

World4RL: Diffusion World Models for Policy Refinement with Reinforcement Learning for Robotic Manipulation

Zhennan Jiang^{1,2,3*}, Kai Liu^{1,2*}, Yuxin Qin^{1,2}, Shuai Tian^{1,2}, Yupeng Zheng^{1,2}
Mingcai Zhou⁴, Chao Yu^{5,3}, Haoran Li^{1,2†}, Dongbin Zhao^{1,2,3}

Abstract—Robotic manipulation policies are commonly initialized through imitation learning, but their performance is limited by the scarcity and narrow coverage of expert data. Reinforcement learning can refine policies to alleviate this limitation, yet real-robot training is costly and unsafe, while training in simulators suffers from the sim-to-real gap. Recent advances in generative models have demonstrated remarkable capabilities in real-world simulation, with diffusion models in particular excelling at generation. This raises the question of how diffusion model-based world models can be combined to enhance pre-trained policies in robotic manipulation. In this work, we propose World4RL, a framework that employs diffusion-based world models as high-fidelity simulators to refine pre-trained policies entirely in imagined environments for robotic manipulation. Unlike prior works that primarily employ world models for planning, our framework enables direct end-to-end policy optimization. World4RL is designed around two principles: pre-training a diffusion world model that captures diverse dynamics on multi-task datasets and refining policies entirely within a frozen world model to avoid online real-world interactions. We further design a two-hot action encoding scheme tailored for robotic manipulation and adopt diffusion backbones to improve modeling fidelity. Extensive simulation and real-world experiments demonstrate that World4RL provides high-fidelity environment modeling and enables consistent policy refinement, yielding significantly higher success rates compared to imitation learning and other baselines. More visualization results are available at <https://world4rl.github.io/>.

I. INTRODUCTION

Despite recent progress in robotic manipulation, the field still faces critical challenges for practical deployment. Imitation learning is widely used to bootstrap policies from demonstrations, but its effectiveness is constrained by the inconsistency [1] and limited diversity [2]–[4] of available datasets. Although offline reinforcement learning (RL) can extract better policies from imperfect data, its susceptibility to overestimation [5] still makes it difficult to work effectively with limited datasets. Online RL offers a natural way to refine such pre-trained policies through interaction. However, real-robot RL, while capable of overcoming dataset limitations, suffers from high interaction costs and significant safety risks that hinder large-scale training. Training

in simulation avoids these risks but inevitably introduces discrepancies from real-world physics, leading to a persistent sim-to-real gap [6].

In recent years, generative models have achieved remarkable progress in the visual domain [7], with diffusion models [8] demonstrating particularly strong performance in image [9] and video generation [10], [11]. Such generative capacity opens new opportunities for modeling complex and dynamic environments, offering a promising path toward learnable world simulators that provide realistic yet flexible environments for RL training in robotic manipulation.

Building on this idea, we introduce World4RL, a framework that systematically integrates diffusion world models into RL for robotic manipulation. World4RL follows a two-stage paradigm: we first pre-train a diffusion world model on multi-task datasets to capture diverse dynamics, and then refine policies entirely within the frozen model to avoid costly and unsafe online interactions. Serving as a high-fidelity simulator, the world model is composed of a diffusion transition model that predicts future observations conditioned on current observations and actions, and a reward classifier that provides sparse success signals, enabling policy optimization without real-world rollouts.

This design of framework contrasts with prior approaches such as IRASim [12] and NWM [13], which primarily use generative video models for planning at test time rather than for direct policy training. A closer line of work, DiWA [14], also employs world models for policy learning. However, it relies on recurrent state-space models (RSSM [15]), which lead to blurry generations and compounding rollout errors. In contrast, World4RL leverages diffusion backbones that generate sharper and temporally coherent rollouts, thereby supporting effective end-to-end reinforcement learning.

To further adapt world models to robotic manipulation, which involves high-dimensional action spaces and complex environment interactions compared to navigation [13] and games [16], we investigate two critical design choices: a two-hot action encoding [17] scheme that provides an efficient representation of continuous actions while enabling lossless reconstruction, thereby serving as a robust bridge between the RL agent and the world model, and diffusion backbone architectures that determine the fidelity and consistency of predictions. These considerations are essential for enabling diffusion world models to serve not only as visual predictors but also as reliable simulators for policy training. To this end, our work makes the following key contributions.

- We propose World4RL, a systematic framework that

¹ The State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

³ Zhongguancun Academy, Beijing, China

⁴ Beijing Zhongke Huiling Robot Technology Co, Beijing, China

⁵ Department of Electronic Engineering, Tsinghua University, Beijing, China

* Equal contribution.

† Corresponding author.

integrates diffusion world model into RL training for robotic manipulation.

- To improve modeling fidelity and enable more effective policy refinement, we design a two-hot action encoding tailored for robotic manipulation and adopt a diffusion backbone as the world model.
- We validate the effectiveness of World4RL through extensive experiments, showing that it consistently outperforms competitive baselines and significantly enhances policy refinement, improving success rates by 16% and 25% in simulation and real-robot experiments, respectively.

II. RELATED WORK

A. World Models

World models have developed rapidly, driven by advances in generative modeling. Early works such as Ha and Schmidhuber’s VAE-RNN world model [18] inspired latent dynamics models like DreamerV3 [17] and TD-MPC2 [19], while more recent Transformer-based approaches (e.g., Genie [20], Drive-WM [21]) significantly extended the temporal horizon and expressiveness. With the rise of conditional diffusion models, world models have achieved high-fidelity video prediction in diverse domains such as autonomous driving (GAIA-1 [22], DriveDreamer [23]), navigation (NWM [13]), and gaming (Diamond [16]). These advances indicate that generative world models are becoming powerful tools for simulating complex environments, motivating their application to robotic manipulation.

B. World Models in Robotic Manipulation

In robotic manipulation, early research primarily adhered to the model-based RL paradigm [17], [24], where policies must interact with the real environment to generate trajectories for updating the world model. While effective, these methods do not fully exploit the generative capabilities of modern models and still rely on online interaction, limiting their applicability to real-world systems. Recent advancements have shifted toward leveraging world models for planning, where the model predicts trajectories under different action sequences and selects the best one. For example, V-JEPA2 [25] employed self-supervised learning to train a latent action-conditioned world model for robotic planning tasks, while IRASim [12] introduced a frame-level action-conditioning module into a Diffusion Transformers (DiT) [26], significantly improving action responsiveness. DiWA [14] employs RSSM [15] as its world model to optimize the policy, where the VAE-based latent space limits image generation quality and consequently constrains the overall performance. In contrast, our World4RL framework applies a diffusion world model together with a two-hot action encoding scheme tailored for robotic manipulation, enabling accurate generation and effective policy refinement entirely within imagined rollouts.

III. METHOD

A. Overall Framework

Given a policy trained with imitation learning from expert demonstrations, our goal is to further improve its performance in robotic manipulation tasks. To this end, we propose the World4RL framework, which addresses the limitations of reinforcement learning (RL), where training in simulators suffers from the inevitable sim-to-real gap, while training directly on real robots incurs expensive and risky interactions. The core idea is to leverage a high-fidelity diffusion world model that enables agents to improve policies entirely in imagined environments. World4RL consists of three key components:

- **Diffusion Transition Model:** Serves as a dynamics approximator, predicting future observations conditioned on current observations and actions.
- **Reward Classifier:** Considering that robotic manipulation tasks typically involve sparse rewards (e.g., success/failure signals), we introduce a binary reward classifier to evaluate imagined rollouts generated by the world model.
- **RL-refined Policy:** Initialized with the gaussian policy to provide a stable starting point, and subsequently refined within the world model using the Proximal Policy Optimization (PPO [27]) algorithm.

The overall training pipeline consists of two stages: pre-training and policy optimization. In the pre-training stage, the diffusion transition model is trained on task-agnostic data [28] to generalize across diverse dynamics, the reward classifier is trained on task-specific data annotated with binary success labels, and the policy is trained via imitation learning to provide a stable initialization. In the policy optimization stage, the pre-trained world model is frozen and used as a simulator, while the policy is refined with PPO under sparse rewards through imagined rollouts. This design improves both sample efficiency and safety, while enabling consistent gains over the initial gaussian policy. An overview of the framework is shown in Fig. 1.

B. Pre-training Stage

1) *Policy Pre-training:* We first pre-train the policy through imitation learning. This stage provides an imitation-based initialization from expert demonstrations, ensuring that the learned policy π_ξ can already execute reasonable actions.

Formally, given a dataset of expert demonstrations $\{(x_t, a_t)\}$, the policy parameters ξ are optimized by minimizing the mean squared error (MSE) between the predicted action $\pi_\xi(x_t)$ and the expert action a_t :

$$\mathcal{L}_{BC}(\xi) = \mathbb{E}_t \|a_t - \pi_\xi(x_t)\|^2. \quad (1)$$

2) *Diffusion Transition Model:* Diffusion models are a class of generative models that learn to reverse a gradual noising process, typically formulated as a stochastic differential equation (SDE). By training a neural network to denoise perturbed samples at different noise levels, the model can generate realistic data through iterative denoising. To

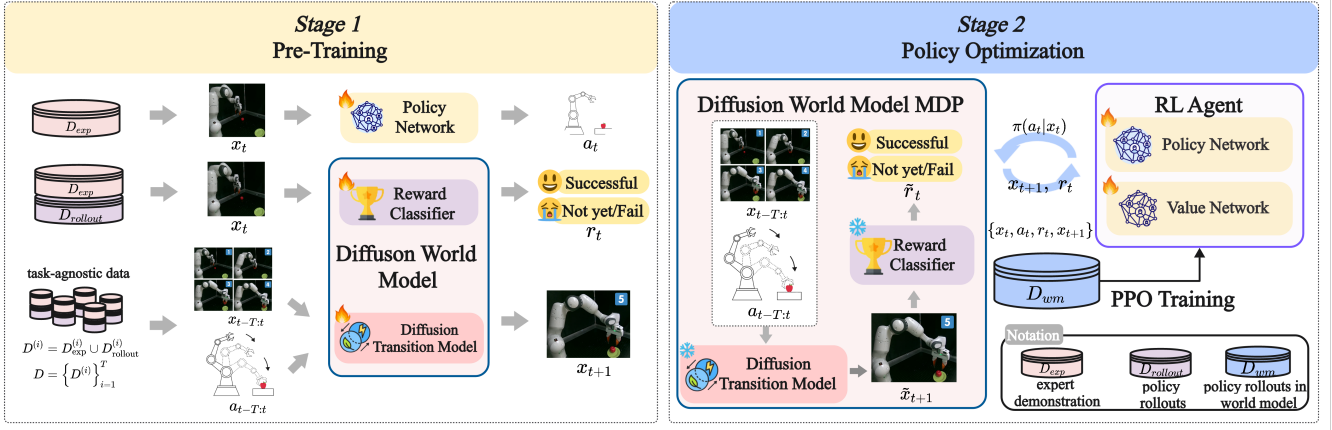


Fig. 1: **Overview of the proposed World4RL framework.** Stage 1 (Pre-training) trains the diffusion transition model on task-agnostic data, optimizes the reward classifier on task-specific success-annotated data, and initializes the policy through imitation learning with expert demonstrations. Stage 2 (Policy Optimization) freezes the pre-trained world model and employs reinforcement learning entirely within imagined rollouts.

stabilize training, we follow EDM [29], which introduces a preconditioned denoising function

$$D_\theta(x^\tau; \tau, c) = c_{\text{skip}}^\tau x^\tau + c_{\text{out}}^\tau F_\theta(c_{\text{in}}^\tau x^\tau; c_{\text{noise}}^\tau, c), \quad (2)$$

where F_θ is the learnable network, τ is the diffusion timestep, and c denotes conditioning information.

Building on this framework, we train a diffusion transition model D_θ that maps a finite history of past T observations $x_{t-T:t}^0$ ¹ and corresponding actions $a_{t-T:t}$ to the next observation x_{t+1}^0 .

To better handle continuous action inputs within the world model, we adopt a two-hot encoding scheme, inspired by DreamerV3 [17]. Unlike one-hot discretization [16], latent-space representation (e.g. VQ-VAE [30]), or token-based approaches (e.g., FAST [31]), two-hot encoding provides a lossless and differentiable representation without introducing reconstruction errors. For each action dimension $a_i \in \mathbb{R}$, given bin values $\mathcal{B} = \{b_1, \dots, b_K\}$, we map a_i to its two nearest bins:

$$\mathbf{t}_i[k] = \frac{b_{k+1} - a_i}{b_{k+1} - b_k}, \quad \mathbf{t}_i[k+1] = \frac{a_i - b_k}{b_{k+1} - b_k}, \quad (3)$$

with $\sum_j \mathbf{t}_i[j] = 1$ and $b_k \leq a_i \leq b_{k+1}$, where $\mathbf{t}_i \in \mathbb{R}^K$ denotes the two-hot weight vector for the i -th action dimension. This interpolation-based representation preserves continuity while embedding a lightweight discrete structure. In practice, two-hot encoding achieves fine-grained modeling with a moderate number of bins (e.g., $K = 21$) and can be optimized end-to-end with policy networks. We denote the encoded action representation as z , and replace the original action $a_{t-T:t}$ with $z_{t-T:t}$ as conditional input to the diffusion model.

¹We denote the observation as x^0 to maintain consistency with the notation in diffusion models. For simplicity, we occasionally use x to denote the original image observation x^0 .

Based on this, the diffusion transition model D_θ is formally defined as a denoising process conditioned on historical observations $x_{t-T:t}^0$ and encoded actions $z_{t-T:t}$, and (2) can be rewrote as follows:

$$D_\theta(x^\tau, x_{t-T:t}^0, z_{t-T:t}) = c_{\text{skip}}^\tau x^\tau + c_{\text{out}}^\tau F_\theta(c_{\text{in}}^\tau x^\tau; c_{\text{noise}}^\tau, x_{t-T:t}^0, z_{t-T:t}). \quad (4)$$

After that, we obtain the training objective:

$$\mathcal{L}_D(\theta) = \mathbb{E}_{x^\tau \sim p_\tau} \left[\left\| \mathbf{F}_\theta(c_{\text{in}}^\tau x_{t+1}^\tau, \tau, x_{t-T:t}^0, z_{t-T:t}) - \frac{1}{c_{\text{out}}^\tau} (x_{t+1}^0 - c_{\text{skip}}^\tau x_{t+1}^\tau) \right\|^2 \right]. \quad (5)$$

In practice, we follow the design principles of EDM [29] in selecting the noise schedule and hyperparameters (e.g., the design of c_{in} and c_{out}). For the network architecture, we employ U-Net [32] 2D to model F_θ .

3) *Reward Classifier*: In real robotic systems, reward functions are often difficult to handcraft. Since our experiments primarily focus on sparse rewards, we introduce a reward classifier that provides a binary signal indicating whether the agent has reached a successful state. Given the next observation x_{t+1} , the classifier C_ψ outputs the probability that the next observation is a success, defined as

$$r(s_t, a_t) := C_\psi(x_{t+1}). \quad (6)$$

Architecturally, we employ a pre-trained ResNet18 [33] as the visual backbone for feature extraction. The classifier is trained on expert demonstrations $\{(x_i, r_i)\}$, where $r_i \in \{0, 1\}$. Its parameters are optimized by minimizing the binary cross-entropy loss:

$$\mathcal{L}_C(\psi) = -\frac{1}{N} \sum_{i=1}^N \left[r_i \log C_\psi(x_i) + (1-r_i) \log (1-C_\psi(x_i)) \right]. \quad (7)$$

After training, the classifier is frozen and used during reinforcement learning to provide a binary success/failure reward signal, thereby avoiding complex reward engineering while ensuring reliable sparse feedback.

C. Policy Optimization Stage

Algorithm 1 Algorithm of World4RL (Pre-training and Policy Optimization)

Input:

$D_\theta, C_\psi, \pi_\xi, V_\phi$: diffusion transition model, reward classifier, policy, value network

$\mathcal{D}_{\text{exp}}, \mathcal{D}_{\text{rollout}}, \mathcal{D}_{\text{wm}}$: expert demos, imitation/random rollouts, world-model rollouts buffers

Pre-training Stage

- 1: **(Policy pre-training)**
- 2: Sample $(x_t^0, a_t) \sim \mathcal{D}_{\text{exp}}$
- 3: Compute $\mathcal{L}_{BC}(\xi)$ using Eq. (1)
- 4: $\xi \leftarrow \xi - \alpha \nabla_\xi \mathcal{L}_{BC}(\xi)$ ▷ update policy
- 5: **(Diffusion transition model pre-training)**
- 6: Sample $(x_{t-T:t}^0, a_{t-T:t}, x_{t+1}^0) \sim \mathcal{D}_{\text{exp}} \cup \mathcal{D}_{\text{rollout}}$
- 7: $z_{t-T:t} \leftarrow \text{TwoHot}(a_{t-T:t})$
- 8: Sample $\tau \sim \mathcal{U}[0, T]$ and construct x_{t+1}^τ by forward noising of x_{t+1}^0
- 9: Compute $\mathcal{L}_D(\theta)$ using Eqs. (4) and (5)
- 10: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_D(\theta)$ ▷ update diffusion
- 11: **(Reward classifier pre-training)**
- 12: Sample $(x_i^0, y_i) \sim \mathcal{D}_{\text{exp}}$ ▷ $y_i \in \{0, 1\}$ success label
- 13: Compute $\mathcal{L}_C(\psi)$ with BCE (cf. Eq. (7))
- 14: $\psi \leftarrow \psi - \alpha \nabla_\psi \mathcal{L}_C(\psi)$ ▷ update classifier

Policy Optimization Stage

- 1: **for** epoch = 1 ... max_epochs **do**
 - 2: Observe context $x_{t-T:t}$
 - 3: $a_t \sim \pi_\xi(\cdot | x_t); \quad z_t \leftarrow \text{TwoHot}(a_t)$
 - 4: $\tilde{x}_{t+1} \leftarrow \text{Sample from } D_\theta(\cdot; x_{t-T:t}, z_{t-T:t})$
 - 5: $r_t \leftarrow C_\psi(\tilde{x}_{t+1}) \in \{0, 1\}$
 - 6: $\mathcal{D}_{\text{wm}} \leftarrow \mathcal{D}_{\text{wm}} \cup (x_t, a_t, r_t, \tilde{x}_{t+1})$
 - 7: **if** $|\mathcal{D}_{\text{wm}}| \geq \text{ppo_batch_size}$ **then**
 - 8: UPDATEPPO(\mathcal{D}_{wm})
 - 9: $\mathcal{D}_{\text{wm}} \leftarrow \emptyset$
 - 10: **end if**
 - 11: **end for**
-

Our objective is to learn an agent π_ξ that maximizes the expected cumulative rewards in the world model. To achieve this, we adopt PPO [27], which separately optimizes a policy model π_ξ and a value model V_ϕ . The policy objective is defined as

$$\mathcal{L}_P(\xi) = \mathbb{E}_t[\min(\rho_t(\xi)A_t(x_t, a_t), \text{clip}(\rho_t(\xi), 1 - \epsilon, 1 + \epsilon)A_t(x_t, a_t))], \quad (8)$$

where $\rho_t(\xi) = \frac{\pi_\xi(a_t|x_t)}{\pi_{\xi_{\text{old}}}(a_t|x_t)}$ is the probability ratio between the updated policy π_ξ and the reference (old) policy $\pi_{\xi_{\text{old}}}$ that was used to collect trajectories, $A_t(x_t, a_t)$ is the advantage function, and ϵ is a hyperparameter controlling the clipping range.

The value function is optimized with the regression objective:

$$\mathcal{L}_V(\phi) = \mathbb{E}_t[(V_\phi(x_t) - (r_t + \gamma V_\phi(x_{t+1})))^2]. \quad (9)$$

During optimization, the frozen diffusion world model D_θ is used to generate imagined rollouts conditioned on past observations $x_{t-T:t}^0$ and encoded actions $z_{t-T:t}$. These synthetic trajectories serve as a substitute for real-world interaction, greatly reducing training costs and avoiding hardware risks. The reward classifier C_ψ evaluates each imagined next state, providing a sparse binary reward signal $R(s_t, a_t) \in \{0, 1\}$.

The policy π_ξ interacts with the world model in the following loop:

- Given the current observation x_t^0 , the policy outputs an action a_t , which is then discretized via two-hot encoding into z_t .
- The diffusion transition model predicts the next observation x_{t+1}^0 conditioned on $x_{t-T:t}^0$ and $z_{t-T:t}$.
- The reward classifier evaluates x_{t+1}^0 to provide a binary success/failure reward.
- The policy and value models are updated via PPO [27] using the generated rollouts.

This integration of PPO [27] with imagined rollouts allows the policy to efficiently explore and improve in sparse reward settings, while the pre-training ensure stable initialization. Together, these design choices enable World4RL to achieve both sample efficiency and robust performance in robotic manipulation tasks.

IV. EXPERIMENTS

We conduct extensive experiments to evaluate the effectiveness of World4RL. Our goal is to answer three key questions: 1) Can World4RL accurately model fine-grained robotic manipulation tasks and capture task-specific dynamics? 2) Does World4RL facilitate reinforcement learning by enabling more efficient policy training and achieving superior performance compared with strong baselines, including imitation learning, offline reinforcement learning, and planning methods? 3) Can World4RL maintain strong performance when deployed on real robot platforms?

To assess the capability of World4RL as a generative world model, we adopt three widely used metrics:

- **LPIS** [34]: perceptual similarity between predictions and ground truth;
- **FID** [35]: distributional quality of generated images;
- **FVD** [36]: video-level consistency capturing both spatial and temporal fidelity.

For this part, we compare against three representative video prediction models: **NWM** [13], a DiT-based dynamics world model designed for temporal sequence modeling, **iVideoGPT** [37], an autoregressive transformer framework with compressive tokenization for multimodal prediction, and **DiWA** [14], a RSSM-based world model designed for policy adaptation.

To evaluate the effectiveness of World4RL in policy learning, we adopt task success rate (SR) as the primary metric. We conduct experiments on the Meta-World benchmark [38], which provides a diverse suite of robotic manipulation tasks commonly used for evaluating policy performance. We compare World4RL against a broad spectrum of baselines, all of which are evaluated in the fixed dataset without additional online interaction, including

- Imitation learning: behavior cloning with **gaussian policy** and Diffusion Policy (DP) [39];
- Offline Reinforcement Learning: **TD3+BC** [40] and **IQL** [41];
- Planning-based method: **IRASim-ft** [12].

For a fair comparison, we adopt planning method as IRASim in our world model, which we call **IRASim-ft**: at test time, multiple candidate trajectories are sampled from the world model, their cumulative rewards $\sum \tilde{r}$ are evaluated with a learned reward model, and the first action of the highest-reward trajectory is executed.

In addition, we compare against hybrid offline-to-online approaches such as **Uni-O4** [42] and **RLPD** [43], and observe the number of online samples they require to reach the same level of performance. This directly assesses the sample efficiency of World4RL.

With the evaluation metrics and baselines established, we now present three sets of experiments to assess the capability of World4RL: modeling robotic dynamics, enhancing policy learning, and transferring to real-world robots.

A. Can World4RL Accurately Simulate Robotic Manipulation Environments?

We collect training data from six representative environments in the Meta-World benchmark. For each task, we gather 50 expert trajectories, 150 trajectories generated by a pre-trained gaussian policy, and 30 trajectories from a random policy rollout. Each trajectory contains 50 timesteps. During training, the model is conditioned on a history of four consecutive frames along with their corresponding actions, and is required to predict future observations. At test time, the model receives only the initial frame and action as input and autoregressively generates the subsequent video sequence. This setup allows us to test not only whether the model can predict short-term dynamics accurately, but also whether it can maintain coherent and stable rollouts over longer horizons through autoregressive generation.

We evaluate the fidelity of learned dynamics on Meta-World, as summarized in Table I. World4RL consistently achieves the lowest FVD, FID, and LPIPS scores under both policy and random rollouts, significantly outperforming NWM [13], iVideoGPT [37] and DiWA [14]. In particular, DiWA exhibits notably poor quantitative results on our multi-task datasets setting, and even its single-task variant (DiWA-ST) still lags far behind. These results highlight the clear advantage of diffusion-based architectures in maintaining temporal consistency and visual fidelity.

In addition to quantitative metrics, we provide visualizations in Fig. 2. World4RL produces more coherent and

TABLE I: Quantitative results on video prediction. “ST” denotes single-task training and evaluation.

Model	FVD ↓		FID ↓		LPIPS ↓	
	Policy	Random	Policy	Random	Policy	Random
World4RL (Ours)	326.5	400.1	17.07	23.43	0.0192	0.0246
NWM [13]	547.4	851.9	30.49	34.88	0.0268	0.0259
iVideoGPT [37]	450.3	531.3	18.65	20.69	0.0256	0.0283
DiWA [14]	803.6	1231.0	62.93	96.47	0.0804	0.1364
DiWA (ST)	644.8	880.2	35.08	52.77	0.0523	0.0596

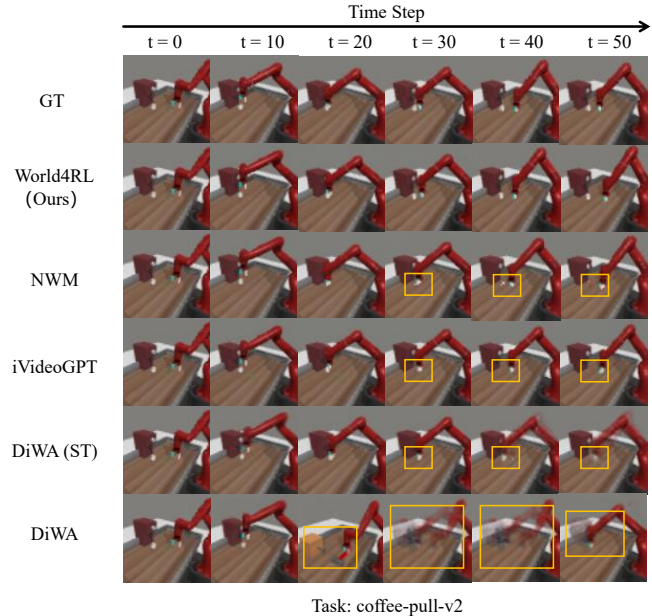


Fig. 2: Visualization of predicted rollouts on the Coffee-Pull-v2 task. The ground truth (GT) trajectory corresponds to a failed execution, where the robot does not successfully pull the cup. World4RL accurately models this failure trajectory, faithfully capturing the underlying dynamics, while baseline models (NWM [13], iVideoGPT [37], and DiWA [14]) incorrectly generate successful executions.

physically consistent rollouts, closely matching ground-truth trajectories, while baseline models often generate blurrier predictions or inconsistent dynamics. Notably, when given failed execution trajectories, World4RL can still faithfully model the underlying failure dynamics, whereas DiWA not only suffers from blurry and inconsistent rollouts but occasionally generates scenes from other tasks, underscoring its inability to generalize to multi-task settings. Together, these findings confirm that our diffusion world models provide superior capacity to capture fine-grained robotic interactions and long-horizon dynamics compared with other architectures.

B. Can World4RL Enhance Policy Learning?

We next investigate whether World4RL can facilitate policy learning and outperform existing approaches. As outlined in Sec. IV, our baselines include imitation learning, offline RL, and planning-based methods.

For a fair evaluation under realistic conditions, although Meta-World provides dense rewards, we instead adopt sparse success signals, which better reflect real robotic scenarios where dense reward shaping is often unavailable. For imitation learning, the training dataset contains 50 expert demonstrations, while offline RL methods are trained with 50 expert trajectories plus 150 rollouts collected from a BC policy.

Table II reports the success rates across tasks. World4RL consistently outperforms baselines. These results demonstrate that integrating World4RL into policy training yields more effective learning under sparse reward conditions. In contrast, IRASim-ft, while competitive in some tasks, relies on exhaustive trajectory sampling and reward evaluation at test time, which incurs up to $40\times$ higher computational cost compared to World4RL.

To further examine sample efficiency, we also compare World4RL with two representative offline-to-online approaches, Uni-O4 [42] and RLPD [43], both of which rely on substantial online interaction. In contrast, World4RL is trained entirely on fixed datasets without any online samples. As illustrated in Fig. 3, World4RL already achieves comparable or superior performance with only expert and policy rollout data, while RLPD and Uni-O4 require 346k and 470k online steps, respectively, to reach the same level. This demonstrates the strong sample efficiency of World4RL, making it particularly suitable for real-robot deployment where online interaction is expensive and limited.

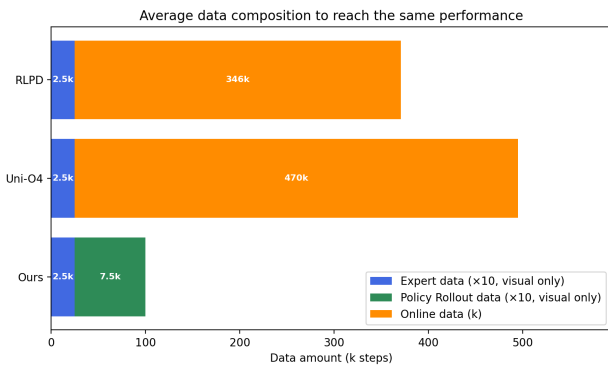


Fig. 3: Comparison of online sample efficiency. World4RL achieves comparable performance on fixed datasets, whereas RLPD and Uni-O4 require over additional 300k online steps.

C. Can World4RL Transfer to Real-World Robots well?

We evaluate World4RL on six real-world manipulation tasks using a Franka Emika Panda robot, shown in Fig. 4. Following the HIL-SERL [44] protocol, we collected data via teleoperation with a space mouse. For each task, the training dataset comprised 50 human expert demonstrations, 50 trajectories generated by a pre-trained gaussian policy, and 50 trajectories from a random policy. Following the methodology in Sec. III, World4RL first pre-trains the policy and the diffusion world model, and then optimizes the policy

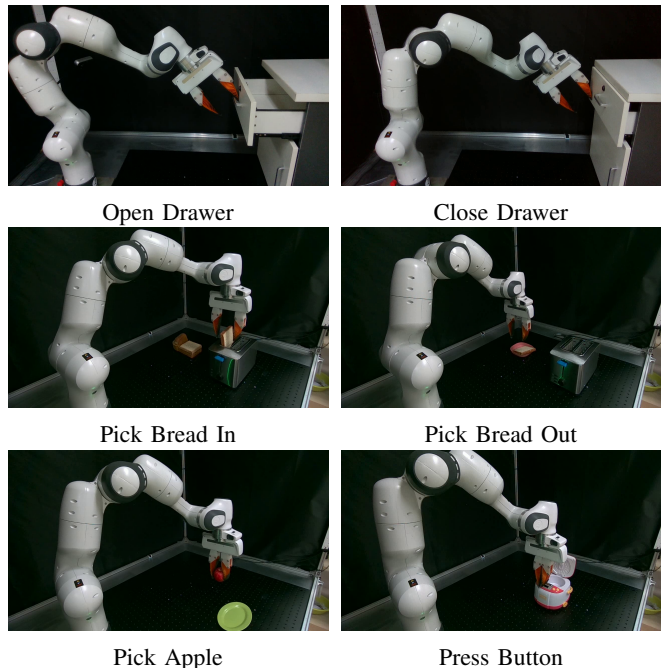


Fig. 4: Real World Tasks

with imagined rollouts, without requiring any additional real-world interaction.

During evaluation, the initial scene configuration and robot starting pose are fixed for each task, and we execute 20 rollouts in the physical environment to measure success rates. We compare World4RL against the initial pre-trained gaussian policy and diffusion policy to assess whether our framework can deliver consistent improvements in real-world policy performance.

Table III reports the success rates across six real-world tasks. World4RL achieves the highest average success rate of 93.3%, significantly outperforming imitation learning methods. Beyond achieving higher success rates, we also observe that policies fine-tuned with World4RL tend to execute tasks more decisively, completing pick-and-place behaviors quickly and accurately; for example, in the *put bread in* task, the fine-tuned policy promptly performs the grasping and placing actions, whereas gaussian policy and diffusion policy [39] often show hesitation or linger in intermediate states without committing to task completion.

V. ABLATION STUDY

To better understand the design choices in World4RL, we conduct ablation experiments focusing on the action encoding strategy of the world model. Our goal is to assess how different encoding mechanisms influence the fidelity of learned dynamics and their impact on downstream policy optimization.

Accurate action encoding is not only critical for world model learning, but also serves as the essential bridge that connects the world model with the RL agent. Many prior works simply adopt a linear MLP [12], [13] to parameterize actions. While effective in domains such as autonomous

TABLE II: Success rate of different methods on Meta-World benchmark over 3 seeds. The notation $\uparrow n$ indicates the absolute improvement over pre-trained gaussian policy.

Task	Imitation Learning		Offline Reinforcement Learning		World Model based Methods	
	Gaussian Policy	DP [39]	TD3+BC [40]	IQL [41]	IRASim-ft [12]	WORLD4RL (Ours)
coffee-pull-v2	47 \pm 7	34 \pm 7	57 \pm 13	47 \pm 9	55 \pm 6	68 \pm 5 \uparrow 21
soccer-v2	18 \pm 2	19 \pm 4	22 \pm 8	14 \pm 7	28 \pm 5	31 \pm 4 \uparrow 13
hammer-v2	79 \pm 5	15 \pm 6	89 \pm 4	73 \pm 10	82 \pm 5	91 \pm 4 \uparrow 12
door-lock-v2	74 \pm 5	86 \pm 8	82 \pm 6	69 \pm 11	78 \pm 5	92 \pm 5 \uparrow 14
lever-pull-v2	31 \pm 5	49 \pm 5	39 \pm 11	24 \pm 3	33 \pm 5	52 \pm 8 \uparrow 21
handle-pull-v2	60 \pm 6	67 \pm 11	57 \pm 6	25 \pm 6	66 \pm 6	71 \pm 4 \uparrow 11
Average SR	51.5	45.0	57.7	42.0	57.0	67.5 \uparrow 16

TABLE III: Real-world success rates across 6 manipulation tasks (20 trials per task).

Task	Gaussian Policy	DP [39]	WORLD4RL (Ours)
Pick bread out	13/20	19/20	20/20
Pick apple	8/20	15/20	19/20
Press button	12/20	16/20	18/20
Put bread in	12/20	18/20	16/20
Open drawer	12/20	18/20	19/20
Close drawer	15/20	20/20	20/20
Average SR	68.3%	88.3%	93.3% \uparrow 25

driving, we find it insufficient for robotic manipulation, where actions often carry more complex semantics.

Beyond simple function approximation, action representation in robotics has been studied through alternative strategies, including discretization (e.g. dividing into bins, also called as ont-hot encoding [16]), latent-space embeddings (e.g. VAE [45]), and tokenization (e.g. FAST tokenizer [31]). While these methods aim to introduce more structured action representations, they inevitably suffer from lossy reconstruction during encoding and decoding, which degrade the predictive fidelity of world models and hinder downstream reinforcement learning, particularly in fine-grained robotic manipulation tasks.

We systematically evaluate different action encoding strategies in our framework, including linear projection, one-hot, VQ-VAE [30] and FAST, on the same dataset as in Sec. IV-A. As shown in Table IV, our proposed two-hot encoding consistently outperforms other approaches. These results highlight the critical role of action encoding in world model learning. Beyond performance gaps, approaches such as binning, FAST, and VQ-VAE introduce lossy action reconstruction, which not only limits model fidelity but also undermines downstream reinforcement learning by introducing unstable policy gradients. In contrast, the two-hot scheme provides a lossless and differentiable representation, enabling both robust world modeling and stable RL training.

VI. CONCLUSION AND FUTURE WORK

In this work, we proposed World4RL, a framework that systematically incorporates diffusion models into reinforcement learning for robotic manipulation. Experimental results demonstrate that World4RL not only serves as a high-fidelity world model capable of accurately modeling trajectories,

TABLE IV: Comparison of different action encoding strategies on Meta-World video prediction.

Model	FVD \downarrow		FID \downarrow		LPIPS \downarrow	
	Policy	Random	Policy	Random	Policy	Random
Two-hot (Ours)	326.5	400.1	17.07	23.43	0.0192	0.0246
One-hot [16]	350.3	471.5	18.52	26.24	0.0193	0.0257
Linear [13]	353.4	514.0	17.85	23.83	0.0218	0.0250
FAST [31]	407.0	748.0	28.92	36.52	0.0284	0.0409
VQ-VAE [30]	525.6	860.0	28.60	43.25	0.0506	0.0633

but also functions as a real-time simulator that enables efficient policy refinement under sparse reward conditions. These findings highlight the potential of diffusion models as a unifying bridge between visual prediction and reinforcement learning, facilitating consistent improvement of pre-trained policies beyond imitation learning.

Although diffusion world models can effectively capture environment dynamics, their modeling capacity is ultimately constrained by the action distribution present in limited offline datasets. During reinforcement learning, the policy may explore actions that fall outside this distribution, resulting in inaccurate rollouts from the world model and thereby restricting policy improvement. A feasible avenue for future work is to incorporate a small amount of real-world interaction data to update the world model, which could in turn enhance its fidelity and enable further policy refinement. In addition, our current design relies on sparse binary rewards to reflect real-world constraints, but this often results in inefficient exploration and potential out-of-distribution behaviors. Introducing denser reward signals could mitigate such issues and promote more stable and effective policy optimization.

REFERENCES

- [1] C. Xu, Q. Li, J. Luo, and S. Levine, “RLDG: Robotic Generalist Policy Distillation via Reinforcement Learning,” in *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025.
- [2] S. Cabi, S. G. Colmenarejo, A. Novikov, *et al.*, “Scaling data-driven robotics with reward sketching and batch reinforcement learning,” *Robotics: Science and Systems XVI*, vol. 9 1, p. 0, 2019.
- [3] W. Cui, C. Zhao, S. Wei, J. Zhang, H. Geng, Y. Chen, and H. Wang, “Gapartmanip: a large-scale dataset for generalizable and actionable part manipulation with material-agnostic articulated objects,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2025.

- [4] X. Liu, Y. Chen, and H. Li, "Sample-efficient unsupervised policy cloning from ensemble self-supervised labeled videos," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 3632–3639.
- [5] A. Kumar, J. Fu, M. Soh, *et al.*, "Stabilizing off-policy q-learning via bootstrapping error reduction," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [6] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*. IEEE, 2020, pp. 737–744.
- [7] T. Karras, M. Aittala, S. Laine, *et al.*, "Alias-free generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2021, pp. 852–863.
- [8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.
- [9] C. Chen, J. Zhu, X. Feng, *et al.*, "S²-guidance: Stochastic self guidance for training-free enhancement of diffusion models," *arXiv preprint arXiv:2508.12880*, 2025.
- [10] J. Ho, T. Salimans, A. A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," in *Advances in neural information processing systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022.
- [11] Y. Chen, H. Li, Z. Jiang, H. Wen, and D. Zhao, "Tevir: Text-to-video reward with diffusion models for efficient reinforcement learning," *arXiv preprint arXiv:2505.19769*, 2025.
- [12] F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong, "Irasim: Learning interactive real-robot action simulators," *arXiv:2406.12802*, 2024.
- [13] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, "Navigation world models," *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 791–15 801, 2024.
- [14] A. L. Chandra, I. Nematollahi, C. Huang, T. Welschhold, W. Burgard, and A. Valada, "Diwa: Diffusion policy adaptation with world models," *Conference on Robot Learning (CoRL)*, 2025.
- [15] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," in *International Conference on Learning Representations*, 2021.
- [16] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret, "Diffusion for world modeling: Visual details matter in atari," in *Thirty-eighth Conference on Neural Information Processing Systems*.
- [17] D. Hafner, J. Pasukonis, J. Ba, *et al.*, "Mastering diverse control tasks through world models," *Nature*, vol. 640, pp. 647–653, 2025.
- [18] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 2455–2467.
- [19] N. Hansen, H. Su, and X. Wang, "Td-mpc2: Scalable, robust world models for continuous control," in *International Conference on Learning Representations (ICLR)*, 2024.
- [20] J. Bruce, M. D. Dennis, A. Edwards, *et al.*, "Genie: Generative interactive environments," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
- [21] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang, "Driving into the future: Multi-view visual forecasting and planning with world model for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 749–14 759.
- [22] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, "Gaia-1: A generative world model for autonomous driving," *arXiv preprint*, 2023.
- [23] X. Wang, Z. Zhu, G. Huang, X. Chen, J. G. Zhu, and J. Lu, "Drive-dreamer: Towards real-world driven world models for autonomous driving," *arXiv preprint*, 2023.
- [24] J. Chai, Y. Fu, D. Zhao, and Y. Zhu, "Aligning credit for multi-agent cooperation via model-based counterfactual imagination," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Auckland, New Zealand, May 2024, pp. 281–289.
- [25] M. Assran, A. Bardes, D. Fan, *et al.*, "V-JEPA 2: Self-supervised video models enable understanding, prediction and planning," *CoRR*, vol. abs/2506.09985, 2025.
- [26] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 4172–4182.
- [27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.
- [28] W. Harvey, S. Naderiparizi, V. Masrani, C. D. Weilbach, and F. Wood, "Flexible diffusion modeling of long videos," in *Advances in Neural Information Processing Systems*, 2022.
- [29] T. Karras, M. Aittala, S. Laine, and T. Aila, "Elucidating the design space of diffusion-based generative models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2022.
- [30] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafullah, and L. Pinto, "Behavior generation with latent actions," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235. PMLR, 21–27 Jul 2024, pp. 26 991–27 008.
- [31] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," 01 2025.
- [32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [35] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.
- [36] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.
- [37] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, "videogpt: Interactive videogpts are scalable world models," in *Advances in Neural Information Processing Systems*, 2024.
- [38] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *3rd Annual Conference on Robot Learning, CoRL 2019*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 2019, pp. 1094–1100.
- [39] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023.
- [40] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- [41] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- [42] K. Lei, Z. He, C. Lu, K. Hu, Y. Gao, and H. Xu, "Uni-o4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.
- [43] P. J. Ball, L. M. Smith, I. Kostrikov, and S. Levine, "Efficient online reinforcement learning with offline data," in *International Conference on Machine Learning, ICML*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 2023, pp. 1577–1594.

- [44] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, p. eads5033, 2025.
- [45] Y. Wang, H. Zhu, M. Liu, J. Yang, H.-S. Fang, and T. He, "Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.