DRAVS: DIRECTIONAL VISUAL REWARD LEARNING WITH ADAPTIVE VIEW SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models provide visual supervision for reinforcement learning, reducing reliance on manually designed rewards. Multi-view videos further alleviate occlusion by revealing task-relevant motion and contacts that may be hidden from a single viewpoint. Existing video-supervised state-wise reward formulations directly add learned state scores to task rewards. However, policies trained with such feedback can exhibit late-stage performance degradation, losing part of their earlier gains. Because the immediate visual term depends only on the current state, it does not explicitly distinguish progress from regression across transitions. Moreover, uniform random sampling of supervision views treats cameras equally despite differences in the visibility of task-relevant behavior. To address these limitations, we propose *Directional Visual Reward Learning with Adaptive View Selection* (DRAVS). First, Direction-Aware Visual Reward replaces state-wise addition with consecutive-state score differences computed using a scorer held fixed within each reinforcement-learning phase, providing direct feedback on progress and regression without additional vision-language model queries. Second, Budgeted Relative View Selection balances exploitation and exploration through sparse cross-view comparisons to adaptively select supervision views, with at most 5% additional scored clips over one-view-per-segment supervision. Compared with the state-of-the-art Multi-view Video Reward method, DRAVS raises mean final success rate on the ten-task Continual World benchmark built on MetaWorld from 0.54 to 0.69 and mean final return across nine Humanoid-Bench tasks from 616.9 to 836.3. Learning curves further show that DRAVS mitigates late-stage performance degradation and better preserves earlier gains. Additional qualitative visual comparisons are available at the anonymous project page: https://anon-visual-research.github.io/DRAVS/.

1 INTRODUCTION

Reinforcement learning enables agents to acquire manipulation and locomotion skills through interaction with their environments, guided by rewards that describe the desired behavior. Task rewards can specify objectives such as reaching a target position or maintaining a forward speed, but capturing coordinated motion and appropriate contact often requires additional, carefully designed rewards. Designing these rewards for individual tasks is difficult and time-consuming. Visual observations offer complementary guidance by revealing behavioral details that are difficult to express through hand-crafted rewards. Vision-language models make it possible to use the alignment between visual observations and task descriptions as a source of reward (Rocamonde et al., 2024; Ma et al., 2023a). Videos capture how behavior unfolds over time (Sontakke et al., 2023), while multiple viewpoints provide complementary evidence by revealing limbs, object motion, and contacts that may be obscured from a single view (Luo et al., 2026).

Multi-View Video Reward Shaping (MVR) brings this supervision into online reinforcement learning (Luo et al., 2026). It renders execution videos from uniformly sampled cameras, with one view per scored segment. Video–text scores provide pairwise preference supervision across segments for a state-relevance model. Comparing each state's learned relevance with that of retained reference states yields a state visual score that is added to the task reward. This pipeline transfers video-level supervision to individual states and involves two connected choices: which visual evidence trains the state scorer, and how the resulting scores become rewards for policy learning.

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.

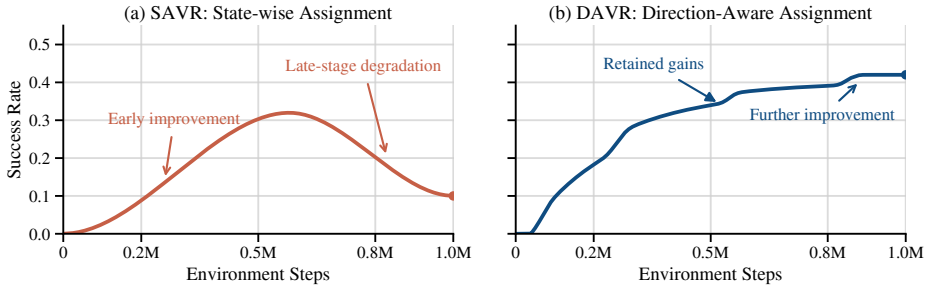


Figure 1: **Training dynamics under different reward assignments.** Training curves on *push*. Both panels use uniform view sampling; only the reward assignment differs. Panel (a) shows early improvement followed by late-stage degradation with SAVR. Panel (b) shows that the proposed DAVR retains earlier gains and achieves further improvement as training continues.

In our training experiments, however, we observe late-stage performance degradation on some tasks: success initially improves but later declines, leaving the final policy below its earlier performance. Figure 1 (a) illustrates this pattern on *push*. This observation motivates us to examine how learned state visual scores are assigned as rewards during policy learning.

We call the direct addition of the current state's visual score to the task reward *State-wise Additive Visual Reward* (SAVR). Its immediate visual term depends only on the current state, so actions from that state receive the same visual term even when they lead to higher-scoring or lower-scoring next states. It therefore does not explicitly encode the score change resulting from the transition. To provide this feedback directly, we introduce *Direction-Aware Visual Reward* (DAVR), which replaces the current-state visual term with the score difference between consecutive states. Let $r^{\mathrm{VLM}}(s)$ denote the existing state visual score and $w > 0$ its reward coefficient. The two assignments are

$$r_t^{\mathrm{SAVR}} = r_{\mathrm{task},t} + w\, r^{\mathrm{VLM}}(s_t), \quad r_t^{\mathrm{DAVR}} = r_{\mathrm{task},t} + w\big[r^{\mathrm{VLM}}(s_{t+1}) - r^{\mathrm{VLM}}(s_t)\big]. \tag{1}$$

Both forms retain the original task reward. DAVR evaluates consecutive states with the same phase-frozen scorer, giving positive, negative, or zero visual feedback when the score increases, decreases, or remains unchanged. This explicitly distinguishes progress, regression, and stagnation in the visual assessment. DAVR reuses the existing state scorer without training another reward model or adding vision-language model queries. Under the same uniform view sampling, DAVR retains earlier gains and achieves further improvement on *push*, as shown in Fig. 1(b).

Beyond reward assignment, the learned scorer depends on its visual supervision. Camera viewpoint affects the visibility of task-relevant motion and contacts, so the same behavior can receive different video–text scores. When preference supervision compares segments from different cameras, score differences may reflect both behavioral variation and viewpoint-dependent visibility. MVR mitigates viewpoint effects through state-representation regularization, but samples supervision views uniformly without adapting to accumulated alignment evidence (Luo et al., 2026). Scoring all V views would allow selecting the highest-scoring view for each segment, but requires V clips rather than one, yielding four times the clip count with four cameras.

To adapt supervision-view selection within a limited scoring budget, we introduce *Budgeted Relative View Selection* (BRVS). For a small subset of stored segments, BRVS scores one additional view and uses the video–text score difference to update persistent camera preferences. Comparing views of the same execution holds behavior fixed, providing relative alignment evidence for view selection. To allocate these limited comparisons effectively, a two-stage upper confidence bound (UCB) rule balances exploitation of accumulated alignment evidence with exploration of under-compared cameras and camera pairs. This uses current preferences to guide comparisons while preserving opportunities to reassess alternatives. The resulting preferences are reused across update rounds to guide subsequent primary-view selections, extending the benefit of each comparison to later segments that require only one scored clip. Only the preselected primary clip is retained for scorer supervision. Relative to scoring one view per segment, BRVS adds at most 5% scored view clips and requires no additional environment interaction.

We combine these designs in *Directional Visual Reward Learning with Adaptive View Selection* (DRAVS). BRVS selects the videos that supervise the state scorer; DAVR converts the learned

scores into feedback on transition-level changes. Through this shared scorer, DRAVS connects task-adaptive supervision with directional reward feedback within a limited additional scoring budget.

We evaluate DRAVS on the ten-task Continual World (CW10) benchmark built on MetaWorld (Yu et al., 2020; Wołczyk et al., 2021) and nine Unitree H1 tasks from HumanoidBench (Sferrazza et al., 2024) under MVR's benchmark protocol. On the ten-task Continual World benchmark, it raises mean final success from the published MVR result of 0.54 to 0.69. On Hard-5, the five tasks with published MVR success at most 0.50, the mean increases from 0.24 to 0.44. On all three tasks where MVR achieves final success of 1.00, DRAVS also achieves 1.00. On the nine HumanoidBench tasks, DRAVS improves mean final return on every task, increasing the overall mean from 616.9 to 836.3. Hard-5 factorial ablations under matched settings further show that DAVR improves mean final success under both view-selection settings, while BRVS improves it under both reward assignments. Our contributions are threefold:

(i) We propose DAVR, which converts consecutive-state score changes into signed visual feedback that explicitly captures progress, regression, and stagnation. It reuses the existing state scorer without additional vision-language model queries, and alleviates the late-stage degradation observed with state-based visual rewards, leading to improved final policy performance.

(ii) We propose BRVS, a two-stage UCB-inspired adaptive view-selection strategy that learns persistent camera preferences from sparse same-segment comparisons while balancing exploration and exploitation. The learned preferences guide single-view supervision for the state-relevance model and consistently improve both SAVR and DAVR, with no more than 5% additional scored clips compared with scoring one view per segment.

(iii) We evaluate DRAVS on the ten-task Continual World benchmark and nine HumanoidBench tasks, where it consistently outperforms the state-of-the-art multi-view video reward method MVR. Matched factorial ablations on Hard-5 further isolate the contributions of DAVR and BRVS, showing that both components independently improve performance and are complementary when combined.

2 Related Work

Visual and video reward learning. Pretrained vision-language models can provide zero-shot visual rewards through vision–language alignment (Rocamonde et al., 2024). Other approaches learn rewards from pairwise preferences generated by vision-language models (Wang et al., 2024b), fine-tuned vision-language representations (Fu et al., 2024), goal-image distances (Ma et al., 2023b), or joint language-image value representation (Ma et al., 2023a). Video-based methods further exploit trajectory similarity, temporal ranking, or subtask structure in demonstrations (Sontakke et al., 2023; Yang et al., 2024; Alakuijala et al., 2025; Kim et al., 2025). MVR learns state relevance from multi-view videos scored by a vision-language model and constructs visual scores relative to retained reference states (Luo et al., 2026). Its pipeline samples supervision views uniformly and adds the source-state score to the task reward. DRAVS builds on this state-scoring pipeline but changes how supervision views are selected and how state scores are converted into transition-level feedback: BRVS uses relative evidence to guide view selection, while DAVR converts consecutive state-score changes into signed transition feedback.

Temporal reward assignment. Potential-based shaping preserves optimal policies through discounted potential differences under standard assumptions (Ng et al., 1999), with dynamic extensions using time-indexed potentials (Devlin & Kudenko, 2012). Other approaches adapt shaping weights (Hu et al., 2020), shorten effective planning horizons with heuristics (Cheng et al., 2021), or learn exploration-guided intrinsic rewards (Devidze et al., 2022). Visual and temporal reward methods use different mechanisms to estimate progress or derive temporally structured learning signals. VIP derives dense goal-conditioned rewards from distances in a value-implicit visual representation (Ma et al., 2023b); T^2-VLM tracks subgoal completion inferred by a vision-language model with Bayesian filtering (Zhao et al., 2025); ARM uses tri-state progress labels for advantage-weighted behavior cloning (Mao et al., 2026); GVL formulates value estimation as temporal ordering over shuffled video frames (Ma et al., 2025); and PROGRESSOR learns and online-refines a progress estimator conditioned on current, initial, and goal observations (Ayalew et al., 2025). TimeRewarder also provides step-level reward signals from temporally adjacent observations, but it learns a separate temporal-distance predictor from passive videos (Liu et al., 2026). In contrast, DAVR derives

transition-level feedback directly from consecutive outputs of MVR's existing reference-conditioned state scorer, without introducing a separate progress model.

View selection for visual supervision. Active vision controls observations alongside task actions (Cheng et al., 2018; Shang & Ryoo, 2023), while multi-view perception selects views for recognition and detection (Hou et al., 2024). BRVS instead focuses on selecting supervision views for an online state scorer while leaving policy observations unchanged. MVR samples supervision views uniformly (Luo et al., 2026); BRVS instead learns persistent view preferences from sparse relative comparisons, balancing exploration and exploitation to guide subsequent supervision-view selection.

3 Proposed Method

Built on MVR's state-scoring pipeline (Luo et al., 2026), DRAVS links visual supervision and policy rewards through a learned state scorer. Budgeted Relative View Selection (BRVS) uses sparse same-segment comparisons to learn persistent camera preferences that guide which views supervise MVR's state-relevance model. MVR translates this video-level supervision into reference-conditioned state scores. Direction-Aware Visual Reward (DAVR) then uses consecutive-state score differences to provide directional transition feedback while retaining the task reward. Figure 2 summarizes this data flow. Panel (a) shows how candidate views and a stored execution segment enter BRVS: routine decisions exploit persistent preferences, while sparse comparisons produce relative evidence that updates those preferences. Only the selected primary clip supplies MVR scorer supervision. Panel (b) shows the resulting periodic MVR update and the shared phase-frozen scorer used by DAVR to evaluate consecutive states, form a signed score difference, and support policy learning.

3.1 Preliminaries: MVR State Score

We follow MVR in learning a state-relevance model from video-level supervision, since policy rewards require scores for individual states.

For task m with language description ℓ_m, MVR renders a fixed-length state sequence $\mathbf{s}_n = (s_{n,1}, \ldots, s_{n,L})$ from a selected camera and scores the resulting video $\mathbf{o}_n$ with a frozen vision-language model, $\psi^{\mathrm{VLM}}(\mathbf{o}_n, \ell_m)$. Each training record stores the state sequence, score, and video embedding.

At update round k, the state-relevance model $f^{\mathrm{MVR}}_{m,k}$ maps an environment state to a scalar. MVR matches pairwise preference probabilities from sequence-averaged relevance to those from video–text scores, using training records that may represent different segments and cameras. It also regularizes state-sequence representations with the similarity structure of video embeddings to mitigate viewpoint effects (Luo et al., 2026). We retain both terms; BRVS changes each record's supervision camera, while the model input remains the environment state.

To evaluate relevance relative to retained high-scoring experience, MVR compares each state with states from its reference sequences. Let $\mu^{\mathrm{ref}}_{m,k}$ denote the empirical distribution over individual states in these sequences, retained online. The resulting state score averages the log-preference scores against these references:

$$r^{\mathrm{VLM}}_{m,k}(s) = \mathbb{E}_{\bar{s}\sim\mu^{\mathrm{ref}}_{m,k}}\left[\log\sigma\big(f^{\mathrm{MVR}}_{m,k}(s) - f^{\mathrm{MVR}}_{m,k}(\bar{s})\big)\right], \qquad \sigma(x) = \frac{1}{1+\exp(-x)}. \tag{2}$$

Higher scores indicate greater relevance relative to the reference states. Update round k produces the model and reference distribution that jointly define the scorer; both remain fixed throughout the following reinforcement-learning phase.

Let $r_{\mathrm{task},t} = r_{\mathrm{task}}(s_t, a_t, s_{t+1})$ be the stored environment reward and $0 < \gamma < 1$ the discount factor used for policy learning. MVR adds the source-state score to each transition with a run-fixed coefficient $w_m > 0$, giving the SAVR reward formulation introduced in Eq. (1):

$$r^{\mathrm{SAVR}}_{t,m,k} = r_{\mathrm{task},t} + w_m\, r^{\mathrm{VLM}}_{m,k}(s_t). \tag{3}$$

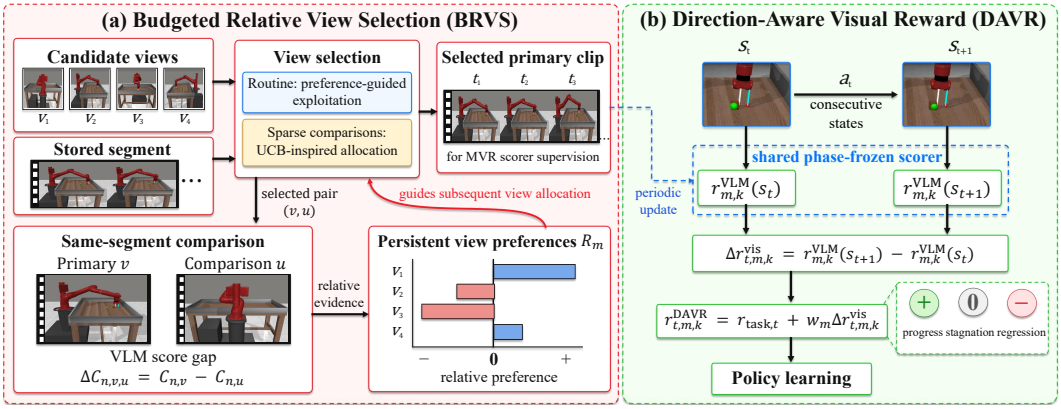


Figure 2: **Framework overview.** (a) Budgeted Relative View Selection (BRVS) receives candidate views and a stored execution segment. Routine decisions use preference-guided exploitation, while sparse UCB-inspired comparisons produce same-segment video–text evidence, update persistent view preferences, and guide subsequent view allocation. The selected primary clip provides MVR scorer supervision. (b) Periodic MVR updates provide a shared phase-frozen scorer. DAVR evaluates consecutive states with this scorer, converts the endpoint-score difference into signed visual feedback, adds it to the task reward, and performs policy learning.

3.2 Budgeted Relative View Selection

Viewpoint-dependent occlusion affects how clearly motion and contacts are visible, changing video–text scores even for the same behavior. BRVS complements MVR's representation regularization by adapting supervision-view selection to accumulated alignment evidence. Rather than exhaustively scoring all V views per segment, sparse same-sequence comparisons learn persistent camera preferences that guide subsequent single-view supervision.

Let $\mathcal{V} = \{1, \ldots, V\}$, $V \geq 2$, denote the fixed candidate cameras. Each camera decision corresponds to one stored execution segment represented by a fixed-length state sequence. A comparison decision scores a primary view and one additional comparison view, while a routine decision scores only its primary view. As shown in panel (a) of Fig. 2, candidate views and the stored segment first enter the view-selection rule. Routine decisions follow preference-guided exploitation, whereas sparse comparison decisions select a primary–comparison pair before scoring; the resulting same-segment evidence updates persistent preferences and feeds subsequent view allocation.

Relative evidence from same-sequence comparisons. Score gaps between different segments can reflect both behavior and viewpoint, so they cannot be attributed directly to camera choice. BRVS compares two views of the same stored state sequence, holding the executed behavior fixed to obtain relative view evidence. These same-sequence comparisons update camera preferences, whereas MVR's cross-record comparisons continue to train the state-relevance model. For sequence n, the video–text alignment score from camera v is

$$C_{n,v} = \psi^{\mathrm{VLM}}(\mathbf{o}_{n,v}, \ell_m), \tag{4}$$

where $\mathbf{o}_{n,v}$ is the rendered clip and ℓ_m is the fixed task text.

Persistent preferences and symmetric updates. Each task maintains camera preferences $R_{m,v}$ as online statistics for view selection. They start at zero and persist across collection batches and scorer updates, accumulating relative evidence for subsequent choices. We fit preference differences to the observed same-sequence alignment gaps. For a queried primary–comparison pair (v, u), the gap and its discrepancy from the current preference difference are

$$\Delta C_{n,v,u} = C_{n,v} - C_{n,u}, \qquad e_{n,v,u} = \Delta C_{n,v,u} - (R_{m,v} - R_{m,u}). \tag{5}$$

We split the correction equally between the two cameras with opposite signs, moving their preference gap toward the observed gap with step size $\eta \in (0, 1]$:

$$R_{m,v} \leftarrow R_{m,v} + \frac{\eta}{2} e_{n,v,u}, \qquad R_{m,u} \leftarrow R_{m,u} - \frac{\eta}{2} e_{n,v,u}. \tag{6}$$

The constant step size keeps these preferences responsive to new evidence as the policy-induced sequence distribution changes. Appendix E analyzes the paired evidence and preference updates.

Two-stage UCB-inspired allocation. With a limited comparison budget, we use a two-stage UCB-inspired rule to balance exploitation and exploration (Auer et al., 2002; Kocsis & Szepesvári, 2006). The primary stage favors high-preference cameras while exploring cameras with fewer comparison participations; the comparison stage favors close estimated competitors while exploring under-compared pairs. Separate camera-level and pair-level counts track coverage for the two stages.

Let N_m, $N_{m,v}$, and $N_{m,\{v,u\}}$ count comparison decisions, camera participation in comparisons, and unordered-pair comparisons, respectively. Among allocated comparison decisions, unseen pairs are prioritized until every unordered pair has been covered once. After coverage, both cameras are chosen before either score is observed:

$$\begin{aligned} v_n &\in \arg\max_{v\in\mathcal{V}} \left[R_{m,v} + c\sqrt{\frac{\log(1+N_m)}{1+N_{m,v}}} \right], \\ u_n &\in \arg\max_{u\in\mathcal{V}\setminus\{v_n\}} \left[-|R_{m,v_n} - R_{m,u}| + c\sqrt{\frac{\log(1+N_m)}{1+N_{m,\{v_n,u\}}}} \right], \end{aligned} \qquad c \geq 0. \tag{7}$$

Routine decisions use these preferences for single-view supervision. Until initial pair coverage is complete, they sample their primary view uniformly. Thereafter, they select the camera with the highest current preference:

$$v_n \in \arg\max_{v\in\mathcal{V}} R_{m,v}. \tag{8}$$

Comparison decisions use the primary from coverage or Eq. (7). In both cases, only the preselected primary supplies MVR's training record and reference-pool candidate under its original update rule. The comparison clip updates preferences for subsequent choices, preserving one stored training clip per decision.

For each task, we enforce a comparison fraction $b \in [0, 1]$ over all camera decisions collected so far. For batch j with T_j decisions, let $S_j = \sum_{i=1}^{j} T_i$, with $S_0 = 0$. We designate

$$B_j = \lfloor bS_j \rfloor - \lfloor bS_{j-1} \rfloor \tag{9}$$

of the T_j decisions as comparison decisions. Initial pair coverage is counted within this budget. Since each comparison decision requires one additional comparison clip, after batch j, the total number of scored clips is $S_j + \lfloor bS_j \rfloor \leq (1+b)S_j$. Thus, $b = 0.05$ adds at most 5% scored clips relative to scoring one view per segment, with no additional environment interaction.

3.3 Direction-Aware Visual Reward

Once visual supervision has been translated into state scores, the remaining choice is how to use them for policy learning. Under SAVR (Eq. (3)), the immediate visual term depends only on the source state. Actions from that state receive the same visual term even when they produce opposite score changes. DAVR replaces this source-state term with the consecutive-state score difference, making immediate visual feedback reflect the transition's assessed progress or regression.

Panel (b) of Fig. 2 depicts this operation: the shared phase-frozen scorer evaluates the two endpoints of a transition; their score difference is added to the task reward, and the resulting reward is used for policy learning.

To prevent changes in the scoring criterion from being mixed with state changes, we hold both the relevance model and reference distribution fixed during replay relabeling and policy learning after update round k, and define

$$\Delta r^{\text{vis}}_{t,m,k} = r^{\text{VLM}}_{m,k}(s_{t+1}) - r^{\text{VLM}}_{m,k}(s_t). \tag{10}$$

The resulting reinforcement-learning reward is

$$r^{\text{DAVR}}_{t,m,k} = r_{\text{task},t} + w_m\, \Delta r^{\text{vis}}_{t,m,k}. \tag{11}$$

For the phase-frozen scorer, the state score in Eq. (2) is strictly increasing in $f^{\mathrm{MVR}}_{m,k}(s)$. Positive, negative, and zero visual differences therefore indicate assessed progress, regression, and stagnation, respectively.

Both endpoints are evaluated directly in state space by the same phase-frozen scorer, so DAVR needs no additional vision-language model query or per-step rendering.

Accumulated directional feedback. To characterize how the visual feedback accumulates along a trajectory, we first consider its undiscounted sum. For any uninterrupted segment $s_p, \ldots, s_q$, with $p < q$, within a single episode and a frozen reinforcement-learning phase,

$$\sum_{t=p}^{q-1} \Delta r^{\mathrm{vis}}_{t,m,k} = r^{\mathrm{VLM}}_{m,k}(s_q) - r^{\mathrm{VLM}}_{m,k}(s_p). \tag{12}$$

Adjacent scores cancel, so the undiscounted visual return measures net assessed progress and, for fixed endpoints, does not depend on time spent at intermediate states. Under discounting, intermediate state scores generally no longer cancel exactly. We therefore analyze the relationship between DAVR and SAVR under discounted objectives, including the coefficient mapping and finite- and infinite-horizon cases, in Appendix B.

Further details on the training settings, reference-score implementation, view initialization, comparison scheduling, tie-breaking, replay relabeling, and boundary handling are provided in Appendix C. Algorithm 1 gives the complete online procedure.

4 Experiments

Following MVR's benchmark protocol, we evaluate DRAVS on the ten-task Continual World benchmark and nine HumanoidBench tasks. The experiments examine performance across manipulation and whole-body control, the contribution of each design under matched settings, and whether higher final performance is accompanied by reduced late-stage degradation.

4.1 Experimental Setup

Benchmarks and metrics. We use the ten-task CW10 subset of Continual World, built on Meta-World (Yu et al., 2020; Wołczyk et al., 2021), and nine Unitree H1 tasks from HumanoidBench (Sferrazza et al., 2024). Following MVR (Luo et al., 2026), training lasts one million environment steps for each Continual World task and ten million for each HumanoidBench task. At the end of training, we report success rates for the Continual World benchmark over five seeds and episodic returns for the HumanoidBench tasks over three seeds. Each seed is evaluated for ten episodes using deterministic actions on both benchmarks. Results are means with one standard deviation across seeds; benchmark averages are macro-averages of task means. Average rank is computed from task-wise mean performance over all six methods, with ties assigned their average rank; lower is better.

Baselines and training. Baselines include DreamerV3 and Truncated Quantile Critics (TQC), alongside the visual-reward methods VLM-RM, RoboCLIP, and MVR, with results from Luo et al. (2026). We retain MVR's TQC backbone (Kuznetsov et al., 2020), frozen ViCLIP-L model (Wang et al., 2024a), state-relevance learner, task rewards, prompts, and training schedule. We use $w = 0.1$ and, when BRVS is enabled, $b = 0.05$. Matched ablations vary the reward assignment and supervision-view selection rule with the remaining settings fixed. Appendix C provides implementation details.

Hard-5 evaluation. Hard-5 comprises the five CW10 tasks with published MVR mean final success at most 0.50: hammer, push-back, stick-pull, push, and shelf-place. We use this subset for matched ablations and learning-dynamics analysis to focus on tasks with substantial room for improvement under MVR. For matched ablations, we average these task scores within each seed, then report the mean and sample standard deviation over five seeds.

Table 1: **MetaWorld CW10 final-policy comparison.** All five baseline columns are reproduced from (Luo et al., 2026). DRAVS reports mean success ± standard deviation over five seeds at the final checkpoint. Hard-5 contains the tasks with published MVR mean final success at most 0.50.

Task	DreamerV3	TQC	VLM-RM	RoboCLIP	MVR	DRAVS
hammer	0.10 ± 0.17	0.20 ± 0.40	0.40 ± 0.49	0.00 ± 0.00	0.20 ± 0.24	$\mathbf{0.60 \pm 0.20}$
push-wall	0.00 ± 0.00	0.10 ± 0.20	0.40 ± 0.37	0.60 ± 0.37	0.60 ± 0.49	$\mathbf{0.70 \pm 0.25}$
faucet-close	0.13 ± 0.15	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
push-back	0.37 ± 0.38	$\mathbf{0.60 \pm 0.37}$	0.30 ± 0.40	0.20 ± 0.40	0.50 ± 0.45	$\mathbf{0.60 \pm 0.20}$
stick-pull	0.00 ± 0.00	0.10 ± 0.20	0.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.24	$\mathbf{0.30 \pm 0.24}$
handle-press-side	0.77 ± 0.17	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
push	0.00 ± 0.00	0.10 ± 0.20	0.00 ± 0.00	0.30 ± 0.40	0.10 ± 0.20	$\mathbf{0.50 \pm 0.00}$
shelf-place	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.20 \pm 0.25}$	$\mathbf{0.20 \pm 0.24}$	$\mathbf{0.20 \pm 0.24}$
window-close	0.61 ± 0.16	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
peg-unplug-side	0.77 ± 0.19	0.30 ± 0.40	0.60 ± 0.49	0.80 ± 0.40	0.60 ± 0.37	$\mathbf{1.00 \pm 0.00}$
Hard-5 macro mean	0.09	0.20	0.14	0.14	0.24	**0.44**
CW10 macro mean	0.28	0.44	0.47	0.51	0.54	**0.69**
Average rank	5.15	3.65	4.00	3.45	3.00	**1.75**

4.2 Comparisons with Baseline Methods

MetaWorld CW10. DRAVS improves overall manipulation performance while preserving success on tasks already solved by MVR (Table 1). It achieves the best or tied-best mean success on all ten CW10 tasks, raising the macro mean from 0.54 to 0.69. The improvement is more pronounced on Hard-5, where the mean increases from 0.24 to 0.44. In particular, success on hammer and push improves from 0.20 to 0.60 and from 0.10 to 0.50, respectively, while push-back and stick-pull also show consistent gains. Beyond Hard-5, peg-unplug-side improves from 0.60 to 1.00, indicating that the gains are not limited to the most difficult subset. At the same time, all three tasks on which MVR already reaches a mean success of 1.00 remain at 1.00, showing that the higher overall performance is obtained without sacrificing previously solved tasks. This broader across-task improvement is also reflected in the average rank, which decreases from 3.00 for MVR to 1.75 for DRAVS.

HumanoidBench H1. The performance gains extend beyond manipulation to HumanoidBench locomotion, posture, and balance tasks (Table 2). DRAVS achieves the highest mean final return on all nine tasks and raises the task-macro mean from MVR's 616.9 to 836.3, a relative improvement of 35.6%. The largest gains occur on Stair and the two Balance tasks, with Stair improving from 208.60 to 783.74, Balance_Simple from 299.87 to 787.63, and Balance_Hard from 95.78 to 479.81. Improvements also appear on stronger MVR tasks such as Stand and Sit_Simple, indicating that the gains are not confined to low-baseline settings. This broader improvement is also reflected in the average rank, which improves from 2.67 for MVR to 1.00 for DRAVS.

Table 2: **HumanoidBench H1 comparison.** Baseline columns are taken from the original MVR paper (Luo et al., 2026). DRAVS reports mean return ± standard deviation over three seeds on the same nine tasks.

Task	DreamerV3	TQC	VLM-RM	RoboCLIP	MVR	DRAVS
Walk	800.2 ± 158.7	510.58 ± 299.16	535.35 ± 355.18	737.34 ± 194.62	927.47 ± 1.83	$\mathbf{947.36 \pm 2.59}$
Run	633.8 ± 222.4	647.87 ± 186.98	14.93 ± 1.11	501.15 ± 179.71	749.23 ± 56.82	$\mathbf{904.40 \pm 7.52}$
Stair	131.1 ± 43.6	282.95 ± 120.54	44.96 ± 4.12	211.33 ± 56.25	208.60 ± 166.22	$\mathbf{783.74 \pm 154.46}$
Slide	436.5 ± 200.1	514.91 ± 106.36	163.13 ± 41.22	494.20 ± 21.66	735.03 ± 142.85	$\mathbf{925.77 \pm 12.57}$
Stand	622.7 ± 404.8	576.59 ± 371.0	728.69 ± 102.19	849.73 ± 108.86	918.55 ± 29.30	$\mathbf{960.20 \pm 1.35}$
Sit_Simple	891.4 ± 38.4	822.07 ± 101.28	293.72 ± 65.46	553.90 ± 244.51	861.07 ± 19.35	$\mathbf{953.93 \pm 0.77}$
Sit_Hard	433.4 ± 355.9	511.85 ± 155.45	322.95 ± 45.70	559.38 ± 192.00	756.67 ± 108.79	$\mathbf{783.95 \pm 33.77}$
Balance_Simple	19.8 ± 7.0	256.79 ± 54.23	654.42 ± 335.19	215.81 ± 18.19	299.87 ± 58.45	$\mathbf{787.63 \pm 107.64}$
Balance_Hard	45.9 ± 27.4	92.35 ± 10.76	107.36 ± 24.93	97.03 ± 18.13	95.78 ± 11.03	$\mathbf{479.81 \pm 57.10}$
Task-macro mean return	446.1	468.4	318.4	468.9	616.9	**836.3**
Average rank	4.56	4.11	4.78	3.89	2.67	**1.00**

4.3 Ablation Study

To distinguish the effects of reward assignment and supervision-view selection, we conduct a matched 2×2 study on Hard-5, combining SAVR or DAVR with uniform view sampling or BRVS (Table 3). With the implementation, seeds, prompts, training steps, and evaluation protocol held fixed, this design tests each component under both settings of the other. SAVR with uniform sampling serves as the matched MVR reproduction, while the benchmark comparisons use published MVR results.

Table 3: **Matched Hard-5 component ablation.**

Reward assignment	Uniform	BRVS	Final success ↑
SAVR	✓		0.25 ± 0.15
DAVR	✓		0.34 ± 0.11
SAVR		✓	0.31 ± 0.13
DAVR		✓	$\mathbf{0.44 \pm 0.09}$

Both reward assignment and view selection contribute to the improvement. The matched control achieves a mean final success of 0.25, close to the published MVR Hard-5 mean of 0.24 in Table 1. Replacing SAVR with DAVR raises mean final success from 0.25 to 0.34 under uniform view sampling and from 0.31 to 0.44 with BRVS. Likewise, replacing uniform sampling with BRVS raises it from 0.25 to 0.31 under SAVR and from 0.34 to 0.44 under DAVR. Thus, each component improves the mean under both settings of the other, and their combination achieves the highest mean success. The BRVS configurations use at most 5% more scored clips relative to scoring one view per segment, without additional environment interaction.

Additional analyses in Appendix D further examine reward calibration, comparison budget, and view-allocation mechanisms. Successor-state indexing and coefficient calibration do not reproduce DAVR's mean performance; the budget sweep reaches its highest mean at the main $b = 0.05$ setting, while mechanism controls support the roles of same-sequence pairing, continued routine adaptation, and two-stage comparison allocation.

4.4 Learning Dynamics

We examine whether higher final success is accompanied by better retention on Hard-5 (Fig. 3). On hammer and push, the matched MVR reproduction improves early but later loses part of its gains, whereas DRAVS continues improving and reaches a higher plateau. Smaller late-stage advantages also appear on stick-pull and push-back, while shelf-place remains close. Additional qualitative results are provided in Appendix F.

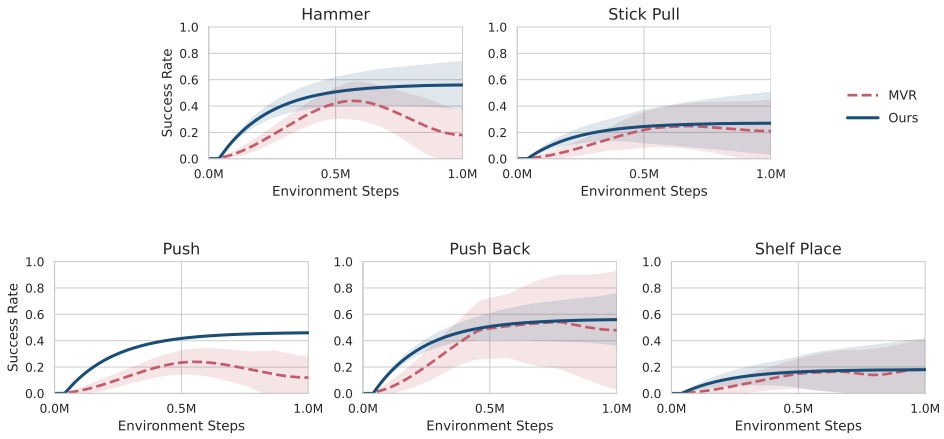


Figure 3: **MetaWorld learning dynamics.** Success across training on the Hard-5 subset for our method and a matched MVR reproduction. Curves summarize evaluations at fixed training checkpoints, and shaded regions denote one standard deviation across seeds.

5 Conclusion

We presented DRAVS, a framework that combines directional visual reward assignment with adaptive supervision-view selection for online reinforcement learning. DAVR converts consecutive state-score changes into signed transition feedback, while BRVS learns persistent camera preferences from sparse same-segment comparisons to guide visual supervision. Experiments on the ten-task Continual World benchmark and nine HumanoidBench tasks show that DRAVS outperforms the state-of-the-art multi-view video reward method MVR. Matched ablations support the effectiveness of both components, while learning curves show better retention of earlier gains on tasks exhibiting late-stage degradation.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improve the readability of the manuscript. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. Video-language critic: Transferable reward functions for language-conditioned robotics. *Transactions on Machine Learning Research*, 2025.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

Tewodros W. Ayalew, Xiao Zhang, Kevin Yuanbo Wu, Tianchong Jiang, Michael Maire, and Matthew R. Walter. PROGRESSOR: A perceptually guided reward estimator with self-supervised online refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10297–10306, 2025.

Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Heuristic-guided reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13550–13563, 2021.

Ricson Cheng, Arpit Agarwal, and Katerina Fragkiadaki. Reinforcement learning of active vision for manipulating objects under occlusions. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 422–431, 2018.

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

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

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. FuRL: Visual-language models as fuzzy rewards for reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14256–14274. PMLR, 2024.

Yunzhong Hou, Stephen Gould, and Liang Zheng. Learning to select views for efficient multi-view understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20135–20144, 2024.

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

Changyeon Kim, Minho Heo, Doohyun Lee, Honglak Lee, Jinwoo Shin, Joseph J. Lim, and Kimin Lee. Subtask-aware visual reward learning from segmented demonstrations. In *International Conference on Learning Representations*, 2025.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pp. 282–293. Springer, 2006. doi: 10.1007/11871842_29.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5556–5566. PMLR, 2020.

Yuyang Liu, Chuan Wen, Yihang Hu, Dinesh Jayaraman, and Yang Gao. TimeRewarder: Learning dense reward from passive videos via frame-wise temporal distance. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Lirui Luo, Guoxi Zhang, Hongming Xu, Yaodong Yang, Cong Fang, and Qing Li. MVR: Multi-view video reward shaping for reinforcement learning. In *International Conference on Learning Representations*, 2026.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23301–23320, 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023b.

Yecheng Jason Ma, Joey Hejna, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. Vision language models are in-context value learners. In *International Conference on Learning Representations*, 2025.

Yiming Mao, Zixi Yu, Weixin Mao, Yinhao Li, Qirui Hu, Zihan Lan, Minzhao Zhu, and Hua Chen. ARM: Advantage reward modeling for long-horizon manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4468–4477, June 2026.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. HumanoidBench: Simulated humanoid benchmark for whole-body locomotion and manipulation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.061.

Jinghuan Shang and Michael S. Ryoo. Active vision reinforcement learning under limited visual observability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10316–10338, 2023.

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. RoboCLIP: One demonstration is enough to learn robot policies. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55681–55693, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018. ISBN 9780262039246.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. InternVid: A large-scale video-text dataset for multimodal understanding and generation. In *Proceedings of the Twelfth International Conference on Learning Representations*, 2024a.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 51484–51501. PMLR, 2024b.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual World: A robotic benchmark for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28496–28510, 2021.

Daniel Yang, Davin Tjia, Jacob Berg, Dima Damen, Pulkit Agrawal, and Abhishek Gupta. Rank2Reward: Learning shaped reward functions from passive video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2806–2813, 2024. doi: 10.1109/ICRA57147.2024.10610873.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020.

Yinuo Zhao, Jiale Yuan, Zhiyuan Xu, Xiaoshuai Hao, Xinyi Zhang, Kun Wu, Zhengping Che, Chi Harold Liu, and Jian Tang. Training-free generation of temporally consistent rewards from VLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8133–8143, 2025.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 10–18. PMLR, 2014.

A Discussion and Limitations

Visual assessment. DRAVS improves supervision selection and reward assignment while retaining MVR's pretrained vision-language model and state-scoring formulation. The usefulness of its feedback therefore depends on how well the underlying visual assessment captures task-relevant behavior. Task-specific adaptation of the visual model or richer language descriptions could extend the framework to behaviors that are less well represented by the current assessment.

Granularity of view selection. BRVS maintains task-level camera preferences to reuse sparse comparison evidence across execution segments. Although these preferences are updated throughout training, view selection is not explicitly conditioned on the current state or execution stage. For tasks in which useful viewpoints vary across stages, state- or stage-conditioned preferences could offer finer-grained selection. Such extensions would need to balance this specificity with the comparison evidence required to learn the additional preferences.

B Discounted Reward-Assignment Relation

With a fixed scorer, DAVR's undiscounted visual feedback telescopes to the difference between endpoint scores. The difference is also invariant to a common additive shift $r^{\mathrm{VLM}}_{m,k}(s) \mapsto r^{\mathrm{VLM}}_{m,k}(s)+\kappa$, where κ is a constant within the phase. The derivations below justify the discounted-objective relation in Section 3.3, separating finite-horizon endpoint terms from the infinite-horizon coefficient mapping to source-state SAVR.

Finite-horizon relation. Let $\Phi = r^{\mathrm{VLM}}_{m,k}$ denote the state scorer, with both its relevance model and reference set held fixed. Consider a state sequence $s_0, \ldots, s_H$, where $H \geq 1$. For $0 < \gamma < 1$, define the source-state and difference visual returns before applying their respective visual coefficients:

$$G_{\Phi,H} = \sum_{t=0}^{H-1} \gamma^t \Phi(s_t), \qquad G_{\Delta,H} = \sum_{t=0}^{H-1} \gamma^t [\Phi(s_{t+1}) - \Phi(s_t)]. \tag{13}$$

Reindexing the successor-state sum gives

$$\begin{aligned} G_{\Delta,H} &= \frac{1}{\gamma}\left[G_{\Phi,H} - \Phi(s_0) + \gamma^H \Phi(s_H)\right] - G_{\Phi,H} \\ &= \frac{1-\gamma}{\gamma} G_{\Phi,H} - \frac{\Phi(s_0)}{\gamma} + \gamma^{H-1}\Phi(s_H). \end{aligned} \tag{14}$$

Thus, the discounted difference return consists of a rescaled source-state return and two endpoint terms. The final endpoint term depends on the reached state and cannot generally be dropped at a finite horizon.

For an uninterrupted segment $s_p, \ldots, s_q$ within a single episode and frozen reinforcement-learning phase, the same identity can be written in the notation of Section 3.3. Writing $h = q - p \geq 1$ and measuring discount exponents relative to the segment start gives

$$\begin{aligned} \sum_{t=p}^{q-1} \gamma^{t-p} \Delta r^{\mathrm{vis}}_{t,m,k} = &-r^{\mathrm{VLM}}_{m,k}(s_p) + \gamma^{h-1} r^{\mathrm{VLM}}_{m,k}(s_q) \\ &+ (1-\gamma) \sum_{t=p+1}^{q-1} \gamma^{t-p-1} r^{\mathrm{VLM}}_{m,k}(s_t). \end{aligned} \tag{15}$$

For $h > 1$, discounting also weights the intermediate state scores by $(1-\gamma)\gamma^{t-p-1}$, so this visual return no longer depends only on the endpoints. Accordingly, the finite-horizon DAVR objective augments the discounted task return with this difference return. Thus, under a fixed scorer, DAVR optimizes the task return together with a rescaled accumulation of state relevance and explicit endpoint terms, rather than an endpoint-only difference under discounting.

Connection to SAVR. Denote the DAVR and SAVR visual coefficients by w_D and w_S, respectively. The per-transition relation to source-state SAVR and potential-based shaping is

$$w_D[\Phi(s_{t+1}) - \Phi(s_t)] = \frac{w_D(1-\gamma)}{\gamma}\Phi(s_t) + \frac{w_D}{\gamma}[\gamma\Phi(s_{t+1}) - \Phi(s_t)].$$

The first term is a rescaled SAVR visual reward; the second is a potential-based shaping term (Ng et al., 1999).

If the same scorer is fixed over an infinite horizon, $G_{\Phi,H}$ converges, and the boundary term satisfies $\gamma^{H-1}\Phi(s_H) \to 0$, the finite-horizon relation reduces to

$$G_\Delta = \frac{1-\gamma}{\gamma}G_\Phi - \frac{\Phi(s_0)}{\gamma}. \tag{16}$$

For a common task reward and initial-state distribution p_0, let J denote the expected augmented discounted return. Assuming finite expected returns and an integrable initial score, the corresponding objectives satisfy

$$J_{\mathrm{DAVR}}(\pi; w_D) = J_{\mathrm{SAVR}}\left(\pi; w_D\frac{1-\gamma}{\gamma}\right) - \frac{w_D}{\gamma}\mathbb{E}_{s_0\sim p_0}[\Phi(s_0)], \tag{17}$$

where w_D and $w_S = w_D(1-\gamma)/\gamma$ are the DAVR and SAVR coefficients. With the same fixed scorer and initial-state distribution across policies, the last term is policy-independent. Thus, under these conditions, the two augmented objectives induce the same policy ordering at the corresponding coefficients.

Interpretation for learning. This relation concerns the two augmented objectives, rather than guaranteeing invariance relative to the task reward alone. It does not imply identical immediate rewards or learning dynamics. Finite-episode boundary terms and online scorer updates additionally require separate consideration. With $\gamma = 0.99$ and $w_D = 0.1$, the corresponding coefficient is $w_S \approx 0.0010101$. This mapping concerns the effective source-state visual coefficient in the discounted objective, rather than initial standard-deviation matching. The same-coefficient ablation in Table 3 uses the same numerical visual coefficient for DAVR and SAVR, rather than coefficients related by this mapping. Section D.1 describes a SAVR control using the discount-converted coefficient and a separate control that matches the initial standard deviations of the visual reward terms.

C Implementation Details

Training settings and update schedule. We follow MVR's TQC training and visual-scoring protocol (Luo et al., 2026). The actor and critic use three 256-unit hidden layers, with discount $\gamma = 0.99$. The visual pipeline uses ViCLIP-L, 224×224 images, and 64-frame clips. The relevance model and replay rewards are updated every 100,000 environment steps. Task prompts, optimization settings, and the four fixed cameras are inherited from MVR. We retain $K_{\mathrm{ref}} = 10$ reference sequences, following MVR. The main comparisons fix the released default visual coefficient at $w = 0.1$; reward-assignment controls are reported in Section D.1.

Algorithm 1 summarizes the online loop for one task. Following MVR's released collection schedule, pending visual-supervision segments are processed every 10,000 environment steps; collection is skipped when no eligible segments are available. Each nonempty collection batch, indexed by j, supplies primary-only visual records. Scorer updates, indexed by k, refresh reward labels every 100,000 environment steps once sufficient visual and reference data are available. These intervals count transitions across all parallel environments. If both events occur at the same environment step, collection is processed first, followed by scorer fitting, reference refresh, replay relabeling, and policy learning. Camera preferences and budget counts are not reset by either event. Policy and critic updates follow the inherited TQC schedule.

Visual supervision and view selection. Following MVR, one out of every nine episodes supplies visual-supervision data. For the Sit_Hard task, we reuse each sampled initial scene configuration for 15 consecutive visual-supervision episodes before resampling it, providing a locally stable geometry

for view selection. Consecutive 64-frame segments are extracted within these selected episodes, without crossing reset boundaries. Episodes determine which interaction data are eligible; segments are the units of camera selection, paired comparison, and scoring-budget accounting. Each segment receives its own preselected primary camera. Matched uniform-view configurations independently sample one camera per segment, using the same episode and segment sampling schedule as BRVS. Each camera decision uses a fixed-length sequence with stored simulator configurations sufficient to reconstruct its rendered observations. Camera selection precedes rendering and scoring. For a comparison decision, the renderer reconstructs the same stored configurations from the two preselected cameras, without advancing the training environment or collecting another rollout. The frozen vision-language model supplies both the score and the video embedding for each rendered clip. Only the primary score, embedding, and state sequence enter the visual training data and reference update. The comparison output is discarded after updating view preferences.

For each task, $R_{m,v}$, N_m, $N_{m,v}$, and $N_{m,\{v,u\}}$ start at zero. We randomly order all unordered camera pairs independently of their scores and assign primary and comparison roles so that primary-camera counts differ by at most one whenever possible. The resulting ordered list $\mathcal{P}_m$ of $\binom{V}{2}$ pairs and cursor $p_m = 1$ implement the initial coverage pass: only allocated comparison decisions advance the cursor. Preferences, counts, and the cursor persist across collection batches and scorer updates. We set the constant preference-update step size to $\eta = 0.1$ and the exploration coefficient to $c = 1.0$ in both stages of Eq. (7). Constant-step updates retain responsiveness to changing evidence (Sutton & Barto, 2018), while the count-based bonuses promote comparison coverage (Auer et al., 2002; Zoghi et al., 2014). Ties in either UCB stage and in routine preference maximization are broken uniformly at random.

Within each task, batch j receives B_j comparison decisions according to Eq. (9). Allocating the increase in $\lfloor bS_j \rfloor$, rather than rounding bT_j separately, carries fractional allowances across batches and gives $\sum_{i=1}^{j} B_i = \lfloor bS_j \rfloor$. All comparisons, including initial pair coverage, share this budget with $b = 0.05$; coverage may span batches. With four cameras, the six initial comparisons are completed by the end of the first batch with $S_j \geq 120$. Before scoring, we sample B_j of the T_j segment indices uniformly without replacement. These comparisons are processed sequentially in a score-independent order, updating preferences after each one and incrementing N_m, both participating camera counts, and the queried unordered-pair count by one. Routine decisions then use the updated preferences: they sample uniformly until coverage is complete and follow Eq. (8) thereafter, without changing comparison counts. The routine group is also processed in a score-independent order, as specified in Algorithm 1.

State scoring and replay updates. Primary records update MVR's bounded visual dataset $\mathcal{D}^{\mathrm{MVR}}$ and persistent top-K_{ref} reference pool $\mathcal{D}^{\mathrm{ref}}$. Reference records are retained independently of visual-dataset eviction. At a scorer update, we first fit the relevance model, then snapshot the current reference pool and recompute all reference relevance values with the fitted model. The model and reference snapshot are frozen together before replay relabeling. Later collection batches may update the retained pool but do not alter this active snapshot. Every replay reward is recomputed from its stored task reward and the new visual term, rather than adding another visual term to an already shaped reward. Newly collected transitions use the same active scorer until the next update; before the first valid scorer, training uses the task reward alone.

The fixed relevance temperature is 0.1, which is absorbed into $f_{m,k}^{\mathrm{MVR}}$ in Eq. (2). We evaluate its expectation as the uniform empirical mean over all states in the frozen reference sequences. Cached reference relevance values are shared by both endpoints and remain unchanged throughout the reinforcement-learning phase. Thus, repeated evaluations of the same state use the same scoring function without resampling references. Log-sigmoid is evaluated in a numerically stable form. The state score is unweighted and unclipped; w_m is applied exactly once when forming the SAVR term or DAVR difference, with no additional clipping of the resulting visual term. Setting $w_m = 0$ disables the visual term in either assignment. Replay retains the original task rewards and terminal/truncation masks. True terminations use the pre-reset final observation for the score difference and zero bootstrap; time-limit truncations retain bootstrap from the final observation. No difference is formed between a final observation and the reset observation of the next episode. The same boundary handling applies when storing new transitions and relabeling existing ones.

Evaluation and aggregation. Let $\mathcal{T}_{\text{Hard-5}}$ denote the five Hard-5 tasks and $Y_{m,i}^{\text{final}}$ final-checkpoint success for task m and seed i. The matched ablation first computes

$$\bar{Y}_i^{\text{Hard-5}} = \frac{1}{|\mathcal{T}_{\text{Hard-5}}|} \sum_{m \in \mathcal{T}_{\text{Hard-5}}} Y_{m,i}^{\text{final}}, \tag{18}$$

and reports its mean and sample standard deviation over five seeds. This seed-level aggregation differs from averaging the five task standard deviations. The final-policy checkpoint is used throughout; no intermediate peak enters Tables 1 to 3. Following MVR's benchmark protocol (Luo et al., 2026), we use the same training budgets and seed counts. Each policy is evaluated for ten episodes per checkpoint using deterministic actions. For the HumanoidBench tasks, episode returns at the final checkpoint are averaged within each seed, and the mean and standard deviation are reported across the three seeds.

Algorithm 1 Online learning with DRAVS

Require: Task m, prompt ℓ_m, cameras $\mathcal{V}$, renderer, frozen vision-language model; settings $b, c, \eta, w_m, K_{\text{ref}}$; collection, scorer-update, and TQC schedules

Initialize the TQC agent, replay $\mathcal{D}^{\text{RL}}$, visual data $\mathcal{D}^{\text{MVR}}$, and reference pool $\mathcal{D}^{\text{ref}}$; no active scorer
Set $j = k = S_0 = 0$; initialize zero preferences and comparison counts, coverage list $\mathcal{P}_m$, and cursor $p_m = 1$
for each environment step **do**
  Collect $(s_t, a_t, r_{\text{task},t}, s_{t+1})$ and boundary masks, retaining the pre-reset final observation when needed
  Store the original transition in $\mathcal{D}^{\text{RL}}$; label it by Eq. (11) with the active scorer, or by $r_{\text{task},t}$ if none is available
  Retain simulator configurations from selected episodes for segment construction
  if collection is due (every 10,000 environment steps) and eligible segments are available **then**
    Set $j \leftarrow j + 1$; take T_j pending eligible segments and initialize $\mathcal{B}_j^{\text{pri}} \leftarrow \emptyset$
    Set $S_j \leftarrow S_{j-1} + T_j$ and $B_j \leftarrow \lfloor bS_j \rfloor - \lfloor bS_{j-1} \rfloor$
    Sample B_j indices $\mathcal{Q}_j$ uniformly without replacement before scoring
    for each segment n, with $\mathcal{Q}_j$ first and score-independent order within both groups **do**
      if $n \in \mathcal{Q}_j$ **then**
        Preselect (v_n, u_n) from $\mathcal{P}_m[p_m]$ and advance p_m during coverage; otherwise use Eq. (7)
      else
        Select v_n uniformly before coverage completes; otherwise use Eq. (8)
      end if
      Render $\mathbf{s}_n$ from v_n; obtain its vision-language model score C_{n,v_n} and video embedding
      if $n \in \mathcal{Q}_j$ **then**
        Render the same $\mathbf{s}_n$ from u_n and compute C_{n,u_n}
        Apply Eqs. (5) and (6); increment N_m, N_{m,v_n}, N_{m,u_n}, $N_{m,\{v_n,u_n\}}$; discard comparison output
      end if
      Add only the primary state sequence, embedding, and score to $\mathcal{B}_j^{\text{pri}}$
    end for
    Update $\mathcal{D}^{\text{ref}}$ with $\mathcal{B}_j^{\text{pri}}$, retaining the top K_{ref} sequences by score
    Insert $\mathcal{B}_j^{\text{pri}}$ into $\mathcal{D}^{\text{MVR}}$ using MVR's capacity and eviction rules
  end if
  if a scorer update is due (every 100,000 environment steps) and visual/reference data are sufficient **then**
    Set $k \leftarrow k + 1$; fit $f_{m,k}^{\text{MVR}}$ using $\mathcal{D}^{\text{MVR}}$ under MVR's original objective
    Snapshot $\mathcal{D}^{\text{ref}}$ to define $\mu_{m,k}^{\text{ref}}$; recompute all reference relevance values with $f_{m,k}^{\text{MVR}}$
    Freeze the complete scorer in Eq. (2) and make it active
    Relabel all replay transitions from their stored $r_{\text{task},t}$ using Eq. (11), preserving boundary masks
  end if
  Perform scheduled TQC updates using the current replay labels
end for

Ensure: Trained policy, persistent view statistics, primary-only visual/reference data, and active scorer

D Additional Analyses and Ablations

The supplementary controls address reward assignment, comparison budget, and view allocation. They use the Hard-5 protocol and seed-level aggregation in Section C. All success entries report final-policy performance at 1M environment steps as a mean and sample standard deviation over five seeds.

D.1 Reward-Assignment Calibration

We further test whether DAVR's gain can be attributed to successor-state indexing or visual-reward scaling rather than consecutive-state differencing. All controls use uniform camera sampling, the same reinforcement-learning and scoring pipeline, and matched training and evaluation seeds.

We consider three SAVR controls in addition to the base SAVR and DAVR settings in Table 3, with $w_{\text{base}} = 0.1$. *Next-state SAVR* replaces the source-state score with $r^{\text{VLM}}_{m,k}(s_{t+1})$ while retaining absolute score addition. *Std.-matched SAVR* matches the initial standard deviation of the DAVR visual signal. For each task and seed, using a common replay sample and the first valid frozen scorer, we compute

$$\widehat{\sigma}_S = \widehat{\text{Std}}[r^{\text{VLM}}(s_t)], \qquad \widehat{\sigma}_D = \widehat{\text{Std}}[\Delta r^{\text{vis}}],$$

and set

$$w^{\text{SAVR-match}} = w_{\text{base}} \frac{\widehat{\sigma}_D}{\widehat{\sigma}_S}, \qquad \widehat{\sigma}_S > 0.$$

The coefficient is calibrated once and then fixed for the remainder of the run. *Discount-rescaled SAVR* instead uses $w_S = w_{\text{base}}(1-\gamma)/\gamma$, following the fixed-scorer discounted-objective relation in Section B; with $\gamma = 0.99$, this gives $w_S \approx 0.0010101$.

Table 4: **Reward-assignment calibration on Hard-5.** Uniform views; entries report final success as mean ± sample standard deviation over five seeds. The base SAVR and DAVR rows reuse the corresponding anchors from Table 3.

Reward assignment	Final success ↑
SAVR, base w	0.25 ± 0.15
Next-state SAVR, base w	0.27 ± 0.16
SAVR, std.-matched	0.29 ± 0.08
SAVR, discount-rescaled	0.28 ± 0.14
DAVR, base w	0.34 ± 0.11

As shown in Table 4, moving the absolute SAVR score from the source state to the successor state yields only a small increase from 0.25 to 0.27, indicating that successor-state indexing alone does not reproduce the gain from differencing. Matching the initial visual-signal magnitude increases the mean to 0.29, while the discount-rescaled coefficient gives 0.28; both remain below DAVR at 0.34. Thus, neither simple state-indexing changes nor the two coefficient calibrations reproduce DAVR's mean performance, supporting the role of consecutive-state differencing itself. These controls complement the same-coefficient factorial comparison in Table 3.

D.2 Comparison-Budget Sensitivity

This analysis fixes DAVR and varies the cumulative comparison fraction b. The $b = 0$ reference reuses the uniform-view DAVR runs, and $b = 0.05$ reuses the full-method runs in Table 3. Other budgets use independently trained runs with the same five seeds and protocol. For S primary decisions, the scored-clip factor is $1 + \lfloor bS \rfloor / S \le 1 + b$ at round boundaries. This counts clips, not total computation or wall-clock time.

Table 5: **Comparison-budget sensitivity on Hard-5.** The clip factors are analytical upper bounds; entries report final-policy success as mean ± sample standard deviation over five seeds.

Budget b	Scored-clip factor	Final success (mean ± s.d.)
0	1.000	0.34 ± 0.11
0.025	≤ 1.025	0.38 ± 0.16
0.05	≤ 1.050	0.44 ± 0.09
0.10	≤ 1.100	0.43 ± 0.13

As shown in Table 5, introducing a 2.5% comparison budget raises mean final success from 0.34 to 0.38, and increasing the budget to the main 5% setting further raises it to 0.44. Doubling the budget to 10% yields a similar mean of 0.43, rather than a further improvement. These results show that the observed view-selection gains can be obtained with a small comparison budget, while increasing the number of comparisons beyond 5% does not provide a further mean gain in this sweep. This does not imply that larger budgets are universally harmful.

D.3 View-Allocation Mechanisms

The adaptive variants in Table 6 use DAVR and the same $b = 0.05$ comparison budget. Uniform random views serve as the reference without additional comparisons. Initial pair coverage and primary-only storage are retained for every adaptive variant.

Unpaired comparisons score the comparison camera on a different sequence from the same collection batch, keeping clip and interaction budgets fixed while removing the same-sequence control. *Frozen routine view* selects the preference-maximizing routine camera at the end of initial pair coverage and keeps that camera fixed thereafter, while comparison allocation and preference updates continue. *Random paired comparisons* retain same-sequence comparisons but, after initial coverage, sample unordered camera pairs uniformly with random orientation; routine decisions continue to follow the current preference maximizer. These variants isolate same-sequence pairing, continued routine adaptation, and two-stage comparison allocation while holding the comparison budget fixed.

Table 6: **BRVS mechanism controls on Hard-5.** All adaptive variants use the same extra-clip budget. Uniform sampling and full BRVS reuse the corresponding DAVR rows of Table 3.

Variant	Same sequence	Adaptive routine	Two-stage allocation	Final success (mean ± s.d.)
Uniform random views	−−	×	×	0.34 ± 0.11
Unpaired comparisons	×	✓	✓	0.30 ± 0.16
Frozen routine view	✓	×	✓	0.38 ± 0.12
Random paired comparisons	✓	✓	×	0.35 ± 0.14
Full BRVS	✓	✓	✓	0.44 ± 0.09

As shown in Table 6, full BRVS achieves the highest mean final success of 0.44. Removing same-sequence pairing decreases the mean to 0.30, the largest reduction among the adaptive variants. Freezing the routine view gives 0.38, while replacing the two-stage allocation with random paired comparisons gives 0.35. These results are consistent with all three design choices contributing to the final BRVS performance.

E Relative View Evidence

The same-sequence comparisons in Section 3.2 hold the executed behavior fixed while varying the camera. We analyze the resulting relative evidence and the online preference update separately.

Expected relative evidence. Recall that $C_{n,v}$ is the video–text alignment score for sequence n from camera v. For analysis, fix a collection batch j and let $\rho_{m,j}$ denote its distribution over sequences selected for rendering. Define the expected alignment utility

$$U_{m,j}(v) = \mathbb{E}_{n\sim\rho_{m,j}}[C_{n,v}]. \tag{19}$$

This utility characterizes a fixed collection distribution and is not computed by BRVS. In contrast, $R_{m,v}$ is an online selection statistic that retains information across batches through constant-step updates to queried preference differences.

To describe the score component shared by views of the same sequence, define

$$\begin{gathered} d_n = \frac{1}{V}\sum_{u\in\mathcal{V}} C_{n,u}, \qquad \nu_{m,j,v} = U_{m,j}(v) - \frac{1}{V}\sum_{u\in\mathcal{V}} U_{m,j}(u), \\ \zeta_{n,v} = C_{n,v} - d_n - \nu_{m,j,v}. \end{gathered} \tag{20}$$

Here ν is a zero-sum view effect and ζ a view–sequence residual. The shared sequence term d_n cancels in a paired comparison:

$$C_{n,v} - C_{n,u} = \nu_{m,j,v} - \nu_{m,j,u} + \zeta_{n,v} - \zeta_{n,u}.$$

All expectations here are taken over $n \sim \rho_{m,j}$. By construction, $\mathbb{E}[\zeta_{n,v}] = 0$, so a fixed pair satisfies $\mathbb{E}[C_{n,v} - C_{n,u}] = U_{m,j}(v) - U_{m,j}(u) = \nu_{m,j,v} - \nu_{m,j,u}$. Same-sequence subtraction thus removes the shared component while retaining view–sequence variation. The all-view mean d_n is an analysis quantity, not an additional rendering requirement.

Symmetric update properties. Subtracting the two updates in Eq. (6) gives

$$R^{+}_{m,v} - R^{+}_{m,u} = (1-\eta)(R_{m,v} - R_{m,u}) + \eta\Delta C_{n,v,u},$$

where $(+)$ denotes the post-update value. The queried preference difference therefore moves toward the observed gap with step size η. Equal and opposite corrections preserve the zero preference sum established at initialization, while unqueried cameras remain unchanged.

F Qualitative Rollout Comparisons

Figs. 4–6 show synchronized four-frame sequences from seven tasks spanning MetaWorld and H1. Within each case, MVR and Ours use the same reset, camera, and environment-action steps. Across the MetaWorld cases, Ours reaches the task goal earlier, whereas the H1 cases show more stable posture, contact, and task progression over the same horizon.

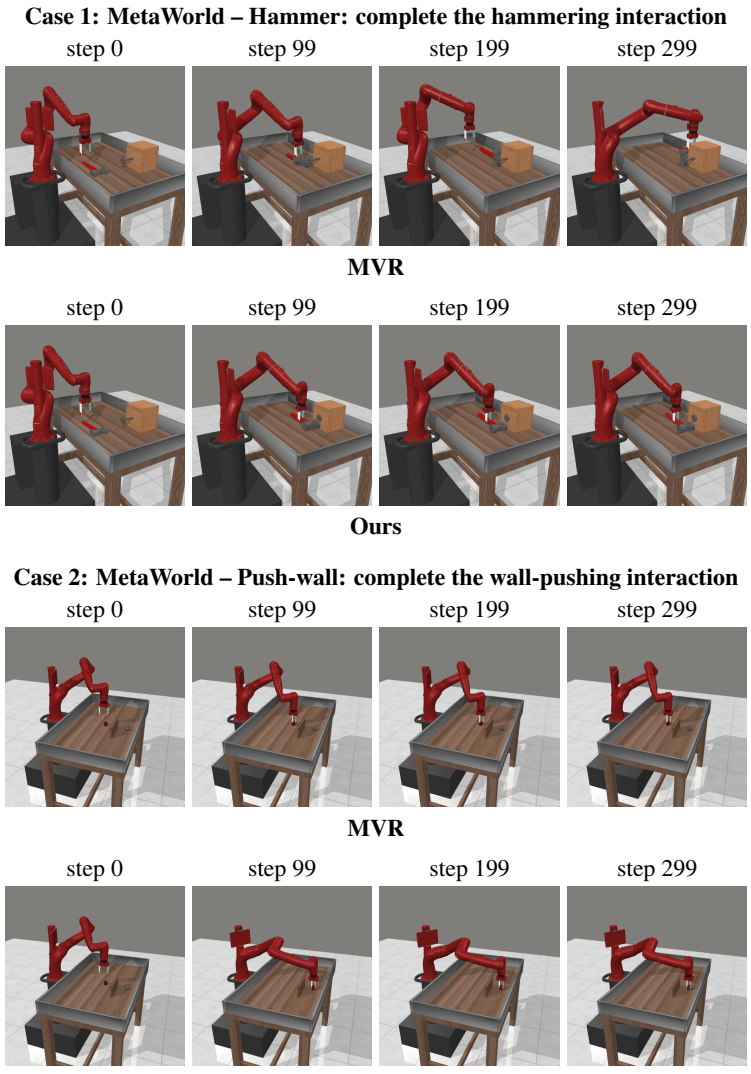
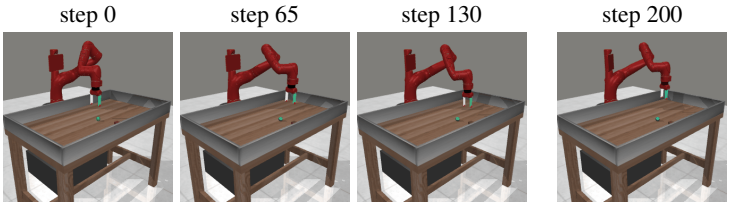
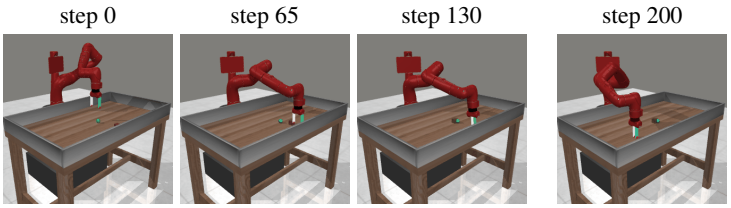


Figure 4: Faster task completion on MetaWorld. At matched environment-action steps, Ours makes task-relevant progress earlier and completes the Hammer and Push-wall interactions within the shown horizon, while MVR remains incomplete at the final step.

Case 3: MetaWorld – Push-back: move the object toward the task target

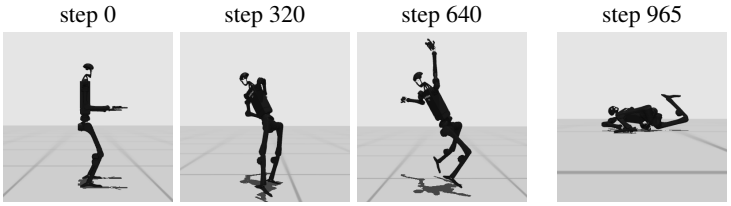


MVR

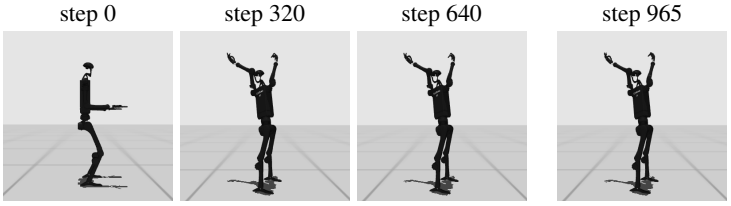


Ours

Case 4: H1 – Stand: retain an upright standing posture

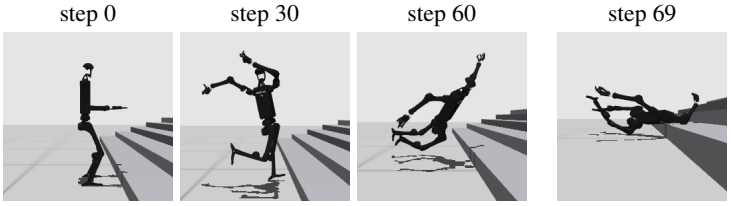


MVR

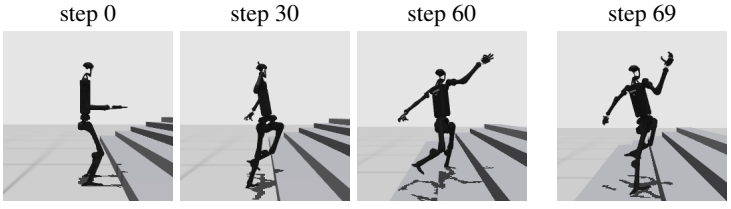


Ours

Case 5: H1 – Stair: advance toward the stair goal



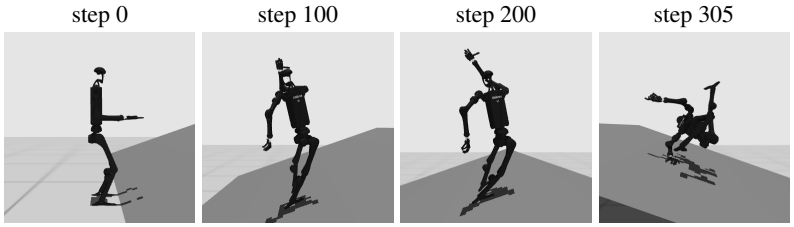
MVR



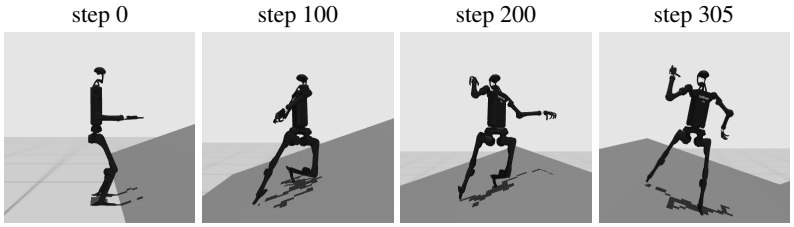
Ours

Figure 5: Faster completion and improved stability. On Push-back, Ours completes the task by step 130, whereas MVR remains incomplete at step 200. On H1 Stand and Stair, Ours maintains an upright, task-relevant posture through the final step, while MVR loses balance.

Case 6: H1 – Slide: maintain support on the slope

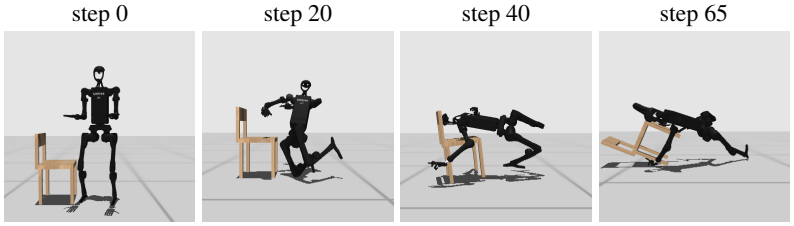


MVR

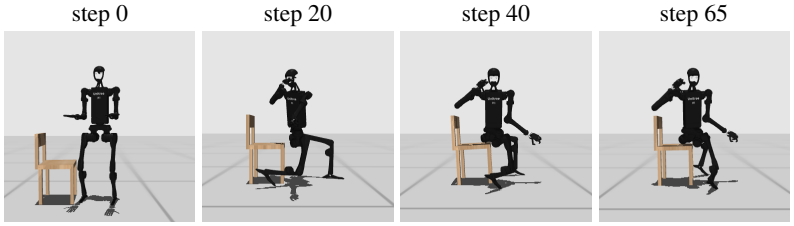


Ours

Case 7: H1 – Sit Hard: reach a seated posture



MVR



Ours

Figure 6: Improved stability on H1. On Slide, Ours remains supported on the slope through step 305, whereas MVR loses balance. On Sit Hard, Ours reaches a stable seated posture by step 65, whereas MVR destabilizes the chair and does not complete the sitting task.