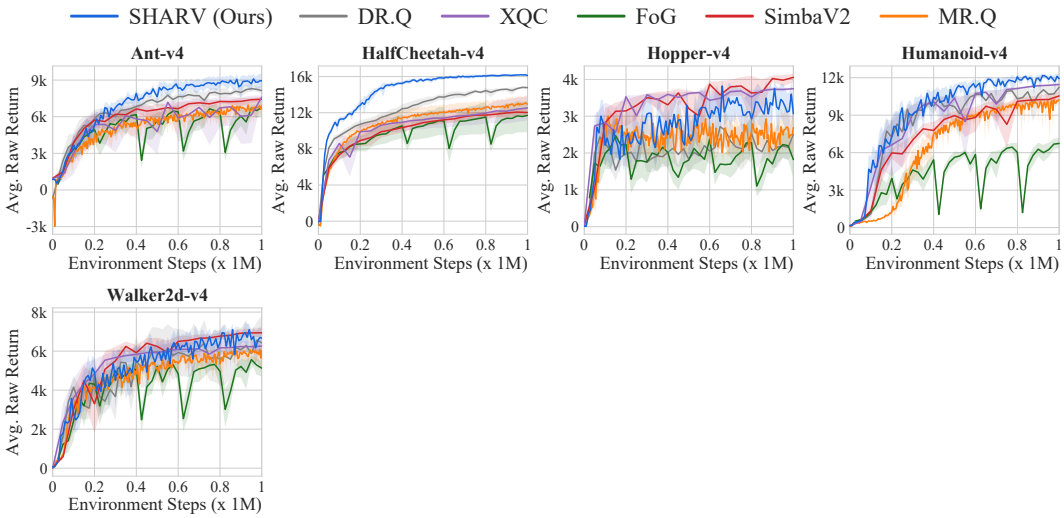

Figure 7: **Learning curves on Gym MuJoCo.** Five tasks are ordered alphabetically. Lines and shading indicate mean raw returns and pointwise 95% percentile-bootstrap CIs over ten seeds.

D.2 DMC-HARD

Table 9: **Full raw-return results on DMC-Hard.** Mean returns with 95% confidence intervals for 7 tasks at 500K decision steps (action repeat 2), equivalent to 1M environment steps.

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
Dog-Run	**812.5** [776.8, 848.2]	721.0 [684.2, 757.7]	645.4 [609.1, 681.6]	612.6 [577.0, 648.2]	562.5 [516.7, 608.2]	569.3 [544.5, 594.1]
Dog-Stand	**983.0** [978.9, 987.1]	972.4 [962.8, 982.1]	981.2 [976.4, 985.9]	975.6 [969.2, 982.0]	981.4 [977.4, 985.3]	967.6 [959.9, 975.2]
Dog-Trot	920.1 [901.9, 938.4]	**924.5** [913.5, 935.5]	909.9 [897.3, 922.5]	901.5 [891.5, 911.4]	861.8 [773.0, 950.5]	877.9 [847.8, 907.9]
Dog-Walk	**954.8** [948.1, 961.5]	949.8 [942.0, 957.7]	940.7 [929.3, 952.2]	920.8 [908.9, 932.7]	935.9 [927.6, 944.3]	916.2 [907.8, 924.6]
Humanoid-Run	**638.1** [621.0, 655.2]	464.9 [444.2, 485.6]	215.7 [195.9, 235.5]	292.4 [268.2, 316.6]	195.0 [182.3, 207.7]	200.4 [165.3, 235.6]
Humanoid-Stand	**962.6** [958.6, 966.6]	938.4 [932.4, 944.5]	906.5 [867.4, 945.5]	931.0 [920.8, 941.1]	916.1 [886.2, 945.9]	868.1 [823.9, 912.3]
Humanoid-Walk	**951.8** [948.4, 955.1]	925.1 [918.1, 932.1]	713.1 [636.3, 789.8]	878.0 [838.6, 917.4]	651.9 [590.7, 713.1]	662.2 [601.1, 723.3]

Table 10: **Normalized results on DMC-Hard.** 1K-normalized scores from Table 9 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
Dog-Run	**0.81** [0.78, 0.85]	0.72 [0.68, 0.76]	0.65 [0.61, 0.68]	0.61 [0.58, 0.65]	0.56 [0.52, 0.61]	0.57 [0.54, 0.59]
Dog-Stand	**0.98** [0.98, 0.99]	0.97 [0.96, 0.98]	**0.98** [0.98, 0.99]	**0.98** [0.97, 0.98]	**0.98** [0.98, 0.99]	0.97 [0.96, 0.98]
Dog-Trot	**0.92** [0.90, 0.94]	**0.92** [0.91, 0.94]	0.91 [0.90, 0.92]	0.90 [0.89, 0.91]	0.86 [0.77, 0.95]	0.88 [0.85, 0.91]
Dog-Walk	**0.95** [0.95, 0.96]	**0.95** [0.94, 0.96]	0.94 [0.93, 0.95]	0.92 [0.91, 0.93]	0.94 [0.93, 0.94]	0.92 [0.91, 0.92]
Humanoid-Run	**0.64** [0.62, 0.66]	0.46 [0.44, 0.49]	0.22 [0.20, 0.24]	0.29 [0.27, 0.32]	0.19 [0.18, 0.21]	0.20 [0.17, 0.24]
Humanoid-Stand	**0.96** [0.96, 0.97]	0.94 [0.93, 0.94]	0.91 [0.87, 0.95]	0.93 [0.92, 0.94]	0.92 [0.89, 0.95]	0.87 [0.82, 0.91]
Humanoid-Walk	**0.95** [0.95, 0.96]	0.93 [0.92, 0.93]	0.71 [0.64, 0.79]	0.88 [0.84, 0.92]	0.65 [0.59, 0.71]	0.66 [0.60, 0.72]
IQM	**0.94** [0.93, 0.94]	0.92 [0.91, 0.92]	0.84 [0.82, 0.86]	0.88 [0.87, 0.89]	0.81 [0.78, 0.83]	0.80 [0.78, 0.82]
Mean	**0.89** [0.88, 0.89]	0.84 [0.84, 0.85]	0.76 [0.75, 0.77]	0.79 [0.78, 0.80]	0.73 [0.71, 0.74]	0.72 [0.71, 0.74]

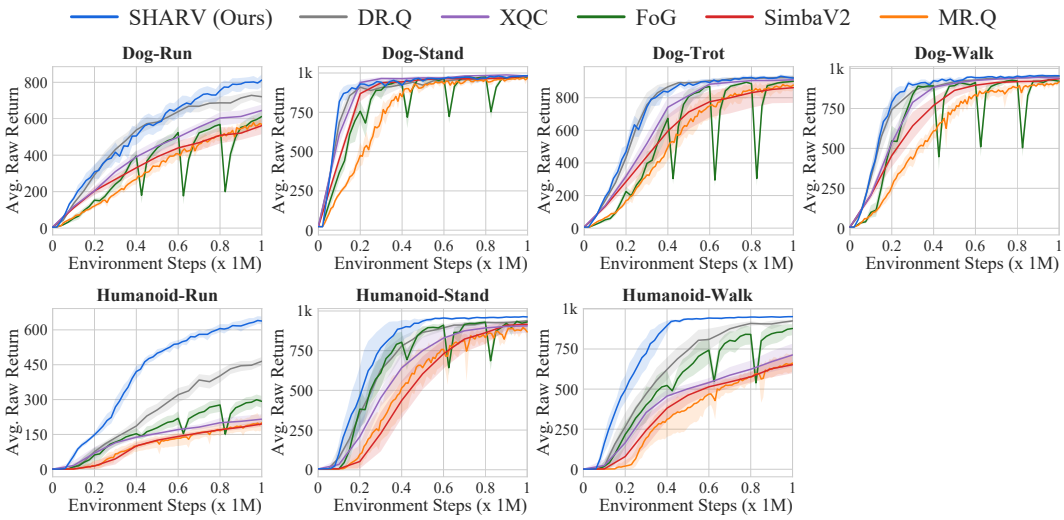

Figure 8: **Learning curves on DMC-Hard.** Seven tasks are ordered alphabetically. Lines and shading indicate mean raw returns and pointwise 95% percentile-bootstrap CIs over ten seeds.

D.3 HUMANOIDBENCH W/ HANDS

Table 11: **Full raw-return results on HumanoidBench w/ Hands.** Mean returns with 95% CIs for 12 tasks at 500K decision steps (action repeat 2), equivalent to 1M environment steps.

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
h1hand-basketball-v0	120.5 [61.7, 179.3]	74.9 [45.3, 104.5]	129.8 [73.1, 186.6]	**181.6** [131.1, 232.2]	56.2 [34.4, 78.0]	53.1 [34.0, 72.3]
h1hand-bookshelf_hard-v0	**659.6** [602.0, 717.3]	348.7 [262.2, 435.2]	632.4 [557.5, 707.4]	576.9 [548.5, 605.3]	496.0 [417.0, 574.9]	332.5 [240.1, 424.9]
h1hand-crawl-v0	650.8 [549.1, 752.5]	525.6 [477.3, 573.8]	778.2 [714.6, 841.8]	793.6 [721.3, 866.0]	640.2 [548.6, 731.8]	**806.6** [782.7, 830.5]
h1hand-door-v0	326.5 [317.0, 336.0]	320.5 [307.6, 333.3]	**334.8** [325.5, 344.0]	243.8 [227.1, 260.6]	310.0 [302.5, 317.6]	292.8 [280.3, 305.2]
h1hand-pole-v0	**916.2** [879.0, 953.5]	424.0 [299.2, 548.7]	460.4 [369.8, 551.1]	257.1 [237.5, 276.8]	492.6 [425.7, 559.5]	237.2 [201.5, 272.8]
h1hand-reach-v0	4351.6 [3731.8, 4971.4]	**4949.7** [4280.2, 5619.3]	3914.2 [3429.7, 4398.8]	2877.2 [2487.2, 3267.1]	3223.3 [2703.0, 3743.6]	4101.1 [3540.4, 4661.7]
h1hand-run-v0	**158.1** [131.3, 184.8]	128.9 [77.1, 180.7]	59.4 [51.3, 67.4]	22.3 [19.2, 25.3]	30.3 [22.4, 38.2]	35.2 [28.9, 41.4]
h1hand-sit_hard-v0	881.7 [853.0, 910.3]	**891.1** [840.9, 941.4]	883.2 [847.2, 919.2]	178.9 [128.4, 229.4]	723.6 [609.5, 837.8]	486.8 [352.7, 620.8]
h1hand-slide-v0	**438.1** [367.2, 508.9]	285.2 [258.2, 312.1]	226.9 [177.9, 276.0]	200.5 [173.3, 227.8]	136.1 [96.8, 175.3]	146.3 [131.3, 161.3]
h1hand-stair-v0	**391.1** [341.2, 441.1]	287.6 [192.9, 382.4]	160.7 [131.5, 189.9]	134.8 [125.5, 144.0]	120.1 [88.9, 151.3]	126.7 [103.7, 149.6]
h1hand-stand-v0	**948.6** [930.9, 966.3]	491.0 [344.2, 637.7]	311.6 [226.6, 396.7]	78.6 [66.1, 91.1]	103.5 [80.8, 126.1]	299.9 [194.5, 405.3]
h1hand-walk-v0	454.0 [320.2, 587.8]	**511.7** [371.4, 651.9]	375.7 [215.0, 536.5]	74.7 [62.6, 86.7]	64.2 [52.0, 76.4]	94.5 [76.7, 112.4]

Table 12: **Normalized results on HumanoidBench w/ Hands.** Success-normalized scores from Table 11 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV (Ours)**	DR.Q	XQC	FoG	SimbaV2	MR.Q
h1hand-basketball-v0	0.09 [0.04, 0.14]	0.06 [0.03, 0.08]	0.10 [0.05, 0.15]	**0.14** [0.10, 0.19]	0.04 [0.02, 0.06]	0.04 [0.02, 0.05]
h1hand-bookshelf_hard-v0	**0.32** [0.30, 0.35]	0.17 [0.12, 0.21]	0.31 [0.27, 0.35]	0.28 [0.27, 0.30]	0.24 [0.20, 0.28]	0.16 [0.11, 0.21]
h1hand-crawl-v0	0.88 [0.64, 1.12]	0.59 [0.47, 0.70]	1.19 [1.03, 1.34]	1.22 [1.05, 1.39]	0.86 [0.64, 1.08]	**1.25** [1.20, 1.31]
h1hand-door-v0	0.54 [0.53, 0.56]	0.53 [0.51, 0.55]	**0.56** [0.54, 0.57]	0.40 [0.38, 0.43]	0.51 [0.50, 0.53]	0.49 [0.46, 0.51]
h1hand-pole-v0	**1.32** [1.26, 1.37]	0.59 [0.41, 0.78]	0.65 [0.51, 0.78]	0.35 [0.32, 0.38]	0.70 [0.60, 0.79]	0.32 [0.27, 0.37]
h1hand-reach-v0	0.37 [0.31, 0.42]	**0.41** [0.36, 0.47]	0.33 [0.29, 0.37]	0.24 [0.21, 0.28]	0.27 [0.23, 0.31]	0.34 [0.30, 0.39]
h1hand-run-v0	**0.22** [0.19, 0.26]	0.18 [0.11, 0.26]	0.08 [0.07, 0.09]	0.03 [0.02, 0.03]	0.04 [0.03, 0.05]	0.05 [0.04, 0.06]
h1hand-sit_hard-v0	1.18 [1.14, 1.21]	**1.19** [1.12, 1.26]	1.18 [1.13, 1.23]	0.24 [0.17, 0.30]	0.96 [0.81, 1.12]	0.65 [0.47, 0.83]
h1hand-slide-v0	**0.62** [0.52, 0.73]	0.40 [0.37, 0.44]	0.32 [0.25, 0.39]	0.28 [0.24, 0.32]	0.19 [0.13, 0.25]	0.21 [0.18, 0.23]
h1hand-stair-v0	**0.56** [0.49, 0.63]	0.41 [0.27, 0.54]	0.23 [0.18, 0.27]	0.19 [0.18, 0.20]	0.17 [0.12, 0.21]	0.18 [0.14, 0.21]
h1hand-stand-v0	**1.19** [1.17, 1.21]	0.61 [0.42, 0.79]	0.38 [0.27, 0.49]	0.08 [0.07, 0.10]	0.12 [0.09, 0.14]	0.37 [0.23, 0.50]
h1hand-walk-v0	0.65 [0.46, 0.84]	**0.73** [0.53, 0.93]	0.54 [0.30, 0.77]	0.10 [0.09, 0.12]	0.09 [0.07, 0.11]	0.13 [0.11, 0.16]
IQM	**0.60** [0.55, 0.64]	0.43 [0.40, 0.47]	0.38 [0.35, 0.41]	0.23 [0.21, 0.24]	0.24 [0.22, 0.26]	0.25 [0.23, 0.27]
Mean	**0.66** [0.63, 0.69]	0.49 [0.46, 0.52]	0.49 [0.46, 0.52]	0.30 [0.28, 0.31]	0.35 [0.32, 0.37]	0.35 [0.33, 0.37]

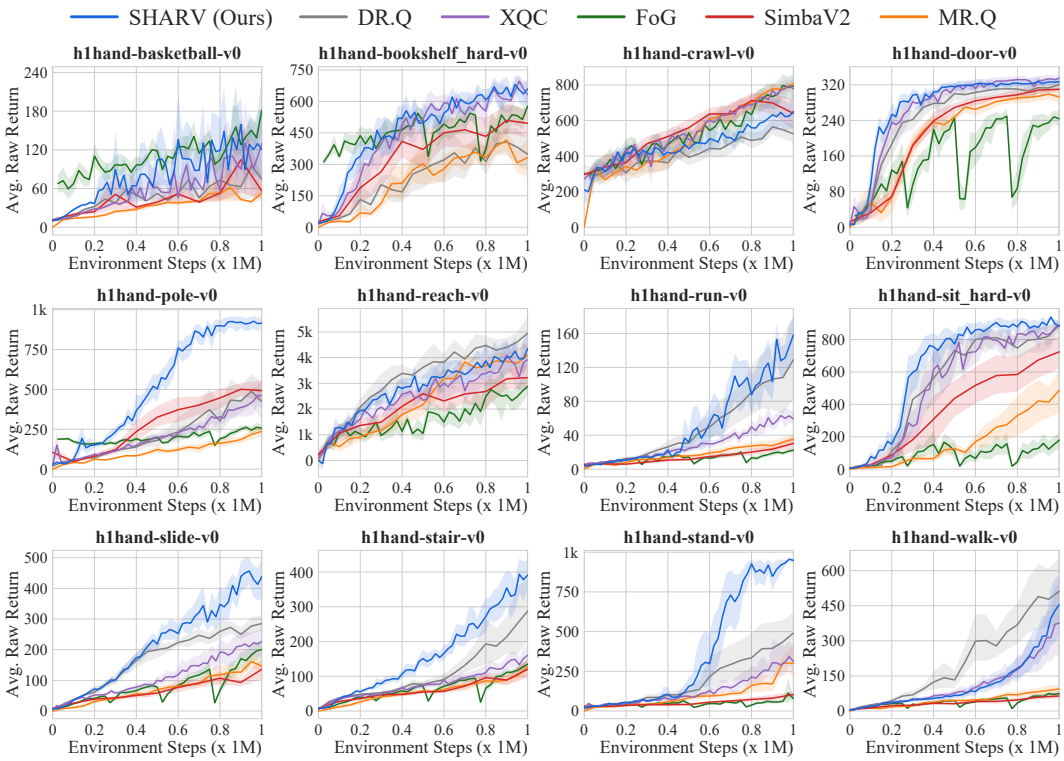

Figure 9: **Learning curves on HumanoidBench w/ Hands.** The 12 tasks are ordered alphabetically. Lines show mean raw returns and shading denotes confidence intervals. Means and pointwise 95% percentile-bootstrap confidence intervals are computed over ten seeds.

D.4 DMC-VISUAL

Table 13: **Full raw-return results on DMC-Visual.** Mean returns with 95% confidence intervals for 10 tasks at 500K decision steps (action repeat 2), equivalent to 1M environment steps.

Task	**SHARV (Ours)**	DR.Q	XQC	MR.Q
Acrobot-Swingup	365.6 [331.8, 399.4]	323.9 [283.3, 364.6]	**382.4** [363.7, 401.1]	287.1 [252.2, 322.1]
Cheetah-Run	761.2 [744.3, 778.2]	701.5 [680.0, 722.9]	724.1 [700.1, 748.1]	**775.6** [746.5, 804.8]
Finger-Turn_Easy	958.9 [931.6, 986.2]	**963.8** [943.3, 984.3]	666.3 [565.8, 766.8]	953.7 [928.8, 978.7]
Finger-Turn_Hard	**952.1** [927.3, 977.0]	937.6 [908.8, 966.4]	691.8 [587.2, 796.5]	932.9 [904.9, 961.0]
Hopper-Hop	**332.0** [297.7, 366.4]	329.8 [283.0, 376.7]	249.3 [213.2, 285.4]	270.3 [221.7, 318.9]
Humanoid-Run	**5.3** [3.7, 6.9]	1.3 [1.2, 1.4]	1.5 [1.2, 1.7]	1.4 [1.3, 1.6]
Humanoid-Stand	**9.8** [7.4, 12.1]	7.9 [7.4, 8.3]	7.7 [7.1, 8.4]	7.9 [7.3, 8.6]
Quadruped-Run	**709.0** [646.0, 772.0]	655.1 [572.8, 737.3]	397.5 [348.3, 446.7]	498.4 [473.6, 523.3]
Quadruped-Walk	875.6 [850.5, 900.8]	**927.5** [914.0, 941.0]	500.1 [397.8, 602.4]	833.9 [797.1, 870.8]
Walker-Run	736.5 [702.9, 770.1]	**745.9** [713.4, 778.4]	683.8 [672.1, 695.5]	615.9 [572.0, 659.9]

Table 14: **Normalized results on DMC-Visual.** 1K-normalized scores from Table 13 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV (Ours)**	DR.Q	XQC	MR.Q
Acrobot-Swingup	0.37 [0.33, 0.40]	0.32 [0.28, 0.36]	**0.38** [0.36, 0.40]	0.29 [0.25, 0.32]
Cheetah-Run	0.76 [0.74, 0.78]	0.70 [0.68, 0.72]	0.72 [0.70, 0.75]	**0.78** [0.75, 0.80]
Finger-Turn_Easy	**0.96** [0.93, 0.99]	**0.96** [0.94, 0.98]	0.67 [0.57, 0.77]	0.95 [0.93, 0.98]
Finger-Turn_Hard	**0.95** [0.93, 0.98]	0.94 [0.91, 0.97]	0.69 [0.59, 0.80]	0.93 [0.90, 0.96]
Hopper-Hop	**0.33** [0.30, 0.37]	**0.33** [0.28, 0.38]	0.25 [0.21, 0.29]	0.27 [0.22, 0.32]
Humanoid-Run	**0.01** [0.00, 0.01]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]
Humanoid-Stand	**0.01** [0.01, 0.01]	**0.01** [0.01, 0.01]	**0.01** [0.01, 0.01]	**0.01** [0.01, 0.01]
Quadruped-Run	**0.71** [0.65, 0.77]	0.66 [0.57, 0.74]	0.40 [0.35, 0.45]	0.50 [0.47, 0.52]
Quadruped-Walk	0.88 [0.85, 0.90]	**0.93** [0.91, 0.94]	0.50 [0.40, 0.60]	0.83 [0.80, 0.87]
Walker-Run	0.74 [0.70, 0.77]	**0.75** [0.71, 0.78]	0.68 [0.67, 0.70]	0.62 [0.57, 0.66]
IQM	**0.64** [0.62, 0.65]	0.61 [0.59, 0.63]	0.45 [0.43, 0.48]	0.55 [0.53, 0.56]
Mean	**0.57** [0.56, 0.58]	0.56 [0.55, 0.57]	0.43 [0.41, 0.45]	0.52 [0.51, 0.53]

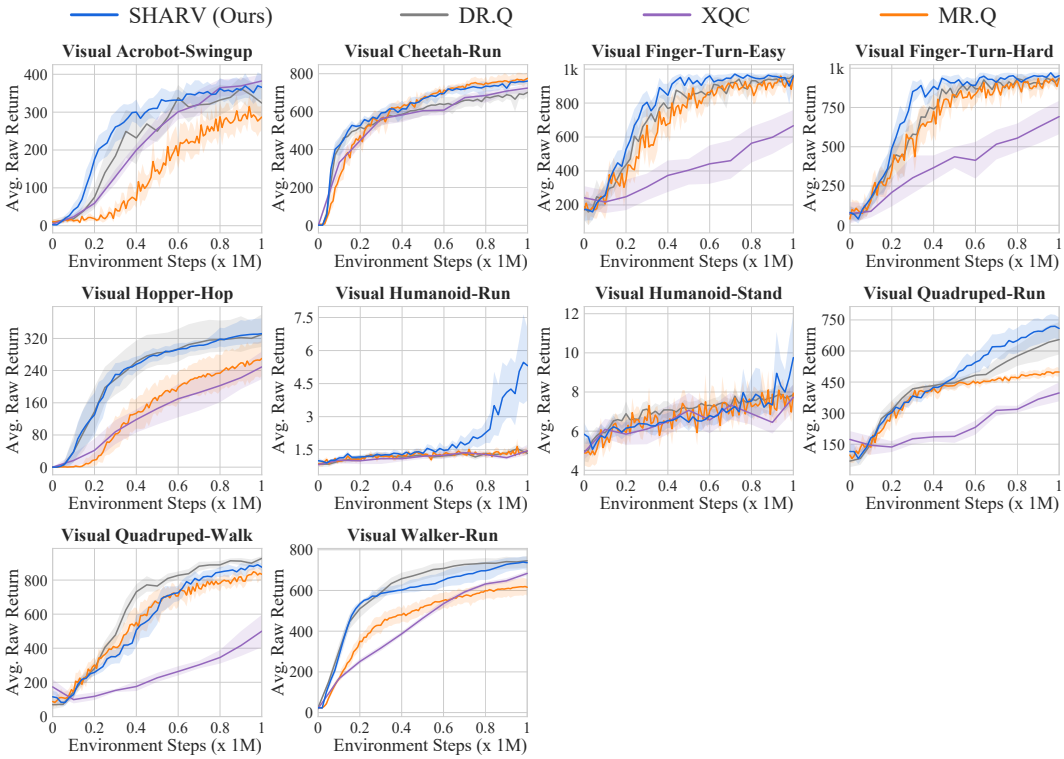

Figure 10: **Learning curves on DMC-Visual.** The 10 tasks are ordered alphabetically. Lines show mean raw returns and shading denotes confidence intervals. Means and pointwise 95% percentile-bootstrap confidence intervals are computed over ten seeds.

E Training Time

We measure training time per step on `humanoid-run` from DMC-Hard using an Intel Xeon E5-2690 v4 CPU and an NVIDIA Tesla V100 GPU. As shown in Table 15, SHARV requires 0.030 seconds per step, compared with 0.020 seconds for DR.Q and 0.015 seconds for XQC.

Table 15: Training time per step on `humanoid-run` from DMC-Hard.

Method	SHARV	DR.Q	XQC
Time (s/step)	0.030	0.020	0.015

F Full Ablations

We report full results for SHARV, w/o DDR (§ 4.1), and w/o MSB (§ 4.2), using the same benchmarks, seed count, and evaluation settings as the main comparison.

F.1 Gym MuJoCo

Table 16: **Full raw-return ablation results on Gym MuJoCo.** Mean returns with 95% confidence intervals for 5 tasks at 1M time steps (action repeat 1).

Task	**SHARV**	w/o MSB	w/o DDR
Ant-v4	**8950.5** [8360.1, 9540.8]	8715.2 [8256.5, 9173.8]	8385.0 [8073.7, 8696.2]
HalfCheetah-v4	**16119.8** [15996.7, 16242.8]	15509.7 [15401.5, 15617.9]	15687.5 [15287.6, 16087.4]
Hopper-v4	**3068.3** [2606.0, 3530.6]	2538.3 [2166.5, 2910.0]	2791.3 [2280.6, 3301.9]
Humanoid-v4	**11922.8** [11547.9, 12297.6]	11498.3 [10104.4, 12892.2]	10580.4 [9029.7, 12131.1]
Walker2d-v4	**6664.6** [5944.2, 7385.0]	6137.6 [5747.5, 6527.6]	6615.2 [6140.2, 7090.2]

Table 17: **Normalized ablation results on Gym MuJoCo.** TD3-normalized scores from Table 16 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV**	w/o MSB	w/o DDR
Ant-v4	**2.25** [2.10, 2.40]	2.19 [2.08, 2.30]	2.11 [2.03, 2.18]
HalfCheetah-v4	**1.51** [1.50, 1.52]	1.45 [1.44, 1.46]	1.47 [1.43, 1.51]
Hopper-v4	**0.95** [0.81, 1.09]	0.79 [0.67, 0.90]	0.86 [0.71, 1.02]
Humanoid-v4	**2.34** [2.27, 2.41]	2.26 [1.98, 2.53]	2.07 [1.77, 2.38]
Walker2d-v4	**1.69** [1.51, 1.87]	1.56 [1.46, 1.65]	1.68 [1.56, 1.80]
IQM	**1.79** [1.74, 1.84]	1.67 [1.59, 1.73]	1.70 [1.61, 1.77]
Mean	**1.75** [1.69, 1.80]	1.65 [1.58, 1.71]	1.64 [1.56, 1.70]

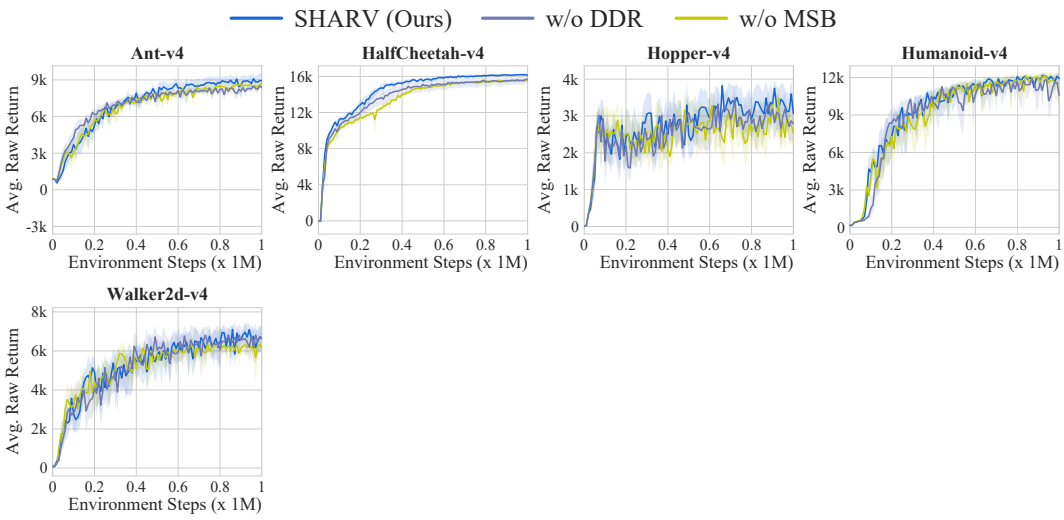

Figure 11: **Ablation learning curves on Gym MuJoCo.** Five tasks are ordered alphabetically. Lines and shading indicate mean raw returns and pointwise 95% percentile-bootstrap CIs.

F.2 DMC-HARD

Table 18: **Full raw-return ablation results on DMC-Hard.** Mean returns with 95% confidence intervals for 7 tasks at 500K decision steps (action repeat 2), equivalent to 1M environment steps.

Task	**SHARV**	w/o MSB	w/o DDR
Dog-Run	**812.5** [776.8, 848.2]	708.6 [669.0, 748.2]	799.2 [771.6, 826.9]
Dog-Stand	983.0 [978.9, 987.1]	979.1 [971.6, 986.6]	**985.0** [978.7, 991.3]
Dog-Trot	920.1 [901.9, 938.4]	912.5 [879.6, 945.4]	**929.4** [920.8, 938.0]
Dog-Walk	**954.8** [948.1, 961.5]	941.1 [923.3, 958.9]	943.3 [925.6, 961.1]
Humanoid-Run	**638.1** [621.0, 655.2]	494.7 [414.6, 574.9]	534.5 [512.0, 556.9]
Humanoid-Stand	**962.6** [958.6, 966.6]	958.6 [953.9, 963.3]	958.9 [953.3, 964.5]
Humanoid-Walk	**951.8** [948.4, 955.1]	936.3 [933.1, 939.5]	950.0 [944.9, 955.1]

Table 19: **Normalized ablation results on DMC-Hard.** 1K-normalized scores from Table 18 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV**	w/o MSB	w/o DDR
Dog-Run	**0.81** [0.78, 0.85]	0.71 [0.67, 0.75]	0.80 [0.77, 0.83]
Dog-Stand	**0.98** [0.98, 0.99]	**0.98** [0.97, 0.99]	**0.98** [0.98, 0.99]
Dog-Trot	0.92 [0.90, 0.94]	0.91 [0.88, 0.95]	**0.93** [0.92, 0.94]
Dog-Walk	**0.95** [0.95, 0.96]	0.94 [0.92, 0.96]	0.94 [0.93, 0.96]
Humanoid-Run	**0.64** [0.62, 0.66]	0.49 [0.41, 0.57]	0.53 [0.51, 0.56]
Humanoid-Stand	**0.96** [0.96, 0.97]	**0.96** [0.95, 0.96]	**0.96** [0.95, 0.96]
Humanoid-Walk	**0.95** [0.95, 0.96]	0.94 [0.93, 0.94]	**0.95** [0.94, 0.96]
IQM	**0.94** [0.93, 0.94]	0.92 [0.91, 0.93]	0.93 [0.93, 0.94]
Mean	**0.89** [0.88, 0.89]	0.85 [0.83, 0.86]	0.87 [0.87, 0.88]

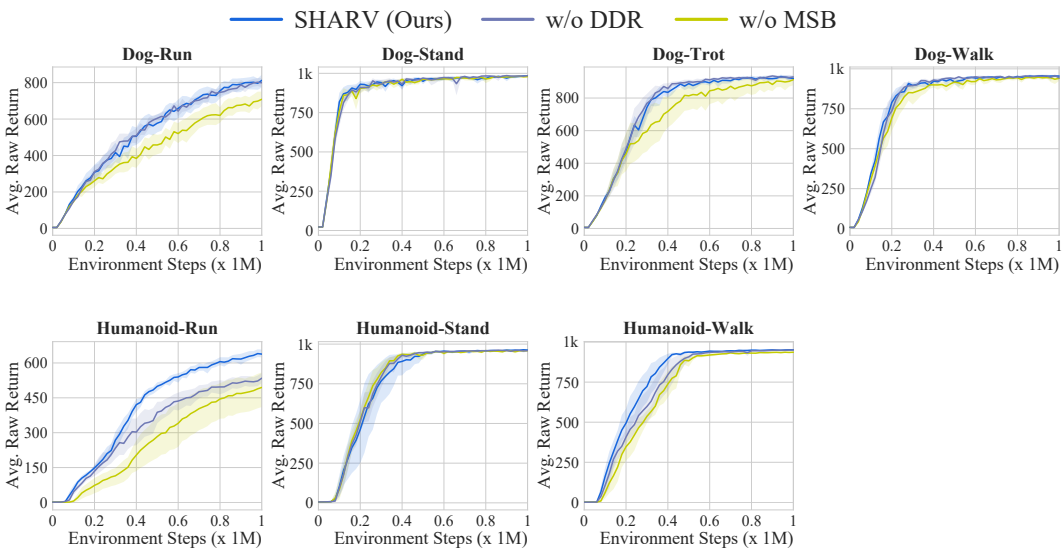

Figure 12: **Ablation learning curves on DMC-Hard.** Seven tasks are ordered alphabetically. Lines and shading indicate mean raw returns and pointwise 95% percentile-bootstrap CIs over ten seeds.

F.3 HUMANOIDBENCH W/ HANDS

Table 20: **Full raw-return ablation results on HumanoidBench w/ Hands.** Mean returns with 95% CIs for 12 tasks at 500K decision steps (action repeat 2), equivalent to 1M environment steps.

Task	**SHARV**	w/o MSB	w/o DDR
h1hand-basketball-v0	**120.5** [61.7, 179.3]	89.8 [48.7, 130.8]	111.8 [52.4, 171.1]
h1hand-bookshelf_hard-v0	**659.6** [602.0, 717.3]	575.2 [497.0, 653.5]	609.6 [519.6, 699.6]
h1hand-crawl-v0	650.8 [549.1, 752.5]	621.7 [565.2, 678.3]	**721.3** [636.6, 806.0]
h1hand-door-v0	326.5 [317.0, 336.0]	**346.1** [288.8, 403.4]	332.4 [321.9, 342.9]
h1hand-pole-v0	916.2 [879.0, 953.5]	847.7 [817.4, 878.0]	**935.9** [898.9, 972.9]
h1hand-reach-v0	**4351.6** [3731.8, 4971.4]	4039.3 [3633.7, 4445.0]	4012.5 [3528.8, 4496.3]
h1hand-run-v0	**158.1** [131.3, 184.8]	155.2 [135.9, 174.4]	90.8 [62.5, 119.2]
h1hand-sit_hard-v0	881.7 [853.0, 910.3]	588.3 [449.4, 727.2]	**898.6** [859.8, 937.5]
h1hand-slide-v0	**438.1** [367.2, 508.9]	344.1 [294.8, 393.5]	421.7 [384.2, 459.2]
h1hand-stair-v0	**391.1** [341.2, 441.1]	381.4 [315.3, 447.5]	300.9 [279.1, 322.7]
h1hand-stand-v0	**948.6** [930.9, 966.3]	885.6 [802.3, 968.8]	752.0 [607.8, 896.2]
h1hand-walk-v0	**454.0** [320.2, 587.8]	266.1 [210.7, 321.6]	286.4 [168.0, 404.7]

Table 21: **Normalized results on HumanoidBench w/ Hands.** Success-normalized scores from Table 20 and aggregate IQM and mean, with 95% CIs.

Task	**SHARV**	w/o MSB	w/o DDR
h1hand-basketball-v0	**0.09** [0.04, 0.14]	0.07 [0.03, 0.10]	**0.09** [0.04, 0.14]
h1hand-bookshelf_hard-v0	**0.32** [0.30, 0.35]	0.28 [0.24, 0.32]	0.30 [0.25, 0.34]
h1hand-crawl-v0	0.88 [0.64, 1.12]	0.81 [0.68, 0.95]	**1.05** [0.85, 1.25]
h1hand-door-v0	0.54 [0.53, 0.56]	**0.57** [0.48, 0.67]	0.55 [0.53, 0.57]
h1hand-pole-v0	1.32 [1.26, 1.37]	1.22 [1.17, 1.26]	**1.35** [1.29, 1.40]
h1hand-reach-v0	**0.37** [0.31, 0.42]	0.34 [0.31, 0.37]	0.34 [0.30, 0.38]
h1hand-run-v0	**0.22** [0.19, 0.26]	**0.22** [0.19, 0.25]	0.13 [0.09, 0.17]
h1hand-sit_hard-v0	1.18 [1.14, 1.21]	0.78 [0.60, 0.97]	**1.20** [1.15, 1.25]
h1hand-slide-v0	**0.62** [0.52, 0.73]	0.49 [0.42, 0.56]	0.60 [0.55, 0.65]
h1hand-stair-v0	**0.56** [0.49, 0.63]	0.54 [0.45, 0.64]	0.43 [0.40, 0.46]
h1hand-stand-v0	**1.19** [1.17, 1.21]	1.11 [1.00, 1.21]	0.94 [0.76, 1.12]
h1hand-walk-v0	**0.65** [0.46, 0.84]	0.38 [0.30, 0.46]	0.41 [0.24, 0.58]
IQM	**0.60** [0.55, 0.64]	0.49 [0.46, 0.52]	0.53 [0.49, 0.57]
Mean	**0.66** [0.63, 0.69]	0.57 [0.54, 0.59]	0.61 [0.59, 0.64]

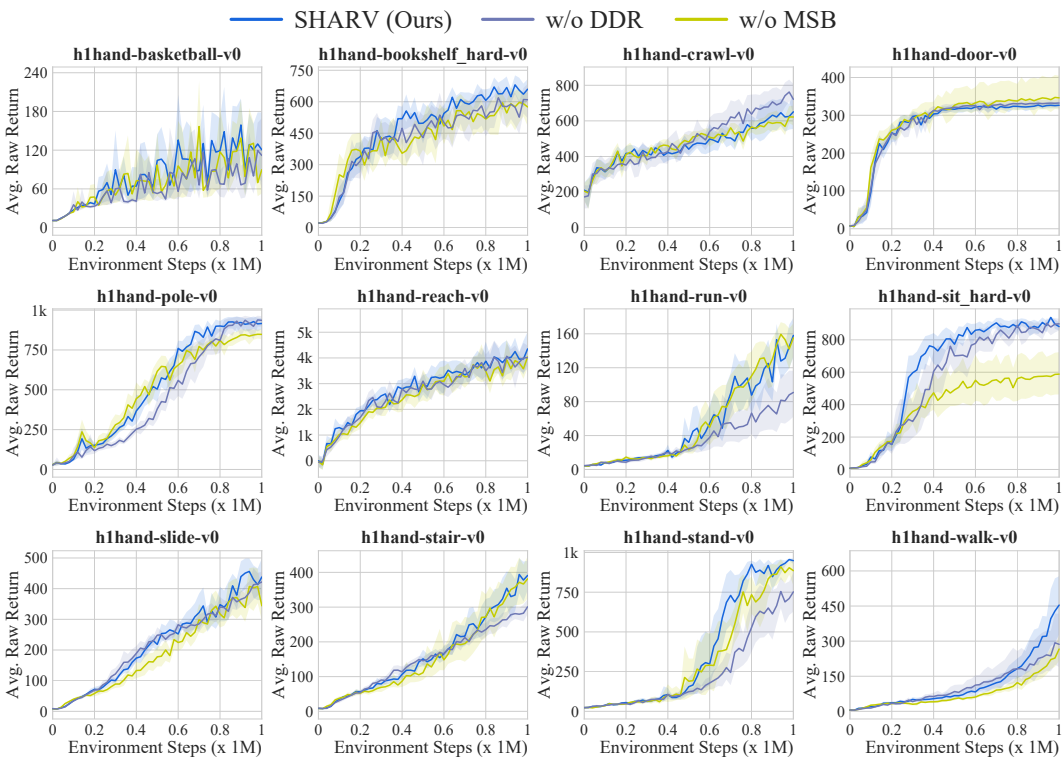

Figure 13: **Ablation learning curves on HumanoidBench w/ Hands.** The 12 tasks are ordered alphabetically. Lines show mean raw returns and shading denotes confidence intervals. Means and pointwise 95% percentile-bootstrap confidence intervals are computed over ten seeds.