DISENTANGLING TASK-RELEVANT LATENT DYNAMICS FOR REWARD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward design remains a major bottleneck for reinforcement learning in robotic manipulation. Handcrafted dense rewards can compensate for the limited guidance provided by sparse rewards, but require substantial task-specific reward engineering. Learning rewards from videos provides a scalable, data-driven alternative to manual reward design. However, existing video-based rewards are often computed in entangled visual spaces where task-relevant state changes are confounded with extraneous visual variation. As a result, the learned reward may respond to visual distractors rather than actual task execution. To address this limitation, we propose **DiTaR** (**Di**sentangling **Ta**sk-Relevant Latent Dynamics for **R**eward Modeling), a framework for learning dense rewards from heterogeneous videos. DiTaR learns a continuous task-relevant representation from multi-frame video latents through an asymmetric two-stage procedure that suppresses task-irrelevant visual variation. In the task-relevant latent space, we train a history- and language-conditioned rectified-flow model to capture task dynamics and derive a directional dense reward from temporal changes in dynamics compatibility. Together, these components enable reward modeling directly over task-relevant dynamics rather than generic visual variation. DiTaR substantially improves policy learning on MetaWorld-v3 and ManiSkill over video-based reward baselines, and transfers to real-world offline RL through reward relabeling. Reward analyses further show that DiTaR provides more reliable rewards.

1 INTRODUCTION

Reward design remains a major bottleneck for reinforcement learning in robotic manipulation. While sparse rewards are easy to specify, their binary signals provide limited guidance for efficient exploration, often leading to slow convergence or failed policy learning in long-horizon tasks. Handcrafted dense rewards can accelerate optimization by providing continuous feedback, but typically require task-specific design using privileged simulator states, making them difficult to scale across diverse tasks. Learning rewards from videos offers a scalable, data-driven alternative to such manual reward design (Huang et al., 2024; Zhang et al., 2025; Tan et al., 2026; Wang et al., 2026; Lee et al., 2026; Zhai et al., 2025; Chen et al., 2026).

Videos provide a natural source of dense rewards because they record continuous state changes throughout task execution. Prior work leverages this visual signal through video prediction likelihoods (Escontrela et al., 2023), history-conditioned diffusion uncertainty (Huang et al., 2024), or progress estimation (Zhang et al., 2025). These methods differ in how they derive rewards from videos, but their reward representations are not explicitly structured to separate task-relevant state changes from extraneous visual variation. As a result, rewards computed from raw observations or generic visual features can respond to camera shifts, lighting changes, or background distractors rather than the state changes that matter for task completion. An RL agent trained on such signals may therefore receive misleading feedback or reinforce correlations unrelated to the intended behavior. This motivates two key requirements for video-based reward learning: evaluating transitions in a task-relevant representation space and providing directional feedback that reflects task performance.

Generic reconstruction and prediction objectives alone do not explicitly isolate the state changes that matter for task execution. Recent work on vision-language-action (VLA) models similarly introduces task-centric latent actions and task-relevant factor specialization to focus representations

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.

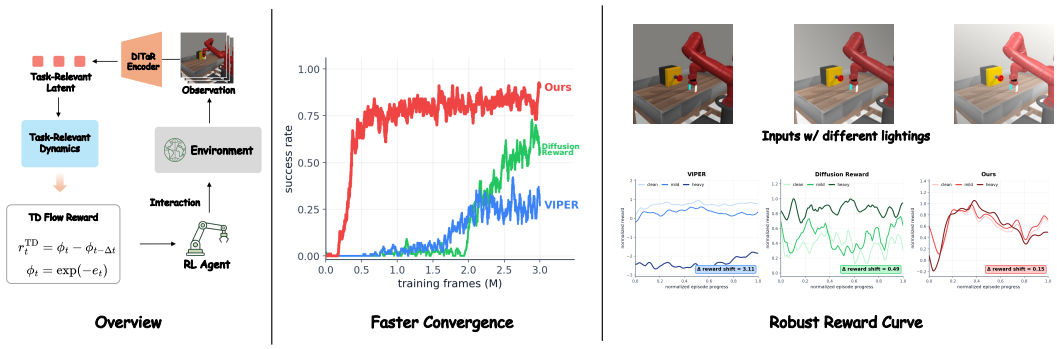


Figure 1: DiTaR learns dense rewards from task-relevant latent dynamics. The DiTaR reward accelerates RL convergence while remaining stable under nuisance visual variations, enabling robust policy learning across diverse tasks.

on information relevant to task execution (Bu et al., 2025b; Zhang et al., 2026; Jia et al., 2026). This issue is especially important for reinforcement learning because errors in the reward representation directly become optimization targets: if task-irrelevant visual variation influences the reward, policy optimization may reinforce these correlations instead of the intended behavior. The latent space used for reward modeling should therefore retain state changes that support task completion while suppressing task-irrelevant visual variation.

A task-relevant representation alone is insufficient for reward construction. Likelihood- or uncertainty-based video rewards measure how well an observation agrees with the learned distribution, but an absolute measure does not necessarily indicate whether the execution is improving relative to its recent history. Progress-based approaches such as ReWiND (Zhang et al., 2025) address directionality by directly predicting task progress, but obtain supervision from predefined progress targets and temporally transformed demonstrations, while the progress estimator operates on generic pretrained visual features. This motivates deriving directional rewards directly from learned task-relevant dynamics without relying on explicit progress supervision from temporal transformations.

To satisfy these requirements, we propose **DiTaR** (shown in Figure 1), a framework that learns dense rewards from videos by modeling task-relevant latent dynamics. DiTaR first uses an asymmetric two-stage representation learning procedure over multi-frame video latents to obtain a continuous task-relevant representation while separating task-irrelevant visual variation. In the first stage, language explicitly provides task semantics, encouraging a capacity-limited latent to encode complementary visual information beyond the language signal. In the second stage, we freeze the first-stage pathway and remove the decoder's direct language conditioning, so that a separate latent must recover from video the task-relevant information previously supplied by language. DiTaR then models task dynamics in this task-relevant latent space with a history- and language-conditioned rectified-flow model. A directional dense reward is derived from how the agreement between observed transitions and the learned task dynamics changes over time, which is evaluated for policy learning on MetaWorld-v3 and ManiSkill.

Our main contributions are summarized as follows:

- We introduce a continuous task-relevant representation learning framework over multi-frame video latents, using an asymmetric two-stage procedure with KL-regularized bottlenecks to yield a task-relevant latent space for reward modeling from action-free videos.
- We model task dynamics in this latent space with a history- and language-conditioned rectified-flow model, and derive a directional dense reward from temporal changes in how observed transitions align with the learned dynamics, without explicit progress supervision.
- We evaluate DiTaR for visual RL on MetaWorldv3 and ManiSkill, and further validate it in real-world offline RL by relabeling robot demonstrations with DiTaR rewards. Reward and representation analyses further demonstrate robustness to nuisance visual variation and sensitivity to task-relevant temporal structure.

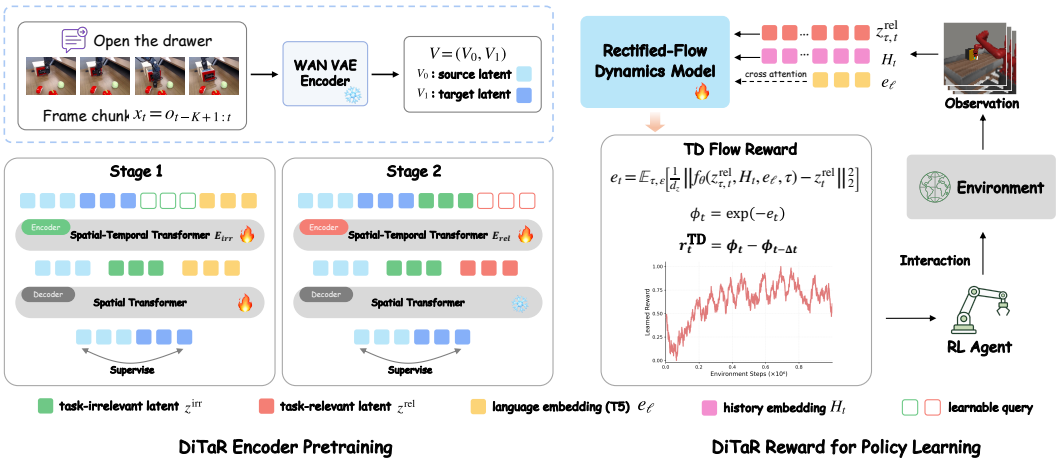


Figure 2: **Pipeline overview.** DiTaR disentangles task-irrelevant and task-relevant latents through an asymmetric two-stage representation learning procedure. A history- and language-conditioned rectified-flow model captures task-relevant transition dynamics, and the temporal difference of its residual-based compatibility score provides a dense reward for RL.

2 Related Work

Learning Rewards from Videos. Prior work learns dense rewards from videos through visual alignment, generative modeling, or progress estimation (Aytar et al., 2018; Sermanet et al., 2018; Dwibedi et al., 2019; Chen et al., 2021; Alakuijala et al., 2023; Yang et al., 2024a). VIP and LIV learn reward-oriented visual representations, while RoboCLIP, VLM-RM, Adapt2Reward, GVL, and VICtoR use visual or vision-language alignment (Ma et al., 2023b;a; Sontakke et al., 2023; Rocamonde et al., 2024; Yang et al., 2024b; Ma et al., 2025; Hung et al., 2025). Generative approaches derive rewards from video likelihood, uncertainty, or generated goals (Escontrela et al., 2023; Huang et al., 2024; Luo et al., 2024; Wang et al., 2026). Recent methods also learn progress-, goal-, or subtask-based rewards, but often operate on raw observations, generated-goal similarity, or generic visual features without explicitly separating task-relevant dynamics from irrelevant visual variation (Alakuijala et al., 2024; Zhang et al., 2025; Lee et al., 2026; Zhai et al., 2025; Chen et al., 2026; Tan et al., 2026; Biza et al., 2025; Ayalew et al., 2025; Kim et al., 2025; Liu et al., 2026).

Task-Relevant Representation. Robotic representation learning has shifted from generic visual features to spaces that capture action-relevant state change. General and self-supervised pretraining methods provide robust features for robot learning and reward specification (Nair et al., 2023; Ma et al., 2023b; Karamcheti et al., 2023; Oquab et al., 2024; Assran et al., 2023; Bardes et al., 2024). Recent work further targets controllable structure by learning latent actions (Ye et al., 2025), including UniVLA (Bu et al., 2025b), which learns vector-quantized task-centric latent actions from paired-frame DINOv2 features for policy learning; other approaches filter task-irrelevant factors (Zhang et al., 2026), ground task factors in VLA policies (Jia et al., 2026), or extract controllable changes through video generation, action grounding, and point tracks (Du et al., 2023; Luo & Du, 2025; Bharadhwaj et al., 2024). Latent-dynamics and world-model methods capture temporal evolution by predicting future latent states or features (Hafner et al., 2023; Hansen et al., 2024; Bruce et al., 2024; Zhou et al., 2025; Assran et al., 2025; Mur-Labadia et al., 2026; Maes et al., 2026; Lyu et al., 2026), motivating our use of task-relevant latent dynamics for video-based reward learning.

3 Methodology

As shown in Figure 2, DiTaR learns a dense reward from action-free videos by modeling task-relevant temporal evolution in a continuous multi-frame video latent space. It first uses an asymmetric two-stage representation learning procedure with KL-regularized bottlenecks to separate task-relevant state changes from task-irrelevant visual variation (Sec. 3.2). DiTaR then models temporal dynamics directly in this task-relevant latent space, learning the conditional distribution of the tar-

get latent given recent visual history and task instruction (Sec. 3.3). The resulting residual measures how well an observed transition agrees with the learned task-conditioned dynamics, and its temporal change provides a directional dense reward for policy learning (Sec. 3.4). Implementation details and pseudocode are provided in Appendices E and G.

3.1 Problem Formulation

We consider an RL agent interacting with a partially observable environment using only RGB observations. At time t, the agent observes $o_t \in \mathcal{O}$ and samples an action $a_t \sim \pi_\omega(\cdot \mid o_t)$. Each task is specified by a language instruction ℓ. We assume an action-free video dataset $\mathcal{D} = \{(\xi^i, \ell^i)\}_{i=1}^N$, where $\xi^i = (o_0^i, \ldots, o_{T_i}^i)$ is a task-execution video paired with its instruction ℓ^i. The videos provide visual observations and language instructions without action labels or privileged environment states.

For reward evaluation at time t, we define the target clip x_t and its preceding visual history h_t as

$$x_t = (o_{t-K+1}, \ldots, o_t), \qquad h_t = (o_{t-K-H+1}, \ldots, o_{t-K}), \tag{1}$$

where K is the target-clip length and H is the history length. Both x_t and h_t contain only observations available up to time t. Let $e_\ell = f_{\text{text}}(\ell)$ denote the language embedding.

DiTaR consists of three learned components. First, the two-stage latent encoders E_{irr} and E_{rel} learn a task-relevant latent z_t^{rel} from the target clip while separating task-irrelevant visual variation. Second, a conditional dynamics model $p_\theta(z_t^{\text{rel}} \mid H_t, e_\ell)$, with $H_t = E_{\text{Wan}}(h_t)$, models task-relevant dynamics from the preceding visual history and task instruction. Finally, the resulting dynamics residual is converted into a dense reward r_t^{TD} for policy learning.

3.2 Task-Relevant Latent Encoding

DiTaR learns continuous task-relevant representations from multi-frame video latents through an asymmetric two-stage procedure. In Stage 1, language supplies task semantics, encouraging a capacity-limited latent to capture complementary visual information for reconstruction. In Stage 2, we freeze this pathway and remove language conditioning from the decoder, so that a second latent must recover the task-relevant information directly from video.

Base Latent Space. We instantiate DiTaR on the latent space of a frozen Wan VAE (Wan et al., 2025). Given a strided 5-frame clip x, the encoder produces source and future video latent tokens $V = (V_0, V_1) = E_{\text{Wan}}(x)$, where V_0 and V_1 denote the source and future latents, respectively. DiTaR operates on these base latents using two spatial-temporal Transformer encoders, E_{irr} and E_{rel}, each equipped with separate learnable query tokens and a KL-regularized variational bottleneck.

Stage 1: Task-Irrelevant Latent. The language embedding e_ℓ is obtained from a frozen T5 encoder (Raffel et al., 2020). Stage 1 trains E_{irr} together with a reconstruction decoder D_ψ, while making e_ℓ explicitly available during both encoding and reconstruction. Using learnable queries Q_{irr}, the encoder produces

$$z^{\text{irr}} \sim q_{\eta_{\text{irr}}}\left(z^{\text{irr}} \mid V, e_\ell\right). \tag{2}$$

The decoder reconstructs the future-target video latent as

$$\hat{V}_1^{(1)} = V_0 + D_\psi\left(V_0, z^{\text{irr}}, e_\ell\right). \tag{3}$$

Because task-level semantics are already supplied through the language pathway, the capacity-limited z^{irr} is encouraged to retain complementary visual information required to reconstruct V_1.

Stage 2: Task-Relevant Latent. Stage 2 removes the decoder's direct access to the language embedding while preserving the representation learned in Stage 1. We freeze both the Stage 1 encoder E_{irr} and reconstruction decoder D_ψ, and introduce a second encoder E_{rel} with learnable queries Q_{rel}. The frozen Stage 1 encoder provides z^{irr}, while E_{rel} produces

$$z^{\text{rel}} \sim q_{\eta_{\text{rel}}}\left(z^{\text{rel}} \mid V, z^{\text{irr}}\right). \tag{4}$$

The fixed decoder then reconstructs the future-target latent using z^{rel} in place of the explicit language conditioning used in Stage 1:

$$\hat{V}_1^{(2)} = V_0 + D_\psi\left(V_0, z^{\text{irr}}, z^{\text{rel}}\right). \tag{5}$$

Because E_{irr} and D_ψ remain fixed, E_{rel} is the only optimized component in Stage 2. It must recover the task-relevant information that was previously supplied directly through language.

Both stages use the same KL-regularized future-target reconstruction objective:

$$\begin{aligned} \mathcal{L}_s &= \mathcal{L}^s_{\text{rec}} + \beta_s \mathcal{L}^s_{\text{KL}}, \qquad s \in \{\text{irr}, \text{rel}\}, \\ \mathcal{L}^s_{\text{rec}} &= \mathbb{E}_{z^s \sim q_s}\left[\frac{1}{d_V}\left\|\hat{V}_1^{(s)} - V_1\right\|_2^2\right], \qquad \mathcal{L}^s_{\text{KL}} = D_{\text{KL}}\left(q_s \,\|\, \mathcal{N}(0, I)\right), \end{aligned} \tag{6}$$

where q_s denotes the corresponding variational posterior, d_V denotes the dimensionality of the flattened target latent, and β_s controls the strength of the KL regularization.

3.3 Conditional Rectified-Flow Dynamics

The encoder maps the observed clip x_t into a task-relevant latent representation but does not model how this representation evolves over time. We therefore model the conditional distribution $p_\theta(z_t^{\text{rel}} \mid H_t, e_\ell)$, where the visual history is encoded by the frozen Wan VAE as $H_t = E_{\text{Wan}}(h_t)$. This conditional distribution captures task-relevant dynamics given the history and task instruction.

We model this distribution with rectified flow (Esser et al., 2024). Given the task-relevant latent z_t^{rel} and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, we construct the linear probability path

$$z_{\tau,t} = (1-\tau) z_t^{\text{rel}} + \tau\epsilon, \qquad \tau \sim \mathcal{U}(\tau_{\min}, 1), \tag{7}$$

where $0 < \tau_{\min} < 1$, with $\tau = 0$ corresponding to the data endpoint and $\tau = 1$ to the noise endpoint.

Rather than directly predicting the flow velocity, DiTaR parameterizes the rectified-flow model by predicting the data endpoint, providing a direct discrepancy in the task-relevant latent space for subsequent reward construction:

$$\hat{z}_{0,t} = f_\theta\left(z_{\tau,t}, H_t, e_\ell, \tau\right). \tag{8}$$

Under Eq. (7), this endpoint parameterization induces the same rectified-flow field as velocity prediction, with endpoint regression corresponding to a τ^2-weighted velocity-regression objective.

We train the dynamics model with

$$\mathcal{L}_{\text{end}} = \mathbb{E}_{t,\tau,\epsilon}\left[\frac{1}{d_z}\left\|\hat{z}_{0,t} - z_t^{\text{rel}}\right\|_2^2\right], \tag{9}$$

where d_z denotes the dimensionality of the task-relevant latent. The resulting endpoint discrepancy measures the deviation between the observed task-relevant latent and the endpoint predicted under the learned conditional distribution, and serves as the residual for reward construction in Sec. 3.4. We instantiate f_θ with an MMDiT (Esser et al., 2024) backbone.

3.4 Temporal-Difference Flow Reward

The endpoint residual provides a measure of how compatible the observed task-relevant latent is with the learned task-relevant dynamics under the visual history and task instruction. Given the observed task-relevant latent z_t^{rel}, we sample $\epsilon \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{U}(\tau_{\min}, \tau_{\max})$, where $0 < \tau_{\min} < \tau_{\max} \leq 1$, and construct $z_{\tau,t}$ using Eq. (7). We define the endpoint residual as

$$e_t = \mathbb{E}_{\tau,\epsilon}\left[\frac{1}{d_z}\left\|f_\theta\left(z_{\tau,t}, H_t, e_\ell, \tau\right) - z_t^{\text{rel}}\right\|_2^2\right]. \tag{10}$$

A smaller e_t indicates greater compatibility between the observed transition and the learned task-conditioned dynamics.

We convert the residual into a bounded compatibility score $\phi_t = \exp(-e_t)$, so that smaller residuals yield larger scores. The absolute compatibility score does not indicate whether task-relevant state evolution is improving over time. We therefore define the dense reward from its temporal change:

$$r_t^{\text{TD}} = \phi_t - \phi_{t-\Delta t} = \exp(-e_t) - \exp(-e_{t-\Delta t}), \tag{11}$$

where Δt is the temporal differencing stride. Positive r_t^{TD} indicates increasing compatibility with the learned task dynamics, providing a directional signal for policy optimization.

Table 1: **Aggregate RL performance on MetaWorld and ManiSkill.** We report final success rate, task coverage, and normalized time-to-threshold. ↑ higher is better, ↓ lower is better.

Benchmark	Method	Final SR ↑	SR≥80 ↑	Reach 0.8 ↑	NormT@0.8 ↓
MetaWorld	Env-only	56.8	50.0	50.0	0.68
	VIPER	39.9	33.3	41.7	0.73
	Diff. Reward	13.1	8.3	8.3	0.92
	ReWiND	65.2	33.3	41.7	0.61
	DiTaR	**94.3**	**91.7**	**91.7**	**0.24**
ManiSkill	Env-only	74.6	50.0	75.0	0.58
	VIPER	40.9	25.0	50.0	0.66
	Diff. Reward	30.8	0.0	25.0	0.88
	ReWiND	55.0	0.0	25.0	0.97
	DiTaR	**87.5**	**100.0**	**100.0**	**0.43**

During policy learning, DiTaR provides this TD reward as a dense shaping signal together with the sparse environment reward and the RND exploration bonus:

$$R_t = \lambda_{\mathrm{D}} \tilde{r}_t^{\mathrm{TD}} + r_t^{\mathrm{env}} + \lambda_{\mathrm{expl}} r_t^{\mathrm{RND}}, \tag{12}$$

where $\tilde{r}_t^{\mathrm{TD}} = (r_t^{\mathrm{TD}} - \mu_D)/\sigma_D$ is the normalized DiTaR reward, and λ_{D} and λ_{expl} control the contributions of the learned reward and RND exploration bonus, respectively.

4 Experiments

4.1 Experimental Setup

Benchmarks and Baselines. We evaluate DiTaR on 16 visual manipulation tasks from two standard simulation benchmarks: MetaWorld-v3 (Yu et al., 2020) and ManiSkill (Gu et al., 2023), including 12 MetaWorld tasks and 4 ManiSkill tasks. We compare against four baselines: **Env-only**, which removes the learned reward term; **VIPER** (Escontrela et al., 2023), which uses conditional video-prediction likelihood as reward; **Diffusion Reward** (Huang et al., 2024), which derives rewards from a history-conditioned video diffusion model; and **ReWiND** (Zhang et al., 2025), which learns a language-conditioned progress estimator using temporally transformed expert videos for supervision. We report task success rate as the primary evaluation metric.

Implementation Details. All methods use the same DrQ-v2 (Yarats et al., 2022) RL pipeline with 64×64 RGB observations, random-shift augmentation, and the same exploration setup, without privileged simulator states. The benchmark reward is retained for all methods, while learned rewards are added as dense shaping signals. VIPER, Diffusion Reward, and ReWiND are trained on expert videos from the corresponding MetaWorld or ManiSkill tasks. DiTaR instead pretrains a single task-relevant encoder and history- & language-conditioned dynamics model on heterogeneous action-free videos, primarily from AgiBotWorld (Bu et al., 2025a), DROID (Khazatsky et al., 2024), and Bridge (Walke et al., 2023), with a small fraction from MetaWorld (McLean et al., 2025) and ManiSkill (Gu et al., 2023). More details are provided in Appendices B and F.

4.2 Main RL Performance

Figure 3 shows the per-task learning curves, and Table 1 summarizes the aggregate performance. Final SR denotes the success rate at the end of training. SR≥80 is the percentage of tasks whose final success rate is at least 80%, while Reach 0.8 is the percentage of tasks that reach 0.8 success at least once during training. NormT@0.8 is the normalized first hitting time to this threshold, assigning the full training budget when the threshold is not reached.

On MetaWorld, DiTaR achieves a final success rate of 94.3%, substantially outperforming the strongest learned-reward baseline, ReWiND (65.2%). It reaches 0.8 success on 91.7% of tasks and achieves a NormT@0.8 of 0.24, compared with 0.61 for ReWiND. On ManiSkill, DiTaR achieves 87.5% final success, exceeding Env-only (74.6%) and ReWiND (55.0%), while attaining 100% Reach 0.8. Its NormT@0.8 of 0.43 is also lower than Env-only (0.58) and ReWiND (0.97). These

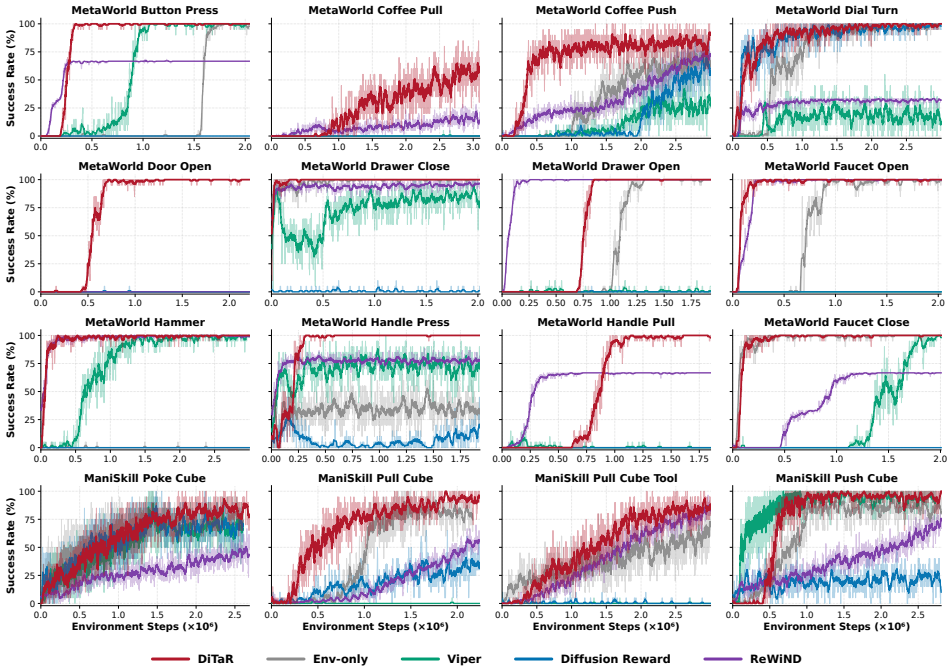


Figure 3: **Main RL performance.** DiTaR achieves strong and consistent policy learning across 12 MetaWorld and 4 ManiSkill tasks, outperforming Env-only, VIPER, Diffusion Reward, and ReWiND in aggregate performance. Panels use task-specific x-axis ranges for readability.

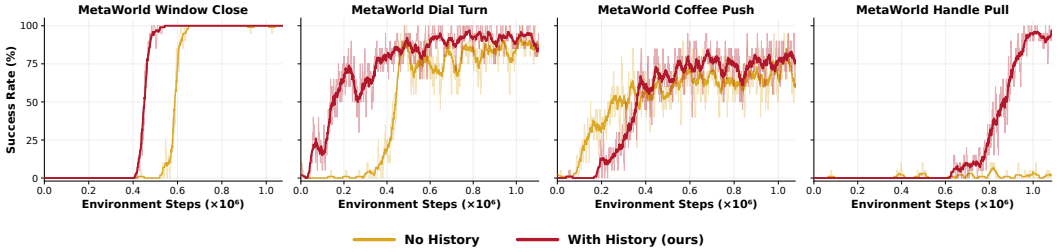


Figure 4: **Ablation on history conditioning.** Removing visual history from the conditional dynamics model reduces learning efficiency and final performance across the evaluated MetaWorld tasks.

results show that DiTaR improves both final policy performance and learning efficiency. The per-task curves in Figure 3 show that DiTaR's aggregate gains come from more consistent learning across the task suite. ReWiND and VIPER perform well on some tasks but degrade substantially on others, whereas DiTaR reaches high success across a broader range of environments with fewer stalled or failed runs. This trend is consistent with the aggregate results in Table 1.

4.3 Ablation Studies

We ablate two designs in DiTaR: conditioning task-relevant dynamics on recent visual history and deriving directional rewards from temporal changes in dynamics compatibility.

History Conditioning. Figure 4 compares DiTaR with a variant that removes visual history from the conditional dynamics model. History conditioning substantially accelerates learning on Window Close and Dial Turn and improves final performance on Coffee Push. The difference is particularly pronounced on Handle Pull, where the no-history variant makes little progress. These results show that history context is important for evaluating task-relevant transitions under the learned dynamics and provides a more effective reward signal for policy learning.

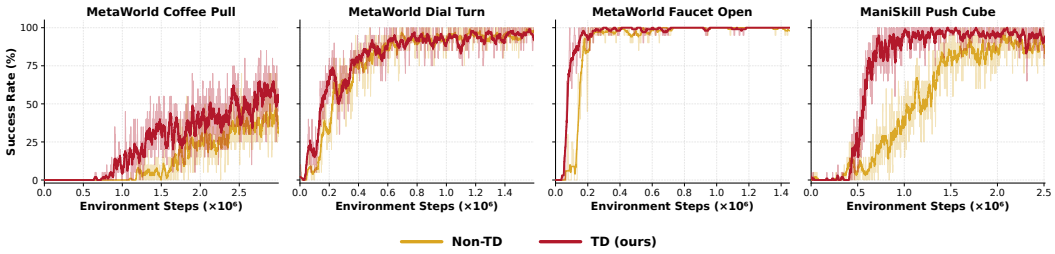


Figure 5: **Ablation on temporal-difference reward.** Temporal differencing improves policy learning over directly using the absolute compatibility score ϕ_t across MetaWorld and ManiSkill tasks.

Table 2: **Reward sensitivity and robustness.** For temporal perturbations, Δ denotes the signed reward change from clean trajectories, with more negative values indicating stronger temporal sensitivity. For visual perturbations, $|\Delta|$ denotes the reward-change magnitude, with smaller values indicating greater robustness. Diff. R.: Diffusion Reward.

	MetaWorld				Real-world			
Perturbation	**DiTaR**	Diff. R.	VIPER	ReWiND	**DiTaR**	Diff. R.	VIPER	ReWiND
(a) Temporal sensitivity: Δ (%) ↓								
Locally shuffled	$\mathbf{-22.2}$	-0.6	-10.2	$+0.1$	-3.2	$+0.1$	$\mathbf{-10.0}$	$+0.2$
Temporal jitter ± 2	$\mathbf{-19.1}$	-0.3	-4.7	$+1.5$	-4.2	-0.2	$\mathbf{-5.5}$	$+0.6$
Shuffled chunks	$\mathbf{-28.9}$	$+0.5$	-9.3	-15.9	-7.0	-1.6	$\mathbf{-11.8}$	$+19.1$
(b) Visual robustness: $\lvert\Delta\rvert$ (%) ↓								
Camera pose shift	**24.3**	42.8	44.6	59.5	**4.2**	32.3	10.2	60.8
Camera yaw 8°	**28.3**	44.7	40.9	55.0	**4.3**	34.7	16.1	30.0
Lighting (heavy)	**19.4**	39.3	49.7	30.1	**1.6**	21.0	29.5	10.2

Temporal-Difference Reward. Figure 5 compares DiTaR with a non-TD variant that directly uses the absolute compatibility score ϕ_t as reward. Temporal differencing improves learning efficiency on both MetaWorld and ManiSkill, with clear gains on Coffee Pull, Faucet Open, and ManiSkill Push Cube, while both variants achieve similar performance on Dial Turn. The compatibility score measures how well the current transition agrees with the learned task dynamics, but does not indicate whether this agreement is increasing over time. Using its temporal change therefore provides directional feedback that is more effective for policy optimization. Appendix D analyzes the task-relevant latent via ablation (94.3% vs. 67.2% final SR) and latent grouping.

4.4 Reward Analysis

We evaluate two complementary properties of the DiTaR reward: sensitivity to task-relevant temporal structure and robustness to task-irrelevant visual variation. Table 2 reports reward changes under temporal and visual perturbations on MetaWorld and real-world expert videos. Figure 6 further shows representative reward trajectories under visual perturbations.

Sensitivity to Temporal Structure. We perturb expert videos using local shuffling, temporal jitter, and shuffled chunks. These perturbations preserve the visual content of individual frames while disrupting the temporal order of task execution. On MetaWorld, DiTaR shows the largest reward decrease under all three perturbations, indicating strong sensitivity to changes in task execution. On real-world videos, DiTaR also consistently assigns lower rewards to temporally corrupted trajectories, although VIPER shows larger decreases. Overall, DiTaR remains sensitive to disruptions in temporal task structure across both settings.

Robustness to Visual Variation. We next perturb camera viewpoint and lighting while keeping the underlying task execution unchanged. In this setting, a desirable reward should remain close to its value on the clean trajectory. DiTaR shows the smallest reward shift across all evaluated visual perturbations on both MetaWorld and real-world videos, as shown in Table 2. Figure 6 shows the same behavior for representative lighting and camera-zoom perturbations: the competing rewards change substantially, whereas DiTaR remains comparatively stable. Together, these results show

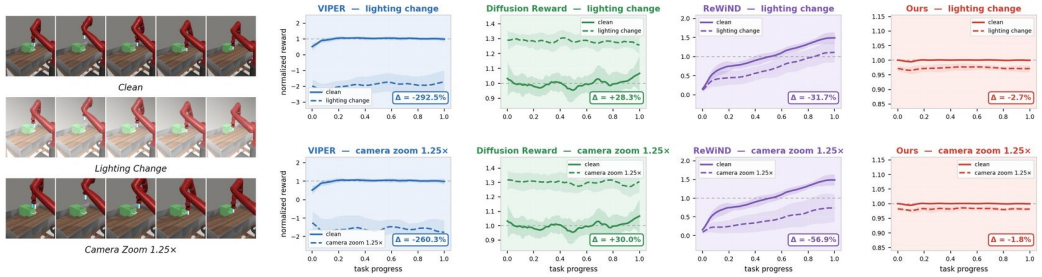


Figure 6: **Reward robustness to visual perturbations.** Per-frame reward trajectories on Meta-World expert rollouts under lighting changes and $1.25\times$ camera zoom. Δ denotes the relative mean reward change from the clean rollout. DiTaR shows substantially smaller reward shifts than VIPER, Diffusion Reward, and ReWiND under both perturbations.

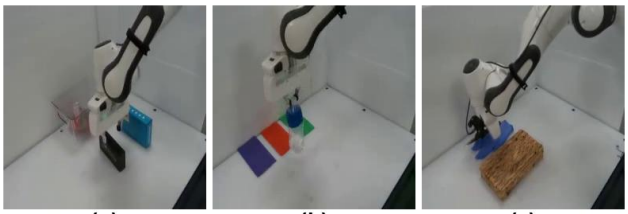


Figure 7: **Real-world task suite.** Three tabletop manipulation tasks used for offline RL evaluation: (a) object-to-basket placement, (b) bottle-to-plate placement, and (c) knife-to-glove placement.

Table 3: **Real-world offline RL results.** All methods relabel the same offline demonstrations and use the same IQL training pipeline. We conduct 10 evaluation trials per task.

Metric	VIPER	Diffusion Reward	ReWiND	**DiTaR**
Success Rate	16.7%	26.7%	43.3%	**83.3%**

that DiTaR responds to changes in temporal task structure while being substantially less affected by task-irrelevant visual variation. Additional analyses of temporal corruptions, corruption localization, viewpoint robustness, and reward evolution during policy training are provided in Appendix C.

4.5 Real-World Offline RL

We further evaluate DiTaR on real-world offline RL using 150 demonstrations collected with a Franka Research 3 robot across three tabletop manipulation tasks (Figure 7). Each demonstration contains synchronized third-person and wrist RGB observations. For each reward model, we relabel the same offline dataset and train a policy using Implicit Q-Learning (IQL) (Kostrikov et al., 2022). Thus, VIPER, Diffusion Reward, ReWiND, and DiTaR differ only in the reward assigned to the fixed dataset, while sharing the same demonstrations and IQL training pipeline. As shown in Table 3, DiTaR achieves an overall success rate of 83.3%, compared with 26.7% for Diffusion Reward, 16.7% for VIPER, and 43.3% for ReWiND.

5 Conclusion and Limitations

DiTaR learns dense rewards by modeling task-relevant latent dynamics and temporal changes in dynamics compatibility. It improves policy learning on MetaWorld-v3 and ManiSkill and transfers to real-world offline RL through reward relabeling. Reward analyses show sensitivity to task-relevant temporal structure and robustness to task-irrelevant visual variation. DiTaR still relies on sparse environment rewards during RL, leaving fully video-based reward learning for future work. Its pre-training data are also dominated by limited-view robot manipulation videos; broader multi-view and egocentric data may further improve generalization.

AI USE STATEMENT

We used generative AI tools to assist with manuscript preparation, including language and figure refinement, code editing and debugging, and discussion of methodological and experimental details. All AI-assisted content was reviewed and revised by the authors, and all technical claims, experimental results, references, and final manuscript content were independently verified by the authors. The authors take full responsibility for the content.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. The work does not involve deployment on humans or automated decision-making about individuals. We use publicly available datasets for research purposes and follow their stated usage terms.

REPRODUCIBILITY STATEMENT

We provide detailed implementation and evaluation information in the Appendix, including the pretraining data and temporal sampling, model architectures and training settings, reward computation, baseline implementations, policy-learning setup, evaluation metrics, and pseudocode. These details are intended to support reproduction of the reported experiments.

REFERENCES

Minttu Alakuijala, Gabriel Dulac-Arnold, Julien Mairal, Jean Ponce, and Cordelia Schmid. Learning reward functions for robotic manipulation by observing humans. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5006–5012. IEEE, 2023.

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. *arXiv preprint arXiv:2405.19988*, 2024.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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*, pp. 10297–10306, 2025.

Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando De Freitas. Playing hard exploration games by watching youtube. *Advances in neural information processing systems*, 31, 2018.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pp. 306–324. Springer, 2024.

Ondrej Biza, Thomas Weng, Lingfeng Sun, Karl Schmeckpeper, Tarik Kelestemur, Yecheng Jason Ma, Robert Platt, Jan-Willem van de Meent, and Lawson LS Wong. On-robot reinforcement learning with goal-contrastive rewards. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4797–4805. IEEE, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025a.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025b.

Annie S. Chen, Suraj Nair, and Chelsea Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.012.

Qianzhong Chen, Justin Yu, Mac Schwager, Pieter Abbeel, Yide Shentu, and Philipp Wu. SARM: Stage-aware reward modeling for long horizon robot manipulation. In *International Conference on Learning Representations*, 2026.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1801–1810, 2019.

Alejandro Escontrela, Ademi Adeniji, Wilson Yan, Ajay Jain, Xue Bin Peng, Ken Goldberg, Youngwoon Lee, Danijar Hafner, and Pieter Abbeel. Video prediction models as rewards for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 12606–12633. PMLR, 2024.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *European Conference on Computer Vision*, 2024.

Kuo-Han Hung, Pang-Chi Lo, Jia-Fong Yeh, Han-Yuan Hsu, Yi-Ting Chen, and Winston Hsu. Victor: Learning hierarchical vision-instruction correlation rewards for long-horizon manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 17550–17577, 2025.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, Qingwen Bu, Haidong Cao, Chao Wu, Qifeng Li, Zhenjie Yang, Chenhe Zhang, Hongyang Li, Zuxuan Wu, Junchi Yan, and Yu-Gang Jiang. GuidedVLA: Specifying Task-Relevant Factors via Plug-and-Play Action Attention Specialization. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.084.

Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-Driven Representation Learning for Robotics. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.032.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

Changyeon Kim, Minho Heo, Doohyun Lee, Honglak Lee, Jinwoo Shin, Joseph Lim, and Kimin Lee. Subtask-aware visual reward learning from segmented demonstrations. In *International Conference on Learning Representations*, volume 2025, pp. 15142–15165, 2025.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

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

Calvin Luo, Mandy He, Zilai Zeng, and Chen Sun. Text-aware diffusion for policy learning. *Advances in Neural Information Processing Systems*, 37:46226–46253, 2024.

Yunhao Luo and Yilun Du. Grounding video models to actions through goal conditioned exploration. In *International Conference on Learning Representations*, volume 2025, pp. 92200–92232, 2025.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Cui Wenbo, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling Latent Dynamics Action Model via Universal Embodied Data Ingestion. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.210.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 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 *The Eleventh 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, et al. Vision language models are in-context value learners. In *International Conference on Learning Representations*, volume 2025, pp. 33984–34009, 2025.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K. R. Zentner, Ryan Julian, J K Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Meta-World+: An improved, standardized, rl benchmark. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pp. 892–909. PMLR, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

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. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Cheng Chi, Yuheng Ji, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. General process reward modeling for robotic reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22412–22422, 2026.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Qi Wang, Mian Wu, Yuyang Zhang, Mingqi Yuan, Wenyao Zhang, Haoxiang You, Yunbo Wang, Xin Jin, Xiaokang Yang, and Wenjun Zeng. Goal-driven reward by video diffusion models for reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Daniel Yang, Davin Tjia, Jacob Berg, Dima Damen, Pulkit Agrawal, and Abhishek Gupta. Rank2reward: Learning shaped reward functions from passive video. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2806–2813. IEEE, 2024a.

Yanting Yang, Minghao Chen, Qibo Qiu, Jiahao Wu, Wenxiao Wang, Binbin Lin, Ziyu Guan, and Xiaofei He. Adapt2reward: Adapting video-language models to generalizable robotic rewards via failure prompts. In *European Conference on Computer Vision*, pp. 163–180. Springer, 2024b.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

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*, pp. 1094–1100, 2020.

Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Fuxian Huang, Haoran Zhang, Ming Zhou, Shengzhe Zhang, Litao Liu, Sixu Lin, and Jiangmiao Pang. A vision-language-action-critic model for robotic real-world reinforcement learning. *arXiv preprint arXiv:2509.15937*, 2025.

Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. CLAP: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. ReWiND: Language-guided rewards teach robot policies without new demonstrations. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 460–488. PMLR, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135, 2025.

A Overview

This supplementary material provides additional experimental analysis and implementation details for DiTaR. Sec. B first describes the pretraining corpus, including the data sources, camera views, and temporal sampling used to train the task-relevant encoder and conditional dynamics model. Sec. C then extends the reward analysis from the main paper with temporal corruption tests, localization of corrupted segments, robustness to viewpoint changes, and reward evolution during policy training. Sec. D studies the task-relevant representation through policy-learning ablations and cross-dataset latent grouping. Sec. E gives the model and reward implementation details, including the latent encoders, conditional rectified-flow model, and reward computation. Sec. F describes the online RL and real-world IQL settings, baseline implementations, and evaluation metrics. Finally, Sec. G gives pseudocode for representation learning, dynamics training, and online policy learning.

B Training Data Details

Table 4 summarizes the action-free video data used for DiTaR pretraining. We train the task-relevant encoder and history- & language-conditioned dynamics model on a mixture of real-world robot datasets and simulated manipulation environments. The pretraining corpus is dominated by real-world robot videos from AgiBotWorld (Bu et al., 2025a), DROID (Khazatsky et al., 2024), and Bridge (Walke et al., 2023), which account for 38.0%, 38.0%, and 19.0% of the data, respectively. These datasets cover different robot embodiments, manipulation behaviors, and camera viewpoints, including head-mounted and exterior views. We additionally include MetaWorld (McLean et al., 2025) and ManiSkill (Gu et al., 2023), each contributing 2.5% of the pretraining data.

Across all sources, DiTaR uses RGB observations and paired language instructions without action labels or other embodiment-specific annotations. This allows the same representation and dynamics objectives to be applied across datasets with different robots and action spaces. Simulated videos account for a small fraction of pretraining; most videos come from real-world robot datasets.

The native frame rates differ across datasets, so we sample clips with dataset-specific frame strides to obtain comparable temporal coverage. We draw strides from $\{1, 2, 3, 4\}$ for AgiBotWorld and DROID, $\{1, 2, 3\}$ for Bridge, $\{1, 2\}$ for MetaWorld, and use a fixed stride of 1 for ManiSkill. Sampling is biased toward the largest stride in each range to include longer-horizon temporal context while retaining shorter transitions.

Table 4: Video sources, proportions, camera views, and frame strides used for DiTaR pretraining.

Dataset	Ratio (%)	View	Frame stride
AgiBotWorld (Bu et al., 2025a)	38.0	Head	$\{1, 2, 3, 4\}$
DROID (Khazatsky et al., 2024)	38.0	Exterior	$\{1, 2, 3, 4\}$
Bridge (Walke et al., 2023)	19.0	Exterior	$\{1, 2, 3\}$
MetaWorld (McLean et al., 2025)	2.5	Left	$\{1, 2\}$
ManiSkill (Gu et al., 2023)	2.5	Default	1

C Reward Analysis

We examine how the DiTaR reward responds to task-relevant temporal changes and task-irrelevant visual variation. Temporal perturbations disrupt the order or continuity of task execution, whereas visual perturbations alter appearance while preserving the underlying behavior. A desirable reward should respond to the former while remaining stable to the latter. We further examine how the DiTaR reward evolves during policy learning.

C.1 Temporal Sensitivity on Bridge and Real-World Videos

We apply backward drift, looped segments, shuffled chunks, and duplicate frames to expert videos. These perturbations alter the sequence without changing the appearance of the individual frames. Shuffled chunks disrupt the order of execution, while duplicate frames introduce repetition without

new state changes. The comparison tests whether a reward responds to these temporal changes even though the observations still come from expert videos.

Table 5: **Temporal sensitivity.** Signed reward changes Δ (%) over 16 expert episodes per dataset. More negative values indicate stronger temporal sensitivity. The largest decrease per row and dataset is bold. Diff. R.: Diffusion Reward.

	Bridge			Real-world		
Corruption	DiTaR	Diff. R.	VIPER	DiTaR	Diff. R.	VIPER
Signed reward change: Δ (%) $\downarrow$						
Backward drift	$\mathbf{-2.4}$	-1.1	-0.5	$\mathbf{-1.6}$	$+7.0$	$+0.6$
Looped segment	$\mathbf{-3.5}$	-0.6	-1.7	$\mathbf{-1.0}$	$+7.4$	$+0.7$
Shuffled chunks	$\mathbf{-16.8}$	-0.7	-6.0	-9.2	-2.5	$\mathbf{-11.8}$
Duplicate frames	$\mathbf{-21.1}$	$+0.9$	$+7.0$	$\mathbf{-26.4}$	$+6.8$	$+5.3$

Table 5 shows that DiTaR's reward decreases under every corruption in both datasets. On Bridge, the decreases are largest for shuffled chunks and duplicate frames, at -16.8% and -21.1%, respectively. Backward drift and looped segments produce smaller decreases. On real-world videos, DiTaR also gives a lower reward in all four cases, although VIPER shows a larger decrease for shuffled chunks.

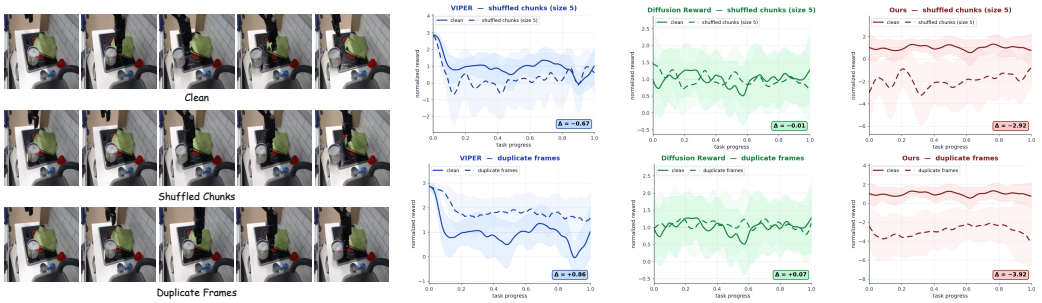


Figure 8: **Temporal corruption on Bridge.** Clean and corrupted rewards over 16 episodes. Bands indicate one standard deviation; Δ is the mean signed reward shift.

Fig. 8 shows how these differences appear over time. DiTaR assigns lower rewards to both corrupted sequences across much of the episode. VIPER also responds to shuffled chunks, but its reward increases under frame duplication; Diffusion Reward changes little in either case. Frame duplication is informative because it preserves the appearance of expert observations but removes continued state changes. DiTaR's lower reward reflects its sensitivity to these missing changes.

C.2 Temporal Corruption Localization

A decrease in the mean reward does not indicate when the model responds to a temporal perturbation. We therefore apply temporal jitter or local shuffling only to the middle third of each video, leaving the remaining frames unchanged. Localization AUC measures whether the clean-to-corrupted reward decrease distinguishes affected from unaffected output positions. For DiTaR, affected positions include clips whose temporal support overlaps the corrupted interval.

Table 6 reports mean localization AUCs. DiTaR localizes both perturbations on MetaWorld and ManiSkill, with higher AUCs for local shuffling than for temporal jitter in both datasets. VIPER also localizes the MetaWorld corruptions. These results measure whether reward changes identify the temporally affected interval. Fig. 9 shows representative examples from each dataset.

In the MetaWorld example, DiTaR's clean reward remains high through the middle third, while temporal jitter introduces several distinct decreases within the perturbed interval. The clean and corrupted curves largely coincide before the perturbation and approach one another again afterward. VIPER also responds near the altered segment, while Diffusion Reward and ReWiND show smaller

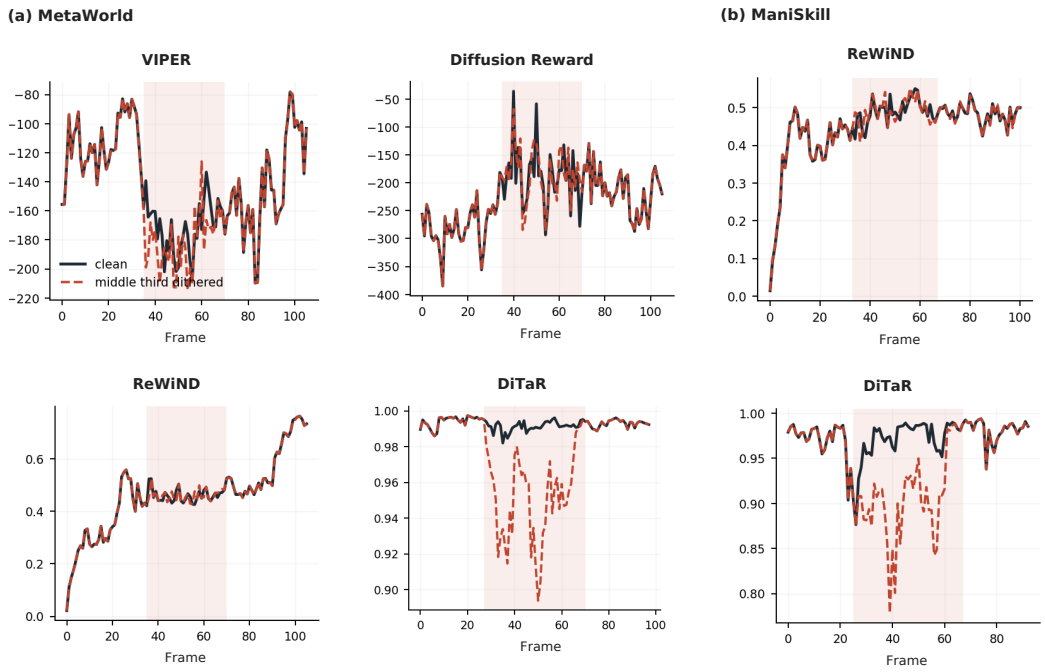


Figure 9: **Temporal corruption localization.** Temporal jitter of ± 2 frames is applied to the shaded middle third. Left two columns: MetaWorld, "take the object out of the hole." Right: ManiSkill, "push the red T-shaped block into place." Solid and dashed curves show clean and corrupted rewards.

Table 6: **Temporal corruption localization.** Mean localization AUCs on MetaWorld and ManiSkill; higher values indicate better localization.

Dataset	Method	Temporal jitter ↑	Local shuffling ↑
MetaWorld	VIPER	0.91	0.96
MetaWorld	DiTaR	0.95	1.00
ManiSkill	DiTaR	0.83	0.94

differences. The localization AUCs for this video are 0.76, 0.47, 0.39, and 0.98 for VIPER, Diffusion Reward, ReWiND, and DiTaR, respectively.

The ManiSkill example shows the same behavior. DiTaR's reward decreases near the perturbed interval and recovers afterward, whereas the two ReWiND curves remain similar throughout. Their localization AUCs are 0.94 and 0.46, respectively.

C.3 Visual Robustness

Viewpoint changes. We test whether viewpoint changes alter the DiTaR reward when the underlying task execution is unchanged. On ManiSkill, we apply synthetic camera yaw angles of $4°$, $8°$, $16°$, $24°$, and $32°$, extending the fixed-angle comparison in the main paper. We also compare two rendered camera views of the same rollout.

Fig. 10 shows smaller relative reward changes for DiTaR than ReWiND throughout the tested yaw range. ReWiND's change is $10.7\times$ that of DiTaR at $24°$ and $6.6\times$ at $32°$. Table 7 further shows that DiTaR has the smaller change in 82–86% of videos at the three smaller angles and in all evaluated videos at the two larger angles.

Table 7: **Viewpoint robustness on ManiSkill.** Relative reward-change ratio (ReWiND/DiTaR) and percentage of videos with a smaller reward change for DiTaR.

Camera yaw	4°	8°	16°	24°	32°
Relative change ratio	3.8×	5.2×	4.8×	10.7×	6.6×
Smaller DiTaR change (% of videos)	82	86	86	100	100

For the second-camera comparison, the aggregate reward changes are 3.2% for DiTaR and 18.5% for ReWiND, further showing that DiTaR is less affected when the same execution is observed from a different viewpoint.

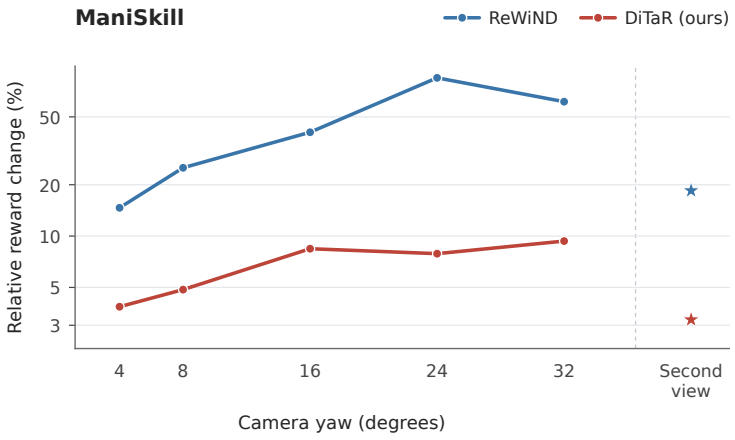


Figure 10: **Viewpoint robustness on ManiSkill.** Relative reward changes under synthetic camera yaw (circles) and a second rendered view of the same rollout (stars). The vertical axis is logarithmic; smaller changes indicate greater robustness.

Reward trajectories. Fig. 11 compares complete reward trajectories for a MetaWorld camera-zoom example and a ManiSkill second-view example.

Full reward trajectories also show how visual perturbations affect the reward over time. In the MetaWorld example, camera zoom shifts VIPER's reward downward and Diffusion Reward's reward upward across much of the episode. ReWiND retains an increasing trend, but its perturbed reward remains below the clean reward. DiTaR's two trajectories largely overlap. The relative mean reward changes are 314.1%, 33.5%, 42.2%, and 1.4% for VIPER, Diffusion Reward, ReWiND, and DiTaR.

In the ManiSkill example, switching cameras lowers ReWiND's reward over most of the rollout, while DiTaR preserves a similar trajectory. Their relative mean reward changes are 29.0% and 0.4%, respectively. Together with the localization results, these examples show that DiTaR responds to temporal disruptions while remaining comparatively stable when the same execution is viewed from another camera.

C.4 Reward Evolution during Policy Learning

We next examine how the DiTaR reward evolves as policy learning progresses. For each task, we compare rewards at the same task-specific timestep across different stages of training.

Fig. 12 shows an overall positive trend on all 12 MetaWorld-v3 tasks. Although the reward is not monotonic on every task, later stages generally receive higher rewards. These results complement the perturbation analyses by showing that the DiTaR reward also tracks improvements in policy behavior during training.

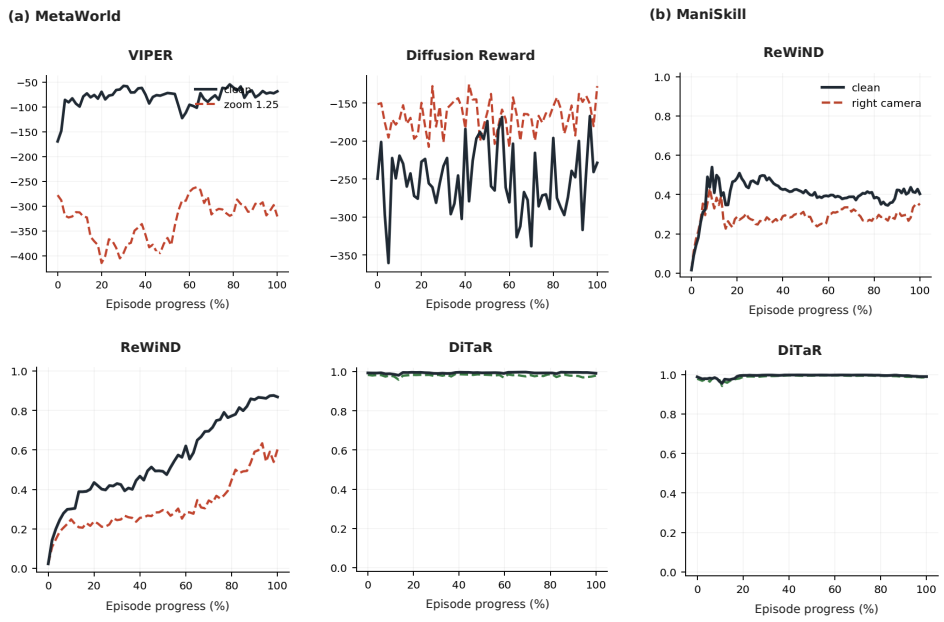


Figure 11: **Rewards under visual perturbations.** Left two columns: MetaWorld, "press the button," under $1.25\times$ camera zoom. Right column: ManiSkill, "push the red T-shaped block into place," under a second camera view. Solid and dashed curves show clean and perturbed rewards.

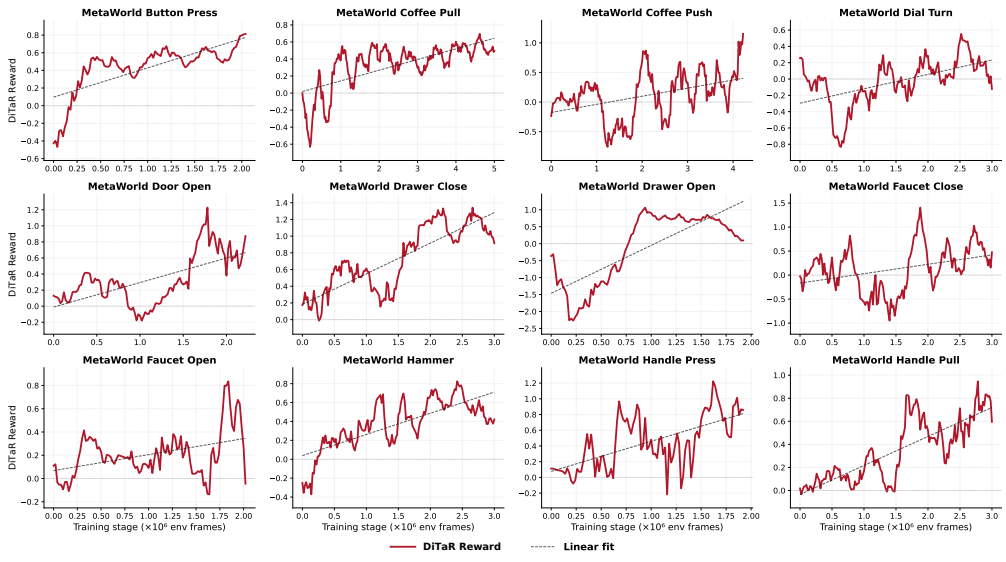


Figure 12: **Reward evolution during policy learning.** DiTaR rewards at a fixed task-specific timestep across training stages on 12 MetaWorld-v3 tasks. Solid curves show smoothed traces and dashed lines show linear fits.

D Ablation and Latent Analysis

D.1 Task-Relevant Latent Ablation

We compare DiTaR's task-relevant latent encoding with a single-stage reconstruction-based encoder trained without language conditioning. The ablated encoder compresses frozen Wan VAE latents, and its output replaces z_t^{rel} as the target for dynamics modeling. We retrain the history- and

language-conditioned rectified-flow model on these latents using the settings in Table 10. The TD reward formulation, reward normalization, and downstream RL settings remain unchanged.

Table 8 shows that DiTaR improves final success from 67.2% to 94.3% and reduces NormT@0.8 from 0.458 to 0.244. SR≥80 increases from 58.3% to 91.7%. The per-task curves in Fig. 13 show substantial differences on Coffee Pull, Drawer Open, and Handle Pull, where the reconstruction-based variant learns more slowly or does not reach high success within the plotted interval. These results support using the task-relevant representation for reward modeling.

Table 8: Task-relevant latent ablation on MetaWorld-v3, with baseline results for comparison. Metrics are defined in Sec. F.3.

Method	Final SR (%) ↑	SR≥80 (%) ↑	Reach 0.8 (%) ↑	NormT@0.8 ↓
Env-only	56.8	50.0	50.0	0.678
VIPER	39.9	33.3	41.7	0.734
Diffusion Reward	13.1	8.3	8.3	0.922
ReWiND	65.2	33.3	41.7	0.609
w/o TR latents	67.2	58.3	66.7	0.458
DiTaR	**94.3**	**91.7**	**91.7**	**0.244**

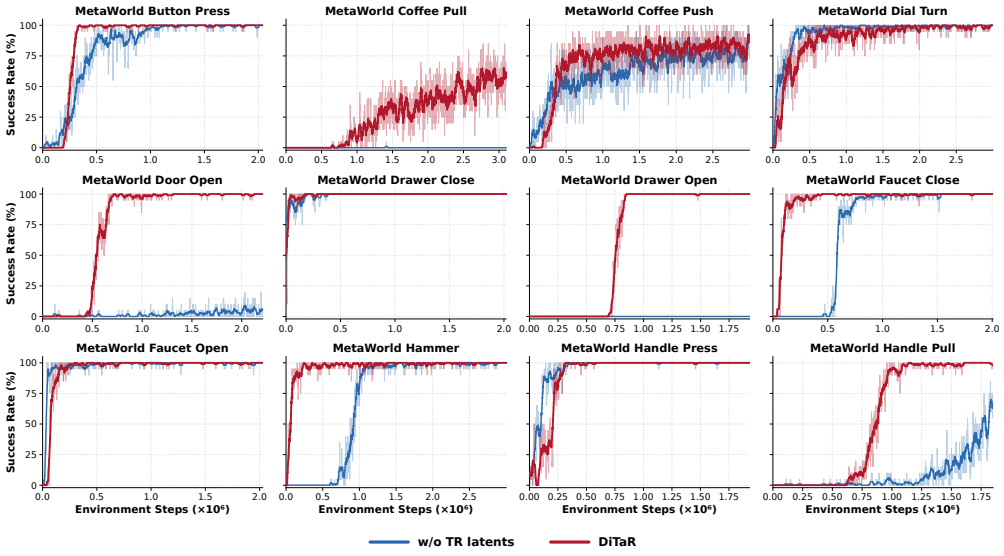


Figure 13: **Task-relevant latent ablation.** Learning curves on 12 MetaWorld-v3 tasks comparing DiTaR's task-relevant latents with latents from a single-stage reconstruction-based encoder trained without language conditioning. Both variants use the same conditional rectified-flow architecture and TD reward formulation, with the dynamics model trained separately for each latent representation. DiTaR achieves higher overall success and learns more efficiently.

D.2 History and Temporal-Difference Ablations

Without history conditioning. We remove the encoded five-frame history from the dynamics model while retaining the task-relevant target, language conditioning, architecture size, training objective, and optimization schedule. Reward inference, normalization, and policy training otherwise follow the full model. The learning curves are shown in Fig. 4 of the main paper.

Without temporal differencing. We keep the latent encoders and conditional dynamics model fixed and replace $r_t^{\mathrm{TD}} = \phi_t - \phi_{t-\Delta t}$ with $r_t^{\mathrm{nonTD}} = \phi_t = \exp(-e_t)$. All other settings are unchanged. This comparison tests the use of temporal changes in compatibility against its absolute value; the results are shown in Fig. 5 of the main paper.

D.3 Cross-Dataset Latent Grouping

For each five-frame clip, we mean-pool its task-relevant latent over the temporal and query axes, ℓ_2-normalize the resulting descriptor, and compute cosine similarity between clips. Each group in Fig. 14 contains only clips with pairwise similarity of at least 0.90. The groups show placing and picking behaviors across datasets, with different robots, viewpoints, backgrounds, and objects. These examples illustrate shared manipulation behaviors in the learned latent space.

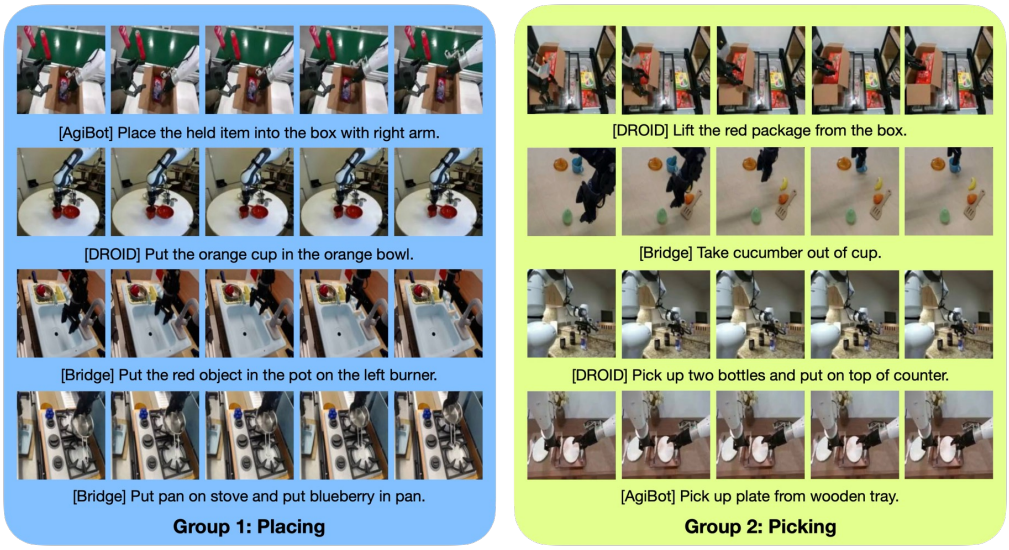


Figure 14: **Cross-dataset latent grouping.** Groups of five-frame clips with pairwise cosine similarity of at least 0.90 in the task-relevant latent space. The left and right groups show placing and picking behaviors across different robots and visual settings.

E Model and Reward Implementation

E.1 Task-Relevant Latent Encoder

We use the frozen Wan-2.2 TI2V-5B VAE (Wan et al., 2025) to encode each strided five-frame clip into video latents $V = (V_0, V_1)$. The encoders E_{irr} and E_{rel} have separate learnable queries and KL-regularized variational bottlenecks. Task instructions are embedded by a frozen T5-base encoder (Raffel et al., 2020).

In Stage 1, we train E_{irr} and decoder D_ψ to reconstruct V_1 from V_0, z^{irr}, and the language embedding e_ℓ. In Stage 2, both E_{irr} and D_ψ remain frozen. We train only E_{rel}, which takes V and z^{irr} as input and produces z^{rel}. The fixed decoder receives z^{rel} in place of its direct language input. With the Stage 1 pathway fixed, E_{rel} learns to recover from video the task-relevant information previously supplied to the decoder through language. The bottleneck tokens are projected to the decoder conditioning dimension before reconstruction. Table 9 lists the architecture and training settings.

E.2 Conditional Rectified-Flow Model

The dynamics model f_θ predicts the clean task-relevant latent from its noisy input, the visual history, and the task instruction. A five-frame history is encoded by the frozen Wan VAE into two latent frames and flattened into $2 \times 256 = 512$ history tokens. History and target tokens are separately projected to the common model dimension ($d = 512$) before being concatenated for self-attention in each DiT block. Language conditioning is applied to target tokens through gated cross-attention, with the scalar gate initialized to zero. The Wan VAE and both latent encoders remain frozen during dynamics training. Table 10 reports the model settings.

Table 9: Architecture and training settings for the latent encoders. The pretrained Wan VAE and T5-base text encoder are frozen throughout representation learning.

Hyperparameter	Value
Input latent dimension	48
Frames per training window	5
Model dimension	768
Attention heads	12
Encoder blocks	4
Decoder blocks	12
Query blocks	8
Query tokens	4
Bottleneck dimension	32
Bottleneck regularizer	KL
Separate task-relevant encoder	Yes
Text encoder	Frozen T5-base; dimension 768; maximum length 64
Optimizer	AdamW
Batch size	128 per GPU
Training steps	$\sim 1.0 \times 10^4$
Training cost	8 GPUs $\times$ $\sim$6 hours $=\sim$48 GPU-hours

Table 10: Architecture and training settings for the conditional rectified-flow model.

Hyperparameter	Value
History input	Five-frame history encoded by the Wan VAE
History tokens / token dimension	512 / 48
Target input	Task-relevant latent of the target clip
Target tokens / token dimension	4 / 32
Text conditioning	Target-only cross-attention; scalar gate initialized to 0
Model dimension	512
DiT blocks	12
Attention heads	8
Timestep conditioning	Sinusoidal embedding and AdaLN modulation
Training objective	Endpoint regression (x_0 parameterization)
History dropout / text dropout	0.0 / 0.15
Optimizer	AdamW
Batch size	128 per GPU
Training steps	$\sim 6 \times 10^4$
Training cost	8 GPUs $\times$ $\sim$24 hours $=\sim$192 GPU-hours

E.3 Reward Computation

For each observed target clip, we extract z_t^{rel} and encode its preceding history as $H_t = E_{\mathrm{Wan}}(h_t)$. We evaluate the endpoint residual using the frozen dynamics model,

$$e_t = \mathbb{E}_{\tau,\epsilon}\left[\frac{1}{d_z}\left\|f_\theta(z_{\tau,t}, H_t, e_\ell, \tau) - z_t^{\mathrm{rel}}\right\|_2^2\right], \qquad z_{\tau,t} = (1-\tau)z_t^{\mathrm{rel}} + \tau\epsilon, \tag{13}$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\tau \sim \mathcal{U}(\tau_{\min}, \tau_{\max})$. The compatibility score and TD reward are

$$\phi_t = \exp(-e_t), \qquad r_t^{\mathrm{TD}} = \phi_t - \phi_{t-\Delta t}. \tag{14}$$

We compute reward normalization statistics before policy training and use $\widetilde{r}_t^{\mathrm{TD}} = (r_t^{\mathrm{TD}} - \mu_D)/\sigma_D$ in the RL objective. The temporal gap is $\Delta t = 4$. We use two Monte Carlo samples to estimate the expectation in Eq. (13), with the same sampled (τ, ϵ) pairs used for ϕ_t and $\phi_{t-\Delta t}$.

F Policy Learning and Evaluation

F.1 Online Visual RL

Our online RL experiments use the DrQ-v2 (Yarats et al., 2022) pipeline from Diffusion Reward (Huang et al., 2024). The policy and critic architectures, observation size, image augmentation,

replay buffer, exploration schedule, RND module, and environment interface are shared across all methods. In particular, Env-only and all learned-reward baselines use the same RND exploration setup; during policy learning, the learned-reward methods differ only in the learned reward signal. Learned rewards are computed before transitions enter the replay buffer.

The policy is trained with

$$R_t = \lambda_D \widetilde{r}_t^{\text{learn}} + r_t^{\text{env}} + \lambda_{\text{expl}} r_t^{\text{RND}}, \tag{15}$$

where $\widetilde{r}_t^{\text{learn}}$ is the normalized learned reward, r_t^{env} is the sparse benchmark reward, and r_t^{RND} is the RND exploration bonus. VIPER, Diffusion Reward, ReWiND, and DiTaR use the same reward-normalization procedure: normalization statistics are computed before policy learning and kept fixed throughout training. DiTaR uses $\widetilde{r}_t^{\text{TD}}$ for the learned term. For learned-reward methods, the task-specific $(\lambda_D, \lambda_{\text{expl}})$ pairs are shared across methods; Env-only sets $\lambda_D = 0$ while retaining the same RND exploration setting. Tables 11 and 12 list the RL settings and reward coefficients, respectively. Algorithm 2 summarizes their use during training.

Table 11: Online visual RL settings for MetaWorld-v3 and ManiSkill.

Hyperparameter	Value
Environment	
Action repeat	1
Frame stack	1
Observation size	$64 \times 64 \times 3$
Episode length	250 environment steps
Per-task training budget	3×10^6 environment frames
DrQ-v2	
Data augmentation	Random shift, ± 4 pixels
Replay buffer capacity	10^6
Discount γ	0.99
n-step returns	3
Initial exploration frames	4,000
Feature dimension	50
Hidden dimension	1024
Exploration standard-deviation clip	0.3
Exploration standard-deviation schedule	Linear$(1.0, 0.1, 3 \times 10^6)$
Soft update rate	0.01
Optimizer	Adam, learning rate 10^{-4}
Batch size	256
Update frequency	2
RND exploration	
CNN feature dimension	$7 \times 7 \times 64$
MLP size	512×512
Learning rate	10^{-4}
DiTaR reward	
Reward formulation	TD flow reward
Temporal gap Δt	4
Paired Monte Carlo samples	2
Sampling steps	16
Reward and exploration coefficients	Table 12

F.2 Reward Baselines

VIPER, Diffusion Reward, and ReWiND are trained on expert videos from the corresponding Meta-World or ManiSkill tasks. We reproduce the reward models for all learned-reward baselines following their official released implementations. All methods use the online RL pipeline described above.

Env-only. We set $\lambda_D = 0$, retaining the sparse benchmark reward and RND exploration bonus without a learned reward model.

Table 12: Task-specific reward and exploration coefficients $(\lambda_D, \lambda_{\text{expl}})$ in Eq. (15), shared across learned-reward methods. Following the reward-mixing setup of Diffusion Reward (Huang et al., 2024), we use task-specific reward and exploration coefficients. Tasks sharing the same coefficient pair are grouped in a single row.

Benchmark	$(\lambda_D, \lambda_{\text{expl}})$	Tasks
MetaWorld-v3	$(0.05, 0.95)$	`button-press`, `coffee-pull`, `coffee-push`, `door-open`, `drawer-open`, `faucet-close`, `faucet-open`, `hammer`, `handle-press`, `handle-pull`
	$(0.95, 0.05)$	`drawer-close`
	$(0.50, 0.50)$	`dial-turn`
ManiSkill	$(0.05, 0.95)$	`PullCube`, `PushCube`
	$(0.95, 0.05)$	`PokeCube`
	$(0.50, 0.05)$	`PullCubeTool`

VIPER. We retrain VideoGPT on the task-specific expert videos and use conditional video likelihood as reward (Escontrela et al., 2023). Reward inference follows the original teacher-forcing implementation and uses the original model hyperparameters.

Diffusion Reward. We retrain VQ-Diffusion on the same expert videos used for VIPER and use its history-conditioned uncertainty reward (Huang et al., 2024). We retain the original reward formulation and evaluate it in the shared RL pipeline.

ReWiND. We train a language-conditioned progress estimator on the corresponding expert videos, using temporal transformations to construct progress supervision (Zhang et al., 2025). The resulting learned reward is evaluated in the same DrQ-v2 pipeline.

F.3 Evaluation Metrics

We use the aggregate metrics defined in the main paper. Final SR is the mean final success rate. SR$\geq$80 is the percentage of tasks with final success rate at least 80%. Reach 0.8 is the percentage of tasks whose rolling-window smoothed success rate reaches 0.8 at least once during training. NormT@0.8 is the mean first hitting time divided by the corresponding training budget, assigning a value of 1 when the threshold is not reached.

F.4 Real-World Offline RL

The real-world evaluation uses 150 demonstrations collected with a Franka Research 3 robot across the three tasks shown in Fig. 7. The recordings contain synchronized third-person and wrist RGB observations. For each reward model, we relabel the same offline transitions and train a policy with Implicit Q-Learning (IQL) (Kostrikov et al., 2022). The demonstrations, policy architecture, and IQL settings are shared across reward models. We conduct 10 evaluation trials per task; results are reported in Table 3 of the main paper.

G Algorithms

Algorithm 1 summarizes the two-stage representation learning and conditional dynamics training. Algorithm 2 shows how the frozen reward model is used during online policy learning.

Algorithm 1 DiTaR Representation and Dynamics Training

Require: Video–instruction dataset $\mathcal{D}$; frozen E_{Wan} and f_{text}
Ensure: Trained E_{irr}, E_{rel}, f_θ; reward statistics (μ_D, σ_D)

Stage 1: task-irrelevant latent encoding

Initialize E_{irr} and reconstruction decoder D_ψ
while not converged **do**
  Sample clip and instruction (x, ℓ) from $\mathcal{D}$
  $V = (V_0, V_1) \leftarrow E_{\mathrm{Wan}}(x)$; $e_\ell \leftarrow f_{\mathrm{text}}(\ell)$
  Sample $z^{\mathrm{irr}} \sim q_{\eta_{\mathrm{irr}}}(\cdot \mid V, e_\ell)$
  $\widehat{V}_1^{(1)} \leftarrow V_0 + D_\psi(V_0, z^{\mathrm{irr}}, e_\ell)$
  Update E_{irr} and D_ψ using $\mathcal{L}_{\mathrm{irr}}$
end while

Stage 2: task-relevant latent encoding

Freeze E_{irr} and D_ψ; initialize E_{rel}
while not converged **do**
  Sample (x, ℓ); compute $V = (V_0, V_1)$ and e_ℓ
  Extract z^{irr} with the frozen E_{irr}
  Sample $z^{\mathrm{rel}} \sim q_{\eta_{\mathrm{rel}}}(\cdot \mid V, z^{\mathrm{irr}})$
  $\widehat{V}_1^{(2)} \leftarrow V_0 + D_\psi(V_0, z^{\mathrm{irr}}, z^{\mathrm{rel}})$
  Update only E_{rel} using $\mathcal{L}_{\mathrm{rel}}$
end while
Freeze E_{rel}

Conditional rectified-flow dynamics

Initialize the endpoint predictor f_θ
while not converged **do**
  Sample target clip, preceding history, and instruction (x_t, h_t, ℓ)
  $H_t \leftarrow E_{\mathrm{Wan}}(h_t)$; $e_\ell \leftarrow f_{\mathrm{text}}(\ell)$
  Extract z_t^{rel} from x_t using the frozen encoders and e_ℓ
  Sample $\tau \sim \mathcal{U}(\tau_{\min}, 1)$ and $\epsilon \sim \mathcal{N}(0, I)$
  $z_{\tau,t} \leftarrow (1-\tau) z_t^{\mathrm{rel}} + \tau\epsilon$
  Update f_θ with $\frac{1}{d_z}\| f_\theta(z_{\tau,t}, H_t, e_\ell, \tau) - z_t^{\mathrm{rel}} \|_2^2$
end while
Freeze f_θ and compute (μ_D, σ_D) on the pretraining videos

Algorithm 2 Online Policy Learning with DiTaR Rewards

Require: Frozen DiTaR model; task instruction ℓ; normalization statistics (μ_D, σ_D); environment $\mathcal{E}$
Ensure: Trained policy π_ω

Initialize π_ω, replay buffer $\mathcal{B}$, and RND predictor
$e_\ell \leftarrow f_{\mathrm{text}}(\ell)$
while not converged **do**
  Sample $a_t \sim \pi_\omega(\cdot \mid o_t)$, step $\mathcal{E}$, and update the visual history
  Form the current and previous reward clips and their preceding histories
  Evaluate e_t and $e_{t-\Delta t}$ using Eq. (13)
  $r_t^{\mathrm{TD}} \leftarrow \exp(-e_t) - \exp(-e_{t-\Delta t})$
  $\widetilde{r}_t^{\mathrm{TD}} \leftarrow (r_t^{\mathrm{TD}} - \mu_D)/\sigma_D$
  Compute r_t^{RND}
  $R_t \leftarrow \lambda_D \widetilde{r}_t^{\mathrm{TD}} + r_t^{\mathrm{env}} + \lambda_{\mathrm{expl}} r_t^{\mathrm{RND}}$
  Store the transition with reward R_t in $\mathcal{B}$
  Update DrQ-v2 from replay samples and update the RND predictor
end while