

Diffusion Reward: Learning Rewards via Conditional Video Diffusion

Tao Huang^{*,1,2} Guangqi Jiang^{*,1,3} Yanjie Ze¹ Huazhe Xu^{4,1,5}

¹Shanghai Qi Zhi Institute ²The Chinese University of Hong Kong

³Sichuan University ⁴Tsinghua University, IIIS ⁵Shanghai AI Lab

diffusion-reward.github.io

Abstract

Learning rewards from expert videos offers an affordable and effective solution to specify the intended behaviors for reinforcement learning tasks. In this work, we propose Diffusion Reward, a novel framework that learns rewards from expert videos via conditional video diffusion models for solving complex visual RL problems. Our key insight is that lower generative diversity is observed when conditioned on expert trajectories. Diffusion Reward is accordingly formalized by the negative of conditional entropy that encourages productive exploration of expert-like behaviors. We show the efficacy of our method over 10 robotic manipulation tasks from MetaWorld and Adroit with visual input and sparse reward. Moreover, Diffusion Reward could even solve unseen tasks successfully and effectively, largely surpassing baseline methods. Project page and code: diffusion-reward.github.io.

1. Introduction

Reward specification poses a fundamental challenge in reinforcement learning (RL), influencing the effectiveness and alignment of an agent's learned behavior with the intended objectives. Manually designing dense reward functions is burdensome and sometimes impossible, particularly in real-world tasks such as robotic manipulation [31], where obtaining privileged state information is difficult. As a substitute, using sparse rewards is often favorable in these scenarios because of its low demand for manual effort [25]. Nonetheless, the sample efficiency of RL drops significantly due to insufficient supervision from sparse rewards.

Learning reward functions from expert videos offers a promising solution because of the low effort of video collection and dense task-execution information contained in the videos [5, 42]. Given unlabeled videos, generative models have been naturally investigated by researchers to extract informative rewards unsupervisedly for RL training [27, 35].

*Equal contribution to this work.

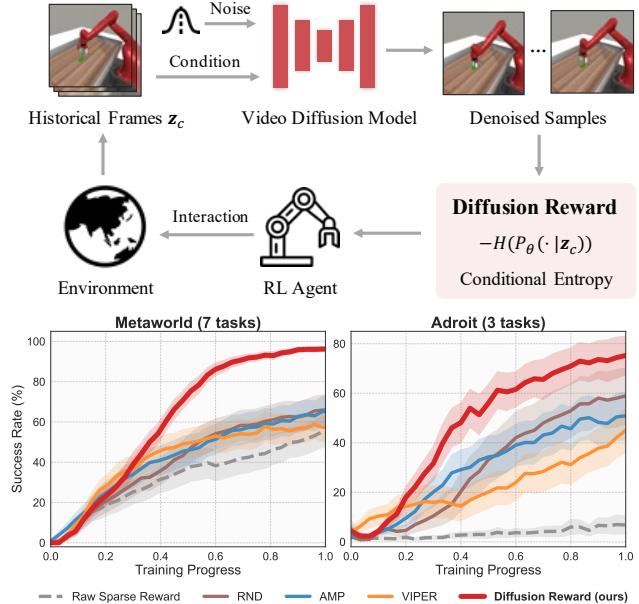


Figure 1. **Overview.** (top) We present a reward learning framework in RL using video diffusion models. We perform diffusion processes conditioned on historical frames to estimate conditional entropy as rewards to encourage RL exploration of expert-like behaviors. (bottom) The mean success rate of 10 visual robotic manipulation tasks demonstrates the effectiveness of our Diffusion Reward over 5 runs. Shaded areas are standard errors.

One classical approach builds on generative adversarial learning to learn a discriminative reward that discerns between agent and expert observations. While straightforward, these methods underutilize the temporal information, whose importance has been shown in solving RL [44], and performance is brittle to the adversarial training. In light of these issues, recent work leverages the VideoGPT [38] to encode temporal information, and directly use the predicted log-likelihood as rewards [9]. However, it struggles with modeling complex expert video distributions, particularly those with intricate dynamics. As shown in Figure 2, there exists a noticeable decline in the learned return is observed for out-of-distribution expert videos, despite sharing

optimal behavioral patterns with in-distribution ones.

Video diffusion models [17] — which have exhibited remarkable performance in the domain of computer vision — have shown great power in capturing the complex distribution of videos, such as text-to-video generation [11, 16] and video editing [4, 24]. Recent works have also examined their capability of modeling expert videos as generalizable planners in robotic manipulation tasks [8, 21]. Despite these advancements, extracting informative rewards from video diffusion models remains an understudied area, while shining great potential for guiding RL agents to acquire expertise from videos and generalizing to unseen tasks.

In this work, we propose **Diffusion Reward**, a reward learning framework that leverages conditional video diffusion models to capture the expert video distribution and extract dense rewards for visual RL. Our key insight is that higher generative diversity is observed when conditioned on expert-unlike videos, while lower given expert videos. This rationale naturally instructs RL exploration on expert-like behaviors by seeking lower diversity. We therefore estimate entropy conditioned on historical frames, which formalizes our insight, and augment it with a novelty-seeking reward [3] and spare environment reward to form dense rewards for efficient RL. In addition, to accelerate the reward inference, we perform latent diffusion processes by utilizing vector-quantized codes [12] for compressing high-dimensional observations.

We empirically validate the efficacy of our framework through experiments on 10 visual robotic manipulation tasks, including 7 gripper manipulation tasks from MetaWorld [40] and 3 dexterous manipulation tasks from Adroit [28], exhibiting 38% and 35% performance improvements over the best-performing baselines given the same training steps, respectively. Furthermore, we surprisingly find that our learned reward can achieve fair zero-shot generalization performance on unseen tasks. Figure 1 overviews this work. The primary contributions of this work can be summarized as follows:

- We present Diffusion Reward, a novel reward learning framework that leverages the generative modeling capabilities of video diffusion models to provide dense reward signals for RL agents.
- We show that our framework significantly outperforms baselines on 10 visual robotic manipulation tasks.
- We find that our pre-trained reward model could produce reasonable rewards and thus instruct RL on unseen tasks.

2. Related Work

Learning rewards from videos. Extracting supervision signals from expert videos provides an affordable but effective solution for reward specification in RL [5, 29, 30]. A number of works have attempted to learn dense rewards indicating task progress by measuring the distance

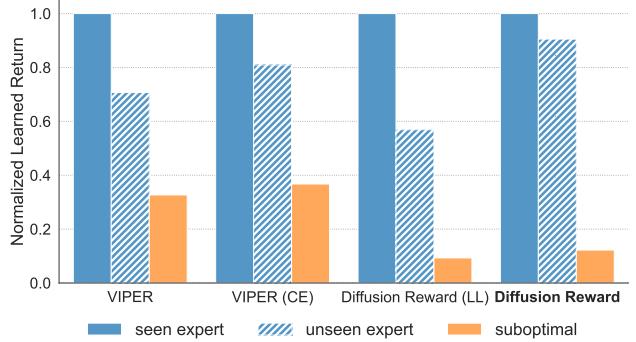


Figure 2. **Rewards from different video models.** Results are averaged over 7 MetaWorld tasks with 10 random seeds for each task. *Suboptimal* represents videos with 25% of actions taken by random policy. *VIPER(CE)* and *Diffusion Reward(LL)* replace their original rewards with conditional entropy (CE) and Log-likelihood (LL), respectively. We observe that LL-based methods assign relatively low rewards to unseen expert videos while CE-based methods are able to assign near-optimal rewards to unseen expert videos. Moreover, such a boost is enhanced by the strong modeling ability of diffusion models.

between current observation and goal image in the latent space [23, 42]. While promising, goal images are often hard to obtain out of simulation, limiting their applicability to open-world tasks [22]. On the contrary, generative models have been widely investigated to extract rewards unsupervisedly without future information. One representative approach builds on generative adversarial learning [27, 35] to discern expert-like and expert-like behaviors. Such an idea is further improved in [9] by predicting the log-likelihood as rewards, where the video prediction model is used to encode more temporal information. Nevertheless, it still struggles with modeling expert video distributions in complex tasks featuring intricate dynamics, thus being prone to produce unproductive rewards. In contrast, our methods leverage the powerful modeling abilities of diffusion models and estimate the negative of conditional as more discriminative rewards to expedite RL exploration. Moreover, our pre-trained reward could be generalized to unseen tasks better.

Diffusion models for RL. Diffusion models have been widely investigated for RL to, for instance, improve the policy expressiveness [7, 14, 37], and augment experience [6, 41]. Apart from these, some works directly learn the diffusion models from offline data unconditional planners [19], or conditional planners specified by task returns [1]. The idea of conditional diffusion is further investigated in [8, 21] with text as task specification, where video diffusion models serve as planners associated with inverse dynamics. Unlike these methods, we intend to learn informative rewards via video diffusion models conditioned on historical frames to accelerate online RL. Such historical conditioning has been used in [18] to inform trajectory generation, which differs from our focus on reward learning as well. Our work is



Figure 3. **Video prediction results.** Our video diffusion model could capture the distribution of expert videos from complex tasks. The outcomes are conditioned on two history frames and predictions. Ground truth has **blue** borders and prediction has **orange** borders.

also close to Nuti et al. [26], which has also attempted to extract reward from two diffusion models that fit different behaviors in 2D tasks. Differently, we learn diffusion-based rewards simply from expert behaviors and achieve favorable performance on complex vision-based manipulation tasks.

3. Preliminaries

Problem formulation. We consider an RL agent that interacts with the environment modeled as a finite-horizon Markov Decision Process (MDP), which is defined by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where \mathcal{S} is the state space, \mathcal{A} is the action space, \mathcal{T} is the environment transition function, \mathcal{R} is the reward function, and γ is the discount factor. The goal of the RL agent is to learn an optimal policy π that maximizes the expected return $\mathbb{E}[\sum_{k=0}^{K-1} \gamma^k r_k]$.

In this work, we focus more specifically on high-dimensional state space with a binary sparse (i.e., 0/1 reward as a success indicator of the task). Particularly, we consider RGB images $x \in \mathbb{R}^{H \times W \times 3}$ as the state observed by the agent. This setting is motivated by the real-world application of RL such as robotics, where vision-based sensory data is more available and specifying sophisticated reward often requires tedious hand-engineering, sometimes even intractable. However, this poses great difficulty to an RL agent, as a large amount of interaction with environments is often required, known as sample efficiency.

Expert videos. To improve the sample efficiency of RL, we assume a set of unlabeled videos generated by the expert policies are accessible by the agent. Notably, these videos are action-free and gathered from multiple tasks without task identification. We denote it as $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^N\}$, where \mathcal{D}^i is the demonstrated videos from task i . Each set of demonstrated videos \mathcal{D}^i contains multiple expert trajectories $\tau = \{x_0, x_1, \dots, x_{K-1}\} \in \mathcal{S}^K$. Our goal now turns to learning effective reward functions of such videos to accelerate the online exploration of RL agents.

Data	Model	SSIM (\uparrow)	PSNR (\uparrow)	LPIPS (\downarrow)
Real	VideoGPT	0.768 (± 0.141)	23.35 (± 4.13)	0.1017 (± 0.0670)
	VQ-Diffusion	0.856 (± 0.114)	27.46 (± 5.24)	0.0596 (± 0.0450)
Sim.	VideoGPT	0.967 (± 0.021)	32.28 (± 3.15)	0.0067 (± 0.0032)
	VQ-Diffusion	0.976 (± 0.009)	33.55 (± 2.31)	0.0053 (± 0.0029)

Table 1. **Quantitative comparison of video models.** Results are evaluated on real robot videos and simulation videos from 7 Meta-World tasks, demonstrating that video diffusion model generates videos of higher quality than VideoGPT.

4. Method

We introduce Diffusion Reward, a novel framework that learns rewards from expert videos via conditional video diffusion models for solving downstream visual RL problems, as illustrated in Figure 1. At a high level, our method leverages entropy information from video diffusion models pre-trained on expert videos to encourage RL agents to explore expert-like trajectories more. In Section 4.1, we first model the expert videos with the conditional video diffusion model in the latent space. In Section 4.2, we formalize our key insight of Diffusion Reward by estimating the history-conditioned entropy and using its negative as reward signals for RL training. We summarize our framework in Algorithm 1 and present the overview and key implementation details of our framework below.

4.1. Expert Video Modeling via Diffusion Model

Diffusion models [32] are probabilistic models that aim to model the data distribution by gradually denoising a normal distribution through a reverse diffusion process [15]. These models showcase their power in capturing highly complex distributions and generating samples that exhibit intricate dynamics, motion, and behaviors in RL literature [1, 19]. Unlike prior works that model expert videos as planners, we aim to learn reward functions from the diffusion model

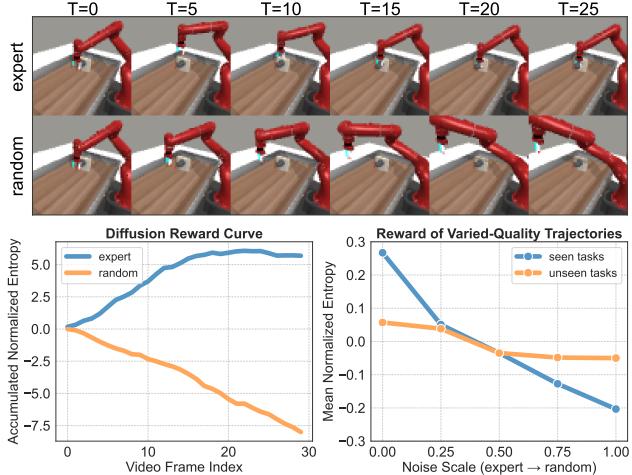


Figure 4. Conditional entropy as rewards. (left) Aggregated learned rewards over 7 tasks from MetaWorld, which assigns expert policy high rewards while random policy low rewards. (right) Normalized conditional entropy on both seen and unseen tasks from MetaWorld, demonstrating that our rewards can distinguish trajectories of high quality from those of decreasing qualities and indicating its potential generalization ability. (top) One pair of the evaluated expert and random trajectories is exemplified above.

trained on expert videos for RL. This motivates our video models to achieve fast inference speed and encode temporal information.

Latent diffusion process. Specifically, we first train an encoder unsupervisedly from expert videos to compress the high-dimensional observations. Here we use the VQ-GAN method [10] to represent the image \mathbf{x} with a vector-quantized latent code $\mathbf{z} = Q(E(\mathbf{x}))$, where E is the encoder and Q is the element-wise quantizer. The whole video is then represented by a sequence of latent variables $\tau = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{K-1}\}$, where we overwrite the definition of τ without ambiguity. Subsequently, the forward process applies noise ϵ in the latent space at each time step $t \in 0, \dots, T$ to the data distribution \mathbf{z}_k , resulting in a noisy sample \mathbf{z}_k^t , where $\mathbf{z}_k^t = \sqrt{\bar{\alpha}_t} \mathbf{z}_k^0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$, and $\bar{\alpha}$ is the accumulation of the noise schedule over past timesteps. To fit data distribution, we learn a parameterized variant of noise predictor $\epsilon_\theta(\mathbf{z}_k^t)$ that aims to predict the noise ϵ during the forward process. Then the parameterized reverse process $p_\theta(\mathbf{z}_k^{t-1} | \mathbf{z}_k^t)$ can be approximated and performed by iterative denoising a initial distribution.

Historical frames as condition. To utilize the temporal information from expert videos with the power of video diffusion, we further condition the reverse process with history frames, i.e., $p_\theta(\mathbf{z}_k^{t-1} | \mathbf{z}_k^t, \mathbf{z}_c)$, where \mathbf{z}_c is the concatenation of all historical frames $[\mathbf{z}_0, \dots, \mathbf{z}_{k-1}]$. This can also be viewed as matching the distribution of expert and agent trajectories [9]. In practice, we use a subsequence of historical frames as a condition to ensure higher computation efficiency while maintaining temporal information. Subse-

quently, one can perform the reverse process from a randomly sampled noise to generate the latent code of future frames and decode the code for video prediction, as shown in Figure 3. In this work, we use VQ-Diffusion [12] as our choice of video diffusion model due to its good performance and compatibility with vector-quantized latent code, but our framework can in principle adopt any off-the-shelf video diffusion models. We first tokenize each latent code \mathbf{z} indexed by its indices that specify the respective entry in the learned codebook of VQ-GAN, and take as a condition embedding the concatenated tokens. The embedding is then fed to the decoder that contains 16 transformer blocks and softmax layers with cross attention.

4.2. Conditional Entropy as Rewards

While previous studies have explored the use of log-likelihood as rewards with video prediction models, exemplified by works such as VIPER [9], this approach encounters two primary challenges. Firstly, it struggles with accurately modeling the distribution of complex expert videos featuring intricate dynamics, as shown in Table 1. Secondly, the moderate video modeling ability leads to undesired rewards. This issue is evident in Figure 2, where a significant drop in learned rewards between in-distribution expert videos and out-of-distribution ones, though both sets of videos demonstrate optimal behaviors.

Key insight behind Diffusion Reward. We address these challenges by harnessing the great generative capability of video diffusion models. Our observations indicate increased generation diversity with unseen historical observations (rand.) and reduced diversity with seen ones (expert), as shown in Table 2. This gives rise to the key insight of our proposed Diffusion Reward: diffusion conditioned on expert-like trajectories exhibits lower diversity where the agent ought to be rewarded more, and the opposite holds on unexpert-unlike ones. Such diversity discrimination not only incentivizes RL agents to chase expert-like behaviors but also enhances exploration through the stochasticity induced by the diffusion process.

ϵ -greedy traj.	Distances as Diversity Metrics		
	SSIM (\uparrow)	PSNR (\uparrow)	LPIPS (\downarrow)
0% (expert)	0.941 (± 0.054)	29.91 (± 5.52)	0.0171 (± 0.0166)
25%	0.901 (± 0.063)	26.01 (± 4.62)	0.0371 (± 0.0304)
50%	0.894 (± 0.065)	25.65 (± 4.29)	0.0421 (± 0.0297)
75%	0.844 (± 0.066)	22.60 (± 2.53)	0.0645 (± 0.0370)
100% (rand.)	0.829 (± 0.730)	22.25 (± 3.43)	0.0597 (± 0.0364)

Table 2. Quantitative results of generative diversities. Referring to [45], generative diversity is proportional to distances among a batch of generated videos. Expert trajectories, which are seen during training, show the lowest diversity while the unseen random ones exhibit the highest diversity. Results are over 7 tasks from MetaWorld and trajectories are generated by ϵ -greedy policies.

Algorithm 1 Diffusion Reward

```

// Pretrain reward model from expert videos
1: Collect expert videos  $\mathcal{D}$  from  $K$  tasks
2: Train diffusion model  $p_\theta$  on expert videos  $\mathcal{D}$ 
// Downstream RL with learned rewards
3: while not converged do
4:   Act  $a_k \sim \pi(\cdot | \mathbf{x}_k)$ 
5:   Generate  $M$  denoised samples  $\tilde{\mathbf{z}}_k^{0:T} \sim p_\theta(\mathbf{z}_k^{0:T} | \mathbf{z}_c)$ 
6:   Estimate entropy  $H(p_\theta(\cdot | \mathbf{z}_c))$  with  $\tilde{\mathbf{z}}_k^{0:T}$  ▷ Eq. (3)
7:   Compute Diffusion Reward  $r_k \leftarrow r_k^{\text{diff}}$  ▷ Eq. (5)
8:   Step environment  $\mathbf{x}_{k+1} \sim \mathcal{T}(\mathbf{x}_k, a_k)$ 
9:   Store transition  $(\mathbf{x}_k, a_k, r_k, \mathbf{x}_{k+1})$ 
10:  Update policy  $\pi$  and  $r^{\text{rnd}}$  with RL algorithm
11: end while

```

Estimation of conditional entropy. To formalize this idea, we seek to estimate the negative conditional entropy given historical frames \mathbf{z}_c , which in principle captures the conditional generation diversity:

$$-H(p_\theta(\cdot | \mathbf{z}_c)) = \mathbb{E}_{p_\theta(\cdot | \mathbf{z}_c)} [\log p_\theta(\mathbf{z}_k | \mathbf{z}_c)]. \quad (1)$$

One primary challenge is the computation of the entropy in Eq. (1), which is intractable as we have no explicit form of the conditional distribution [34]. Therefore, we instead attempt to estimate the variational bound of such entropy. Specifically, we first present the variational bound of conditional log-likelihood as follows:

$$\log p_\theta(\mathbf{z}_k^0 | \mathbf{z}_c) \geq \mathbb{E}_{q(\mathbf{z}_k^{0:T} | \mathbf{z}_c)} \left[\log \frac{p_\theta(\mathbf{z}_k^{0:T})}{q(\mathbf{z}_k^{1:T} | \mathbf{z}_k^0, \mathbf{z}_c)} \right], \quad (2)$$

where \mathbf{z}_k^0 is the denoised prediction of current observation \mathbf{z}_k . This bound could be estimated via noise predictor ϵ_θ [15, 33], or with the closed-form distribution [20, 32] (e.g., discrete multivariate distribution). We use the latter one as our choice of estimation because of its better compatibility with VQ-Diffusion.

Next, to estimate the whole entropy, we denoise from randomly sampled noise and generate the latent variables $\tilde{\mathbf{z}}_k^{0:T} \sim p_\theta(\mathbf{z}_k^{0:T} | \mathbf{z}_c)$, repeating with M times. Subsequently, we use the generated samples from randomized noise $\tilde{\mathbf{z}}_k^T$ (e.g., random tokens) to estimate the whole conditional entropy term as follows:

$$r^{\text{ce}}(\mathbf{x}_{k-1}) = \frac{1}{M} \sum_{j=1}^M \log \frac{p_\theta(\tilde{\mathbf{z}}_k^{0:T})}{q(\tilde{\mathbf{z}}_k^{1:T} | \tilde{\mathbf{z}}_k^0, \mathbf{z}_c)}, \quad (3)$$

We visualize the aggregated reward in Figure 4 and present curves of each task in the Appendix. The results show that conditional entropy can successfully capture the varied generative diversity on different videos, echoed with the aforementioned insight of Diffusion Reward. Notably, here we

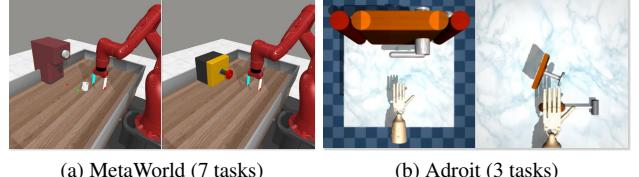


Figure 5. Task visualization. We evaluate our method on 10 challenging visual RL tasks from MetaWorld and Adroit, with 64×64 -dimensional RGB images and sparse rewards. Tasks are chosen to cover a wide range of manipulation skills.

use the standardized entropy reward \bar{r}^{ce} to mitigate the burden of hyperparameter tuning, as we observe that the scale of conditional entropy varies significantly across different tasks and domains, partially attributed to the varied objects and environment dynamics. Concretely, the conditional entropy is standardized by the empirical mean and standard deviation of the expert videos:

$$\bar{r}^{\text{ce}} = (r^{\text{ce}} - \text{mean}(\mathcal{D}, r^{\text{ce}})) / \text{std}(\mathcal{D}, r^{\text{ce}}). \quad (4)$$

Exploration reward. As the reward \bar{r}^{ce} incentivizes the agent to mimic the behavioral patterns of the expert, the exploration may still be prohibitively challenging in complex tasks with high-dimensional input. To alleviate this issue, we incorporate RND [3] as the exploration reward, termed as r^{rnd} .

Diffusion Reward. To this end, we combine our proposed diffusion-based entropy reward with the exploration reward and the raw sparse reward r^{spar} from the environment,

$$r^{\text{diff}} = (1 - \alpha) \cdot \bar{r}^{\text{ce}} + \alpha \cdot r^{\text{rnd}} + r^{\text{spar}}, \quad (5)$$

where α is the reward coefficient that steers the weight of two rewards. The integration of r^{spar} is crucial, as the complete absence of environmental supervision may hinder progress in tackling complex tasks.

4.3. Training Details

We first provide the expert videos from various tasks as the whole dataset for pertaining reward models, which can be generated by scripted policies or other means. Then, we first use VQ-GAN [10] to train the encoder, associated with 8×8 size of codebook across all domains, with an additional perceptual loss [43] calculated by a discriminator to increase perceptual quality. We subsequently use VQ-Diffusion [12] for training the conditional video diffusion model, where the number of condition frames is set as 2 for all tasks. For reward inference, the reward coefficient is set as 0.95 for all tasks except 0 for the Pen task. Moreover, during the diffusion process, we also use a DDIM-like sampling strategy [33] to accelerate the diffusion process with denoise steps set as 10 and repeated with 1 times for all tasks, which shows fair performance and retains high inference speed.

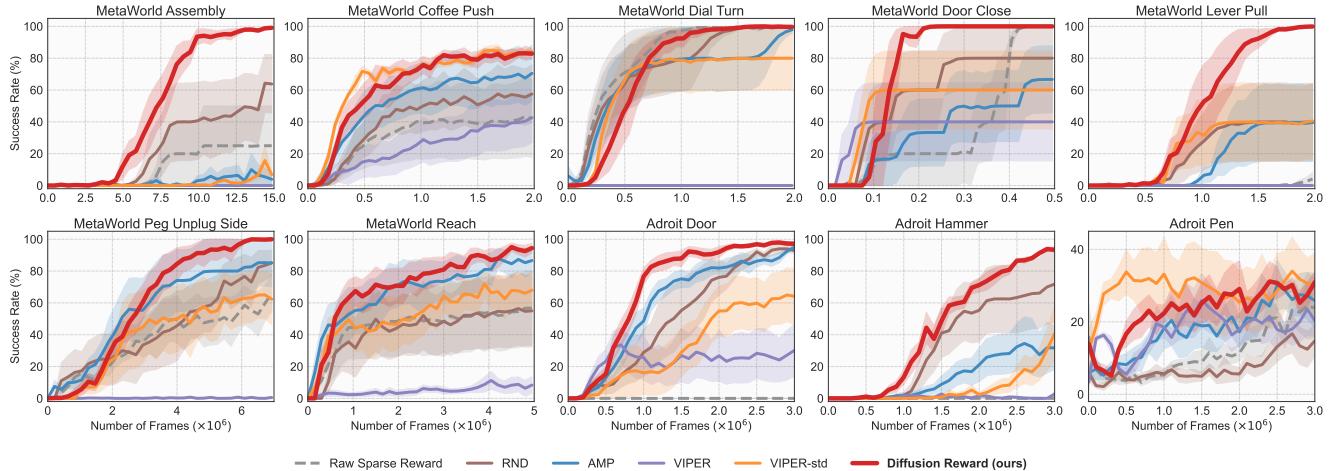


Figure 6. **Main results.** Success rate for our method and baselines on 7 gripper manipulation tasks from MetaWorld and 3 dexterous manipulation tasks from Adroit with image observations. Our method achieves prominent performance on all tasks, and significantly outperforms baselines on complex door and hammer tasks. Results are means of 5 runs with standard errors (shaded area).

5. Experiments

In this section, we first introduce the overall experimental setups in Section 5.1. We then present the significant performance improvement of our method against competitive baselines in Section 5.2, investigate the generalization capability and effectiveness on real robot videos of our learned reward in Section 5.3 and Section 5.4, respectively. We also conduct comprehensive ablations to study the effect of each component and implementation details in Section 5.5.

5.1. Experimental Setup

Simulation environments. We intend to demonstrate the effectiveness of Diffusion Reward on 10 complex visual robotic manipulation tasks, including 7 gripper manipulation tasks from MetaWorld [40] and 3 dexterous hand manipulation tasks from Adroit [28], as visualized in Figure ???. We choose these two simulation environments because of their task diversity and complexity. Each task is associated with 64×64 -dimensional RGB images, ± 4 pixel shift augmentation [39], and 0/1 environmental sparse reward. For pertaining reward models, we collect 20 expert videos for each Metaword task via the scripted policy provided by the official repository, and 50 for Adroit via the policies trained with performant RL method [36]. For downstream RL training, the interaction budget of the Adroit task is set as 3 million to ensure convergence of using our rewards, and is respectively set for the MetaWorld task due to the varied task complexity.

Baselines. We compare our method against the followings:

- **Raw Sparse Reward** that uses the environmental sparse reward. This comparison tests the benefit of adding our pre-trained reward.

- **Random Network Distillation** (RND, [3]) that encourages exploration with a novelty-seeking reward. This comparison tests the benefit of rewarding the agent with expert-like behaviors.

- **Adversarial Motion Priors** (AMP, [27]) that learns a discriminator to discern agent behaviors and expert behaviors based on current observations. This comparison tests the benefit of encoding temporal information in learned reward and utilization of novelty-seeking reward.

- **Video Prediction Rewards** (VIPER, [9]) and its standardized variant (VIPER-std.) that use VideoGPT [38] as video prediction model and predicted log-likelihood of agent observation as reward. This comparison tests the benefit of utilizing the generative capability of video diffusion models and the conditional entropy as a more explorative reward.

For a fair comparison, all methods use DrQv2 [39] as the RL backbone and maintain all settings except reward-pretraining (if exist) identical.

5.2. Main Results

Comparison with non-pretraining methods. We present the learning curves of success rates for each method over two simulation domains in Figure 1 and 6. The results show that solely using sparse task rewards enables progress in relatively straightforward tasks such as Reach and Dial Turn. However, it encounters significant challenges in more complex ones, exemplified by the Door and Hammer within the dexterous hand manipulation domain. The incorporation of pure novelty-seeking rewards (RND) unsurprisingly enhances the RL agent’s exploration, particularly evident in addressing moderately complex tasks like Coffee Push and Assembly. Nevertheless, the performance on dexter-

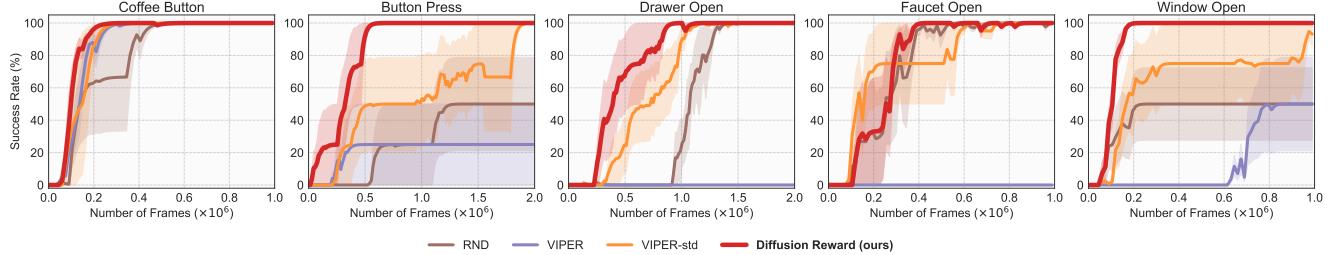


Figure 7. **Success rate curves on 5 unseen MetaWorld tasks.** Diffusion Reward could generalize to unseen tasks directly and produce reasonable rewards, largely surpassing other baselines. Results are means of 4 runs with standard errors (shaded area).

ous hand manipulation tasks is still unsatisfactory due to the lack of expert-instructed exploration at prohibitively large configuration space. Conversely, AMP explicitly employs expert-guided rewards to incentivize the exploration of expert-like behaviors. While it generally outperforms RND in simpler tasks, its efficacy diminishes in more complex ones such as Lever Pull and Hammer.

Comparison with reward pretraining method. The above observations suggest that the combination of expert-instructed rewards and novelty-seeking rewards is likely to perform favorably. Despite the incorporation of such a combination in the VIPER reward, its empirical performance unexpectedly falls short of both RND and AMP. We posit this is attributed to the significantly varied scales of expert-instructed rewards (i.e., log-likelihood), which may eliminate the effect of novelty-seeking rewards. VIPER-std alleviates this issue and shows better performance than the non-standardized version, especially on Coffee Push and Pen, while still underperforming our method. These results are in line with two limitations outlined in Section 4.2 and further verified in Figure 2, indicating VIPER’s struggle in capturing complex video distributions within intricate tasks. In sharp contrast to VIPER, our proposed method not only utilizes the modeling capabilities of diffusion models, but also uses conditional entropy as a reward function to accelerate exploration. The results showcase remarkable performance improvements of 38% and 35% over the best-performing baselines with the same training steps across MetaWorld and Adroit, respectively.

5.3. Zero-Shot Reward Generalization

Reward visualization. Video diffusion models have exhibited powerful abilities for modeling videos and, notably, for generating samples beyond their training data, as exemplified in text-to-image video generation [11, 16]. Such an advance motivates our exploration into the potential of Diffusion Reward to generalize to previously unseen tasks. To investigate this, we start by visualizing the learned returns of trajectories with varying qualities derived from 15 unseen gripper manipulation tasks in MetaWorld (see Figure 4). Interestingly, the distinctions between trajectories

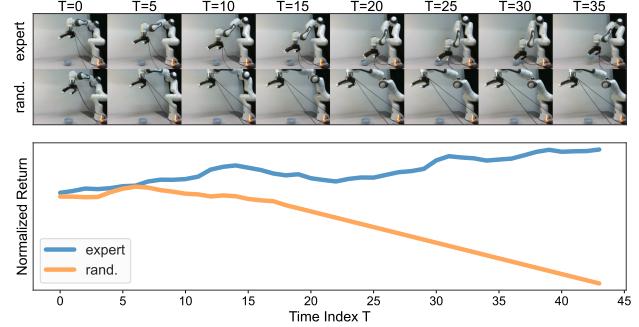


Figure 8. **Reward curve of real robot videos.** Our method assigns higher rewards to expert videos than to random videos.

of different qualities are less noticeable in comparison to those observed in tasks seen during training. Nevertheless, our pre-trained reward model still exhibits a consistent trend where expert-like behaviors receive relatively higher learned returns, owing to the generalization prowess of the video diffusion model.

RL performance. Subsequently, we directly apply our pre-trained rewards to 5 tasks involving diverse objects without additional tuning. The outcomes, as shown in Figure 7, affirm that our reward effectively guides RL exploration and largely outperforms other baseline methods across all tasks. Notably, our approach proves helpful in constraining the exploration space of RND due to its retained ability to discriminate between expert-like and expert-unlike behaviors. In contrast, VIPER struggles to generalize effectively on most tasks, partially attributed to the combined limitations of the adopted video models and log-likelihood rewards. These results not only verify the efficacy of our method, but also point towards the potential of employing larger diffusion models and integrating other modalities (e.g., text embedding for task specification) to enhance the generalization capabilities of our approach further.

5.4. Real Robot Evaluation

We consider the real robot task that aims to pick up a bowl on the table. To train our reward model, we collect 20 real robot videos (10 expert and 10 random) with an Allegro hand, a Franka arm, and a RealSense. Visualization results

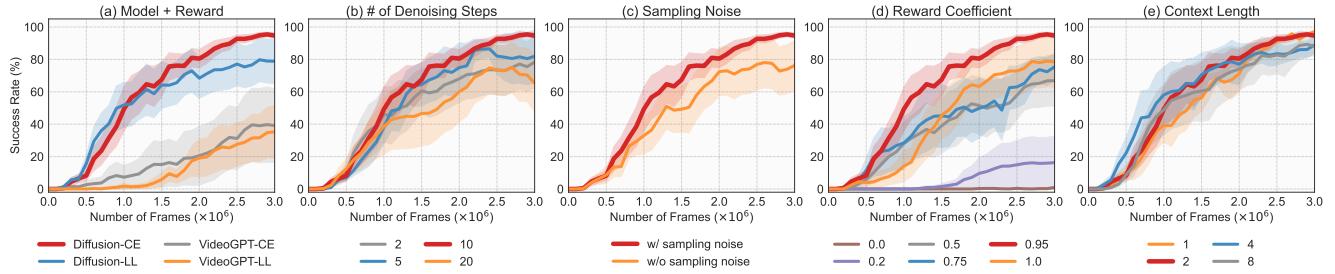


Figure 9. **Ablations.** Success rate curves for ablated versions of Diffusion Reward, aggregated over Door and Hammer from Adroit. (a) We test the combinations of generative models and rewards to show the benefit of estimating conditional entropy with the diffusion model. (b) We ablate the choice of the number of denoising steps. (c) We demonstrate that the inherent randomness of Diffusion Reward from the reverse process helps RL exploration. (d) We ablate the choice of reward coefficient. (e) We test the effect of the number of conditional frames. Results are means of 3 seeds with standard error (shaded area). **Red** is our default.

in Figure 8 show that Diffusion Reward can appropriately assign expert videos relatively higher rewards and, in contrast, random videos lower rewards, indicating the potential of our method for real-world robot manipulation tasks. We present more curves of multiple trials in the Appendix.

5.5. Ablation Studies

As shown in Figure 9, we ablate key design choices in our proposed framework in the previous experiments, aiming to reveal more insights into the quantitative performance of our method. We present more detailed analyses below.

Conditional entropy with diffusion model. As our method has demonstrated better performance than VIPER, we take a further step to see the joint effect of the reward types and the video prediction models. Specifically, we systematically evaluate all possible combinations of conditional entropy and log-likelihood as reward signals, each paired with either diffusion-based or transformer-based video prediction models. Note that the same vector-quantized encoder is used for all models, ensuring the variations are solely attributable to the chosen video prediction model and reward. The outcomes consistently align with the observations delineated in Figure 2, indicating two-fold conclusions: (1) video diffusion models are more adept at capturing the complex distribution of expert videos in complex tasks, thus resulting in more informative rewards; (2) employing conditional entropy as rewards prove more productive in RL exploration than using log-likelihood, partially owing to the more generalizable reward inference on trajectories unseen in reward pretraining stage.

Denoising steps. The number of timesteps involved in the reverse process governs the quality and diversity of generated frames [33]. This study seeks to investigate its impact on the derived reward and subsequent RL performance by gradually increasing the number of denoising steps from 2 to 20. The findings indicate that an intermediate choice, approximately around 10 steps, achieves the best performance. This suggests that an intermediate choice balances

generative quality and diversity well, thereby producing effective rewards for RL exploration. Furthermore, we observe that the speed of reward inference declines with an increase in denoising steps. For instance, the Frames Per Second (FPS) during RL training drops from approximately 100 with 2 steps to 60 with 20 steps in Adroit tasks with NVIDIA A40, suggesting the importance of adopting advanced techniques to expedite the diffusion process.

Sampling noise in diffusion process. We hypothesize that the introduction of randomness in the diffusion process holds the potential to enhance RL exploration, akin to the stochastic characteristics inherent in maximum entropy RL [46]. To substantiate this point, we design a variant of Diffusion Reward wherein the sampling noise is deliberately set as 0 during the reverse process, ensuring the reward becomes deterministic given the identical historical observations. The outcomes show a discernible degradation in performance when employing a deterministic reward. Notably, this decline in performance aligns with results observed when combining the diffusion model with log-likelihood rewards, wherein the rewards are also deterministic. Consequently, our findings demonstrate that the inherent randomness of Diffusion Reward from the reverse process indeed contributes to RL exploration.

Reward coefficient α . The reward coefficient α determines the relative importance of the conditional entropy reward against novelty-seeking. We investigate the effect of this parameter by gradually decreasing the value of α from 1 to 0 and repeatedly train the RL agent with the remaining settings identical. The results show that α around 0.95 achieves the best performance, while too-large ones (akin to RND only) and too-small ones (akin to no RND) exhibit significant performance drops. This suggests the domination of Diffusion Reward may still result in getting stuck to local optima, while our proposed reward effectively helps RL agent to narrow down the wide intended exploration space of novelty-seeking rewards.

Context length. The number of historical frames deter-

mines the extent of temporal information being encoded during video diffusion, thus influencing the generating process of video diffusion and induced reward inference. To investigate its effect on downstream RL, we test different choices of context length. The results suggest that opting for 1 or 2 historical frames proves sufficient to generate highly effective rewards, owing to the robust generative capabilities inherent in the diffusion model. Interestingly, a marginal decline in performance is observed when the context length is extended to 4 or 8 frames. This phenomenon may be attributed to potential overfitting to expert trajectories, resulting in inferred rewards that exhibit suboptimal generalization to previously unseen trajectories.

6. Conclusion

In this work, we propose Diffusion Reward, a novel framework that extracts dense rewards from a pre-trained conditional video diffusion model for reinforcement learning tasks. We first pre-train a video diffusion model on expert videos and find that the entropy of the predicted distribution well discriminates the expert-level trajectories and under-expert-level ones. Therefore, we use its standardized entropy, plus the exploration reward and the sparse environmental reward, as an informative reward signal. We evaluate Diffusion Reward over 10 visual robotic manipulation tasks from MetaWorld and Adroit and observe prominent performance improvements over 2 domains. We also demonstrate that our pre-trained reward could directly produce reasonable rewards in unseen tasks, largely surpassing baseline methods. This underscores the potential of large-scale pretrained diffusion models in reward generalization.

Future work will leverage larger diffusion models sourced from a wider dataset to solve diverse simulation and real-world tasks. The incorporation of additional modalities, such as language, will be explored to augment the generalization capabilities of Diffusion Reward. In addition, enhancing the diffusion-based reward itself, including strategies to balance entropy reward and exploration reward and the estimations of conditional entropy, holds promise for yielding better outcomes.

References

- [1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations (ICLR)*, 2023. [2](#), [3](#)
- [2] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam M. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. [11](#)
- [3] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019. [2](#), [5](#), [6](#)
- [4] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [2](#)
- [5] Annie S. Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos. In *Robotics: Science and Systems (RSS)*, 2021. [1](#), [2](#)
- [6] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *ArXiv*, abs/2302.06671, 2023. [2](#)
- [7] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023. [2](#)
- [8] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. [2](#)
- [9] 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 (NeurIPS)*, 2023. [1](#), [2](#), [4](#), [6](#), [11](#)
- [10] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [4](#), [5](#), [11](#)
- [11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [2](#), [7](#)
- [12] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [2](#), [4](#), [5](#), [11](#)
- [13] Nicklas Hansen, Zhecheng Yuan, Yanjie Ze, Tongzhou Mu, Aravind Rajeswaran, Hao Su, Huazhe Xu, and Xiaolong Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. In *International Conference on Machine Learning (ICML)*, 2023. [11](#)
- [14] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *ArXiv*, abs/2304.10573, 2023. [2](#)
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [3](#), [5](#)
- [16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen

- video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022. 2, 7
- [17] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022. 2
- [18] Jifeng Hu, Yanchao Sun, Sili Huang, SiYuan Guo, Hechang Chen, Li Shen, Lichao Sun, Yi Chang, and Dacheng Tao. Instructed diffuser with temporal condition guidance for offline reinforcement learning. *ArXiv*, abs/2306.04875, 2023. 2
- [19] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022. 2, 3
- [20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ArXiv*, abs/1312.6114, 2013. 5, 11
- [21] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to Act from Actionless Video through Dense Correspondences. *ArXiv*, abs/2310.08576, 2023. 2
- [22] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. In *Robotics: Science and Systems (RSS)*, 2020. 2
- [23] 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 (ICLR)*, 2023. 2
- [24] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *ArXiv*, abs/2302.01329, 2023. 2
- [25] Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1
- [26] Felipe Nuti, Tim Franzmeyer, and João F. Henriques. Extracting reward functions from diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [27] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph (ToG)*, 2021. 1, 2, 6
- [28] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018. 2, 6, 11
- [29] Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *ArXiv*, abs/1612.06699, 2016. 2
- [30] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 2
- [31] Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *ArXiv*, abs/1904.07854, 2019. 1
- [32] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 3, 5, 11
- [33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2022. 5, 8
- [34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020. 5
- [35] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *ArXiv*, abs/1807.06158, 2018. 1, 2
- [36] Che Wang, Xufang Luo, Keith W. Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 6
- [37] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023. 2
- [38] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *ArXiv*, abs/2104.10157, 2021. 1, 6
- [39] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022. 6, 11
- [40] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan C. Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Metaworld: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020. 2, 6, 11
- [41] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *ArXiv*, abs/2302.11550, 2023. 2
- [42] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2022. 1, 2
- [43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5
- [44] Ruijie Zheng, Xiayao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daum'e, and Furong Huang. Taco: Temporal latent action-driven contrastive loss for visual reinforcement learning. *ArXiv*, abs/2306.13229, 2023. 1

- [45] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 4
- [46] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2008. 8

A. Implementation Details

In this section, we provide further implementation details on Diffusion Reward and baselines. Note that all methods use the same RL backbone and maintain all settings except reward pretraining (if exist) identical.

A.1. Diffusion Reward Implemenatation

Codebase. Our codebase of VQ-GAN [10] is built upon the implementation in <https://github.com/dome272/VQGAN-pytorch>, which provides clean code structure and show fast inference speed. The codebase of VQ-Diffusion [12] is built upon the official implementation, which is publicly available on <https://github.com/microsoft/VQ-Diffusion>. For the downstream RL, we adopt the official implementation of DrQv2 [39] as RL backbone, which is publicly available on <https://github.com/facebookresearch/drqv2>, and the implementation of RND as exploration reward available on <https://github.com/jcwleo/random-network-distillation-pytorch>.

Network architectures. The major network architectures employed in Diffusion Reward follow the original implementation provided by the codebase above (refer to corresponding papers for more details), except for modifications performed in the conditional video diffusion part. Specifically, each historical frame, encoded by the encoder E and quantizer Q learned with VQ-GAN, is tokenized (8×8) by the condition network, concatenated with others ($2 \times 8 \times 8$), fed into embedding networks with a dimension of 1024. The resulting condition embedding with a dimension of 128×1024 is passed to subsequent conditional diffusion.

Hyperparameters. We list the important hyperparameters of VQ-GAN, VQ-Diffusion, DrQv2 with Diffusion Reward in Table 4, 5, and 6, respectively.

Entropy estimation details. As described in Section 4.2, the variational bound of conditional entropy in Eq. (1) can be estimate by Eq. (3). Such estimation is realized with the closed-form distribution [20, 32] (e.g., discrete multivariate distribution) in this work. Specifically, the variational bound of conditional log-likelihood in Eq. (1) can be simplified following [32], resulting in our estimation of entropy

reward r^{ce} as follows:

$$\begin{aligned} r^{\text{ce}}(\mathbf{x}_{k-1}) &= \frac{1}{M} \sum_{j=1}^M \left(\log p_\theta(\tilde{\mathbf{z}}_k^0 | \tilde{\mathbf{z}}_k^1, \mathbf{z}_c) \right. \\ &\quad + \sum_{t=1}^{T-1} D_{KL}(q(\tilde{\mathbf{z}}_k^{t-1} | \tilde{\mathbf{z}}_k^t, \tilde{\mathbf{z}}_k^0) \| p_\theta(\tilde{\mathbf{z}}_k^{t-1} | \tilde{\mathbf{z}}_k^t, \mathbf{z}_c)) \\ &\quad \left. + D_{KL}(q(\tilde{\mathbf{z}}_k^T | \tilde{\mathbf{z}}_k^0) \| p(\tilde{\mathbf{z}}_k^T)) \right), \end{aligned} \quad (6)$$

where D_{KL} denotes the Kullback–Leibler divergence, $p(\tilde{\mathbf{z}}_k^T)$ follows the prior distribution of random noise at timestep T , and $\tilde{\mathbf{z}}_k^{0:T} \sim p_\theta(\mathbf{z}_k^{0:T} | \mathbf{z}_c)$ represents the denoised samples via inverse diffusion process. We set $M = 1$ to ensure high reward inference speed while retaining its discrimination on expert-like and -unlike behaviors. We present more analyses in Section D.

A.2. Baselines Implementations

RND implementation. This baseline combines the sparse environmental reward with the RND exploration reward, which is equivalent to setting the reward coefficient α as 1 in Diffusion Reward. To this end, we implement this baseline by simply removing the entropy reward and keeping other settings identical.

VIPER implementation. We implement their adopted VideoGPT based on the official code provided in <https://github.com/wilson1yan/VideoGPT> with clean GPT implementation from <https://github.com/karpathy/minGPT>. The calculation of video prediction (log-likelihood) rewards follows the official JAX implementation provided in https://github.com/Alescontrela/viper_rl, which uses ‘teacher-forcing’ practice [2] (where ground truth context is provided for each step) for fast inference speed. The coefficient between video prediction reward and RND exploration reward is set as 0.5 following their paper [9].

AMP implementation. The implementation is based on the official code in <https://github.com/xbpeng/DeepMimic> and re-implementation in <https://github.com/med-air/DEX>. The encoder consists of three 32-channel convolutional layers interpolated with ReLU activation. The discriminator is implemented as a 3-layer MLP with hidden dimensions of 256 and Tanh activation.

B. Task Descriptions

We select 7 gripper manipulation tasks from MetaWorld [40] and 3 dexterous manipulation tasks from Adroit [28], as visualized in Figure 10. The tasks are widely used in visual RL [13] and are chosen to be diverse in objects and manipulating skills. All tasks render 64×64 -dimensional RGB as the agent’s observation and produce sparse environmental rewards. According to task complexity, we collect 20 expert videos for each MetaWorld task and 50 for each Adroit task. We describe each task below:

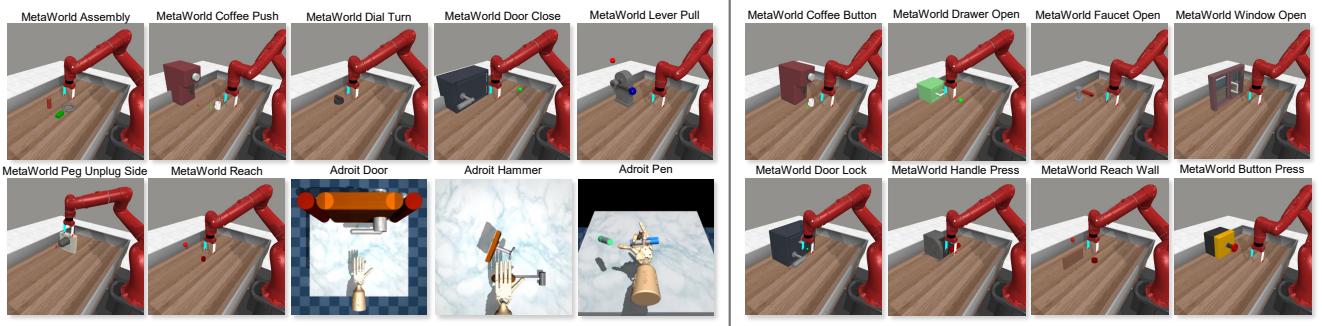


Figure 10. **Task descriptions.** (left) 10 seen training tasks from MetaWorld and Adroit. (right) 8 unseen tasks from MetaWorld.

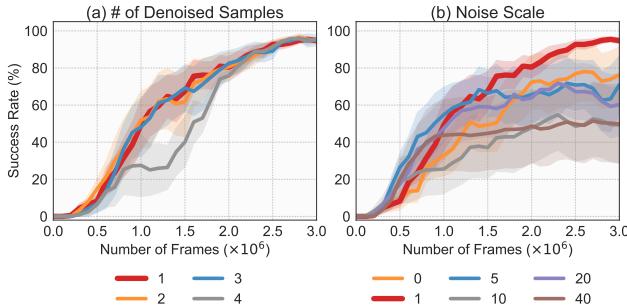


Figure 11. **Ablations on diffusion process.** The results in (a) suggest that overmuch denoised samples (i.e., 4) may hinder the exploration due to the low variance of estimated entropy. This is further verified in (b) where an appropriate choice of sampling noise scale results in more productive explorations. Results are means of 3 seeds with standard errors (shaded area). Red is our default.

- **Assembly** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to pick up a nut and place it onto a peg with the gripper.
- **Coffee Push** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to push a mug under the coffee machine to a target position with the gripper.
- **Dial Turn** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to rotate the dial with the gripper.
- **Door Close** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to close the door with the arm.
- **Lever Pull** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to pull a lever up with the arm.
- **Peg Unplug Side** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to unplug a peg sideways with the gripper.
- **Reach** (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to reach a target position with the end effector.
- **Door** (Adroit, $\mathcal{A} \in \mathbb{R}^{28}$): the task is to open the door to touch the door stopper.
- **Hammer** (Adroit, $\mathcal{A} \in \mathbb{R}^{26}$): the task is to pick up the hammer to hit the nail into the board.
- **Pen** (Adroit, $\mathcal{A} \in \mathbb{R}^{18}$): the task is to reorient the pen in-hand to a target orientation.

C. More Video Prediction Results

Qualitative results. We present the comparison between expert videos and prediction results in Figure 14. We find that the adopted video diffusion model is able to capture the complex distribution of expert videos from pretraining data and generalize well to unseen expert videos. Interestingly, we also observe that the colored target points in Door Close and Reach are sometimes mispredicted, which may explain the relatively slow exploration at the initial RL training stage, suggesting that a more powerful video diffusion model could be used to further improve the reward quality.

Quantitative analysis. We compare the video prediction quality between Diffusion Reward and VIPER in terms of three video metrics, SSIM, PSNR, and LPIPS. Results are shown in Table 1, verifying that Diffusion models hold a stronger generalization ability on unseen real robot and simulation trajectories than VideoGPT and thus produce more informative rewards. For real robot videos, we use 16×16 codes to represent images due to the scene complexity.

D. More Ablations

We conduct more ablations on our proposed Diffusion Reward in this section. Results are aggregated over Door and Hammer from Adroit with 3 random seeds.

Sparse environmental reward. The sparse environmental reward r_{spar} is integrated into Eq. (5), as the environmental supervision of completion is helpful for RL. We remove r_{spar} to study its effect. The results in Figure 12 show that the performances decrease dramatically without sparse environmental rewards, indicating that the signal of task completion is necessary for solving complex manipulation tasks. Interestingly, more favorable performance is observed in the Door task. We attribute this to the overlap supervision of RND reward and sparse reward, i.e., exploring novel states (door opening) is partially equivalent to completing the task.

Diffusion process. Recall that, in Eq. (3), we perform inverse process for M times, and use the generated M sam-

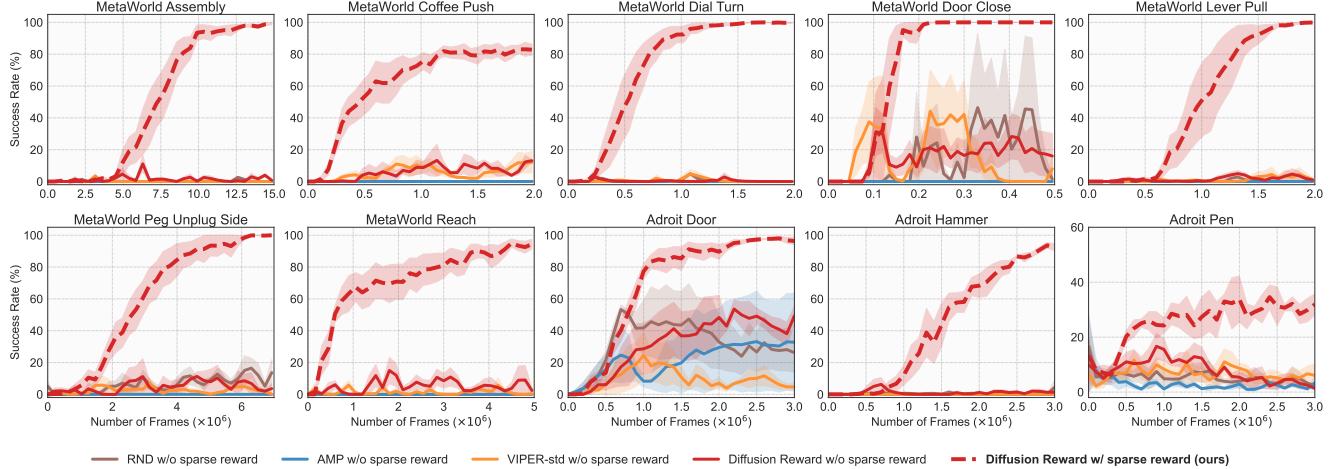


Figure 12. Effect of sparse environmental reward. We demonstrate that incorporating sparse environmental reward as a task completion signal is necessary for solving complex manipulation tasks. Results are means of 3 seeds with standard errors (shaded area).

ples to estimate the conditional entropy as rewards. The results in Figure 11(a) show that the effect of the number of denoised samples has a slight influence on RL performance when increasing from 1 to 3. However, the learning progress gets stuck in the middle state with 4 denoised samples, though the asymptotic performance is still satisfactory. We posit that this is due to the low variance of estimated entropy, which may lead to more weight on exploitation instead of exploration.

To verify our hypothesis, we make a further ablation on the scale of sampling noise (e.g., uniform distribution) during the diffusion process. Different from Figure 9(b), we gradually increase the sampling noise scale from 0 to 40. The results in Figure 11(b) indicate that an excessively low noise scale (e.g., 0) will bring low randomness of learned reward, resulting in more exploitative behaviors, and too high noise scale may produce more random explorations. In contrast, an intermediate choice of noise scale will bring an appropriate variance of estimated entropy, contributing to productive explorations.

Meanwhile, we present the time efficiency of different numbers of diffusion processes with NVIDIA A40 in Table 3, where the Frames Per Second (FPS) decreases from 87.7 to 45.9 when the number of denoise samples increases from 1 to 4. It suggests that using 1 denoised sample is sufficient to provide informative rewards and retain high inference speed.

	$M = 1$	$M = 2$	$M = 3$	$M = 4$
FPS	87.7	67.5	55.8	45.9

Table 3. Time efficiency of the number of samples M on Door.

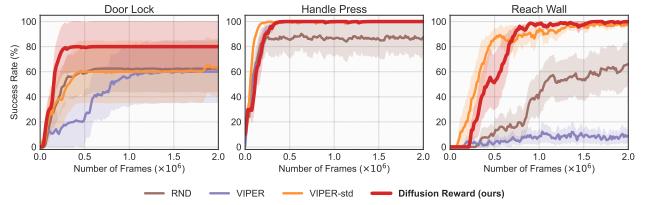


Figure 13. More results of reward generalization. Diffusion Reward exhibit better generalization ability than VIPER. Results are means of 5 seeds with standard errors (shaded area).

E. More Generalization Experiments

Apart from 5 unseen tasks in Figure 7, we randomly select 3 more unseens tasks from MetaWorld to verify the zero-shot generalization capability of Diffusion Reward. The result in Figure 13 demonstrates that our method significantly outperforms VIPER on most tasks. Notably, our method also outperforms RND in Reach Wall in terms of productive exploration at the initial training stage. In the future, we will investigate the possibility of incorporating other modalities (e.g., text embedding for task description) to enhance the generalization ability of Diffusion Reward.

F. Visualization of Reward and Trajectory in Simulation and Real Robot

Simulation Results. We first visualize the reward curve in 10 simulation tasks in Figure 15. Our proposed reward can greatly distinguish the expert-like and -unlike behaviors. Interestingly, we observe that two door-opening tasks show a return drop at the final execution stage. This may be attributed to the difficulty of modeling the dynamics of the door, suggesting that explicit modeling of environmental dynamics is worth investigating in the future.

Real Robot Results. We collect 20 real robot video trajec-

tories with an Allegro hand, a Franka arm, and a RealSense D435i camera. There is only one task, which is picking up a bowl on the table. 10 of the videos are success trajectories while the other 10 are random trajectories. We train our pipeline on the expert demonstrations and evaluate Diffusion Reward on both expert and random trajectories. Visualization results in Figure 16 show that Diffusion Reward can correctly assign expert demonstrations relatively higher reward and random trajectories lower reward.

Table 4. Hyperparameters for VQ-GAN.

Hyperparameter	Value
Input size	$64 \times 64 \times 3$
Latent code size	8×8
β (commitment loss coefficient)	0.25
Codebook size	1024
Codebook dimension	64
Base channels	128
Ch. mult.	[128, 128, 256, 256]
Num. residual blocks	2
Use attention	True
Disc. start steps	1000
Disc. loss weight	0.1
Reconstruction loss weight	1
Perceptual loss weight	0.1
Training epochs	200
Batch size	32
Learning rate	10^{-4}
Adam optimizer (β_1, β_2)	(0.5, 0.9)

Table 5. Hyperparameters for VQ-Diffusion.

Hyperparameter	Value
Num. transformer blocks	16
Attention type	Cross attention
Num. attention head	16
Embedding dimension	128
Block Activation	GELU2
Layer Normalization	Adaptive LN
Num. conditional frames	2
Condition embedding dimension	1024
Num. denoising steps	10
Sampling noise type	Uniform
Adaptive auxiliary loss	True
Auxiliary loss weight	10^{-3}
Training epochs	100
Batch size	4
Learning rate	4.5×10^{-4}
AdamW optimizer (β_1, β_2)	(0.9, 0.96)

Table 6. Hyperparameters for DrQv2 with Diffusion Reward.

Hyperparameter	Value
Environment	
Action repeat	3 (MetaWorld) 2 (Adroit)
Frame stack	1
Observation size	$64 \times 64 \times 3$
Reward type	Sparse
DrQv2	
Data Augmentation	± 4 RandomShift
Replay buffer capacity	10^6
Discount γ	0.99
n -step returns	3
Seed frames	4000
Exploration steps	2000
Feature dimension	50
Hidden dimension	1024
Exploration stddev. clip	0.3
Exploration stddev. schedule	Linear(1.0, 0.1, 3×10^6)
Soft update rate	0.01
Optimizer	Adam
Batch size	256
Update frequency	2
Learning rate	10^{-4}
RND	
CNN feature dimension	$7 \times 7 \times 64$
MLP size	512×512
Learning rate	10^{-4}
Diffusion Reward	
Reward coefficient α	0 (Pen) 0.95 (Others)
Sampling noise	True (scale 1)
Reward standardization \bar{r}^{ce}	True
Num. diffusion processss M	1

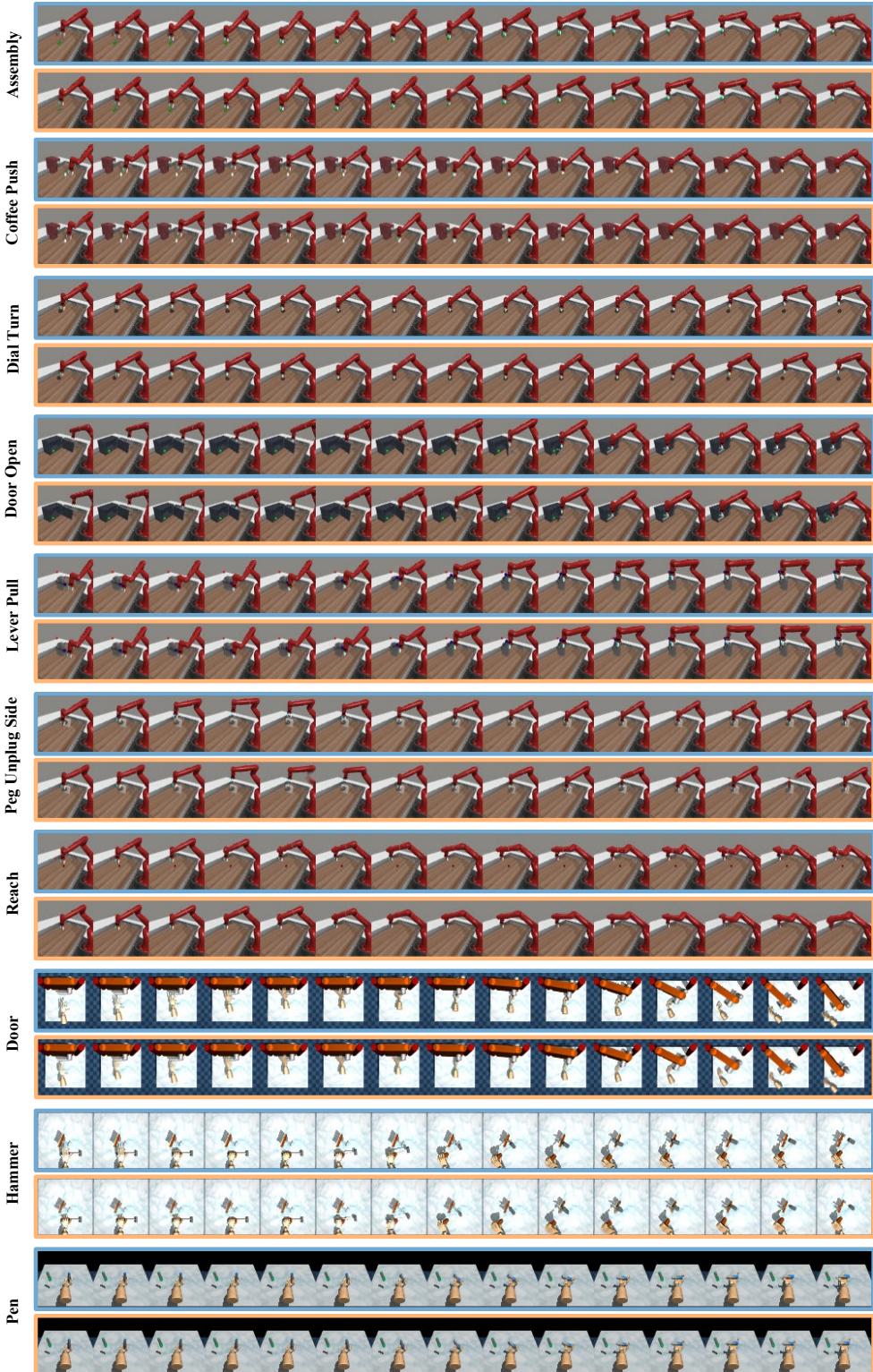


Figure 14. **Video prediction results.** Ground truth has blue borders and prediction has orange borders.

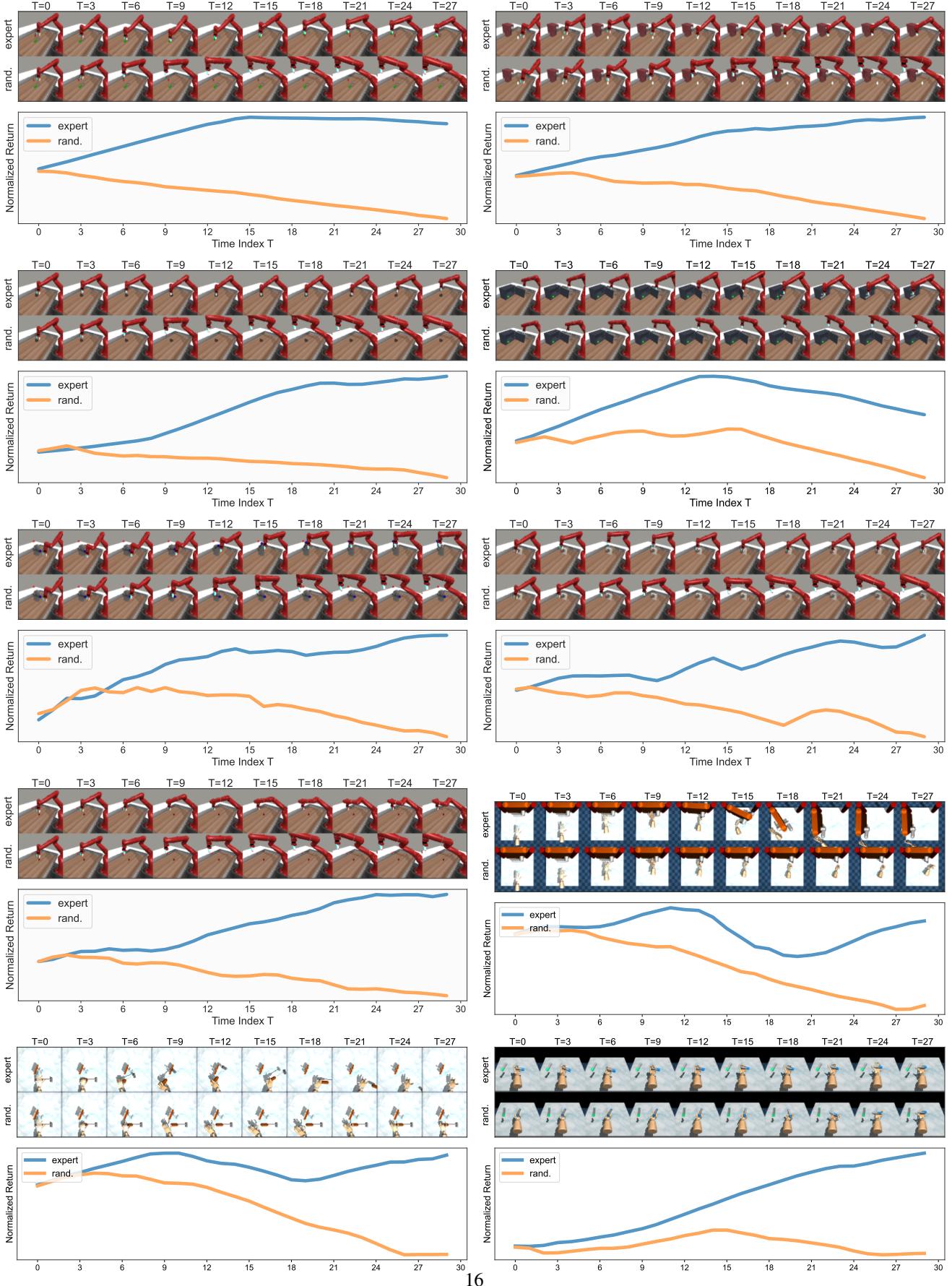


Figure 15. Reward curve of 10 simulated tasks from MetaWorld and Adroit.

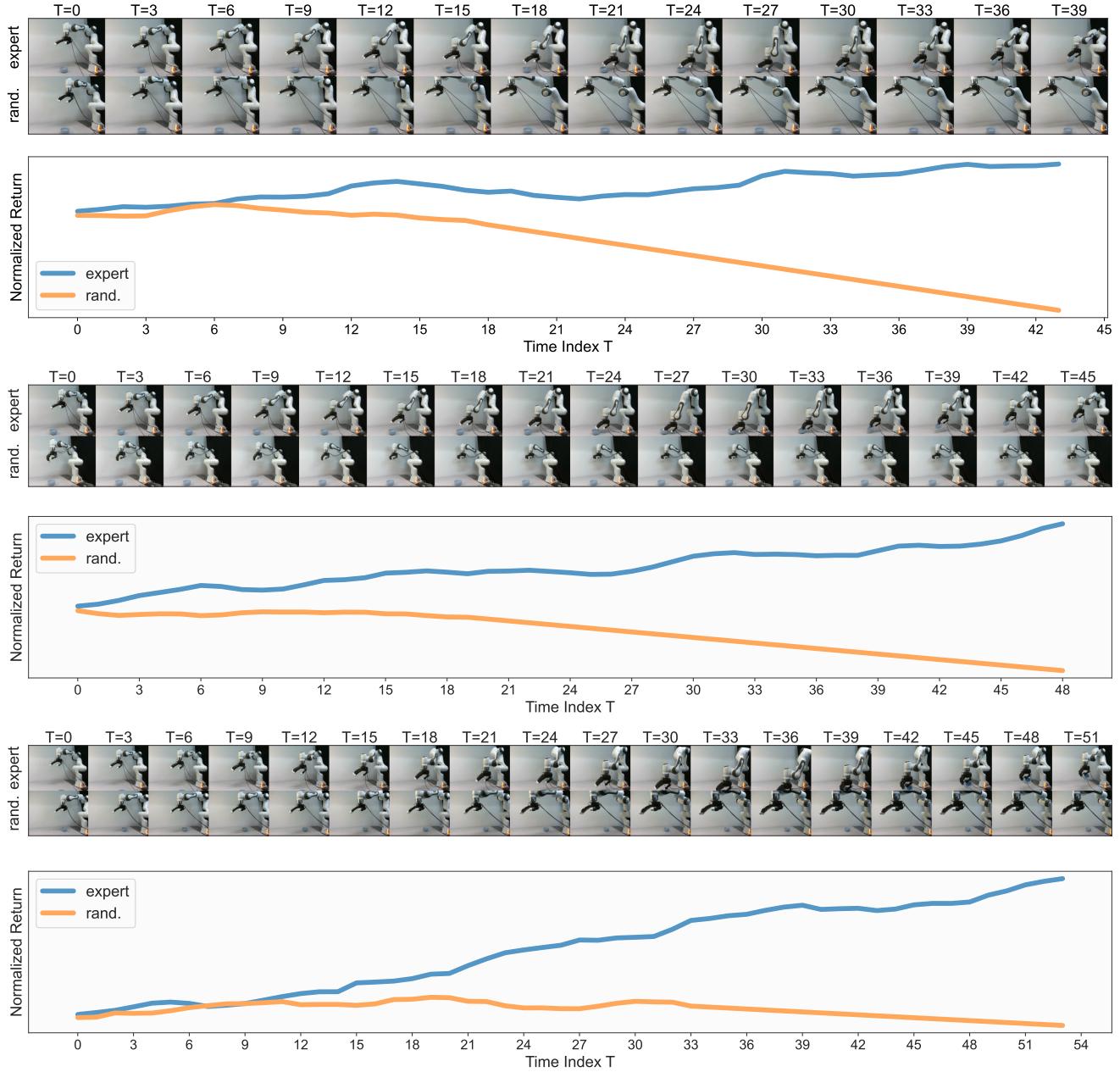


Figure 16. Reward curve of real robot trajectories.