WHALE-X: LEARNING SCALABLE EMBODIED WORLD MODELS WITH ENHANCED GENERALIZABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

World models play a crucial role in decision-making within embodied environments, enabling cost-free explorations that would otherwise be expensive in the real world. However, to support faithful imagination in out-of-distribution (OOD) regions, world models must possess significant generalizability, which poses substantial challenges for previous scalable approaches. This paper addresses two primary sources of the world model generalization error: the *policy distribution shift* caused by the divergence between test and data-collection policies, and the *compounding error* arising from long-horizon autoregressive rollout. To tackle these issues, we introduce the *policy-conditioning* and the *retracing-rollout* techniques, respectively. Incorporating these two techniques, we present Whale, a scalable spatial-temporal transformer-based world model with enhanced generalizability. We first demonstrate the effectiveness of the two techniques, showcasing their consistent superiority over previous baselines in both trajectory generation quality and value estimation accuracy. Furthermore, we propose Whale-X, a 414M parameter world model trained on 970K trajectories from Open X-Embodiment datasets. We show that Whale-X exhibits promising scalability and strong generalizability in real-world manipulation scenarios using minimal demonstrations.

1 INTRODUCTION

Human beings have the capability to envision an imagined world in their minds, predicting how different actions might lead to different outcomes (Maus et al., 2013; Nortmann et al., 2015). Inspired by this aspect of human intelligence, world models (Ha & Schmidhuber, 2018) are designed to abstract real-world dynamics and provide such *"what if"* prediction. As a result, embodied agents can interact with world models instead of real-world environments to generate simulation data, which can be used for various downstream tasks, including counterfactual prediction (Chen et al., 2023), off-value estimation (Fu et al., 2021), and offline reinforcement learning (Levine et al., 2020). However, the requirement for accurate out-of-distribution (OOD) predictions for reliable model imagination poses significant challenges to the generalizability of world models, which has not been well addressed by previous approaches (Schubert et al., 2023).

In this work, we investigate the sources of the generalization error in world models, identifying two primary factors: 1) *policy distribution shift* (Janner et al., 2019), stemming from the divergence between the test policy and data-collection policies, and 2) *error compounding* (Xu et al., 2020), resulting from long-horizon autoregressive rollouts. The interplay of these two factors intensifies the challenge of generalization in world models.

To mitigate the generalization error caused by policy distribution shift, we introduce the **policy conditioning**, building upon the concept of policy-conditioned model learning (Chen et al., 2024a), aims to embed the policy information into the dynamics model learning, allowing the model to adapt to different policies actively to mitigate the extrapolation error caused by distribution shift. Furthermore, we propose a simple yet effective technique called **retracing rollout**, to reduce the long-horizon compounding error during test time. This approach fixes the first frame of the moving contexts to be the initial real observation and relabels the corresponding action by retracing the effects of the original actions at the history timesteps. As a plug-and-play solution, retracing rollout can be efficiently applied to end-effector pose control in various embodiment tasks without necessitating any changes to the training process.

Incorporating these two techniques, we present Whale, a scalable embodied world model based on the spatial-temporal transformer (Ma et al., 2024; Bruce et al., 2024), designed to enable faithful long-horizon imagination for real-world visual control tasks. To substantiate the effectiveness of Whale, we conduct extensive experiments on both simulated Meta-World (Yu et al., 2019) benchmark and a physical robot platform, encompassing a variety of pixel-based manipulation tasks. Experimental results on the simulated tasks show that Whale outperforms existing world model learning methods in both video fidelity and value estimation accuracy. Moreover, we also validate the effectiveness of policy-conditioning and retracing-rollout techniques in reducing the generalization error. As a further step, we introduce Whale-X, a **414M parameter** world model trained on **970k** real-world demonstrations from Open X-Embodiment datasets (Collaboration et al., 2023). Whale-X serves as a foundational embodied world model for evaluating real-world behaviors. With fine-tuning on a few demonstrations in completely unseen environments and robots, Whale-X demonstrates strong OOD generalizability across visual, motion, and task perspectives. Furthermore, by scaling up the pre-training dataset or model parameters, Whale-X shows impressive scalability during both the pre-training and fine-tuning phases.

The primary contributions of this work are outlined as follows:

- We introduce two key techniques: **policy conditioning** and **retracing rollout**, to tackle two main challenges of world model generalization: *policy distribution shift* and *long-horizon error compounding*;
- By incorporating these two techniques, we propose **Whale**, a scalable embodied world model with enhanced generalizability, and further present a 414M parameter **Whale-X** pre-trained on 970K robot demonstrations;
- We conduct extensive experiments to showcase the effectiveness of two techniques while highlighting Whale's remarkable scalability and generalization across both simulated and real-world tasks.

2 Backgrounds

2.1 Sequential Decision-Making

A typical formulation of sequential decision tasks is the Markov decision process (MDP) (Puterman, 1990) specified by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, T^*, \gamma, H, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r(s,a)$ is the reward function, $T^*(s'|s,a)$ is the real transition probability, $\gamma \in (0,1]$ is the discount factor, H is the decision horizon, and $\rho_0(s)$ is the initial state distribution. In this work, we simply consider the case where $\gamma = 1$ and $H < \infty$. In reinforcement learning (Sutton & Barto, 2018), the objective is to learn a policy that maximizes the expected return in the MDP, which involves estimating the value of different policies. Specifically, the value of policy π is defined as:

$$V^{\pi}_{T^*} = \mathbb{E}_{\tau_H \sim (\pi, T^*)}\Big[\sum_{t=1}^{H} r(s_t, a_t)\Big], \tag{1}$$

where the state-action trajectory $\tau_H = (s_1, a_1, \ldots, s_H, a_H)$ and rewards are generated by the rollouts of policy π within the dynamics T^*. Therefore, an unbiased estimation of policy values requires online interactions with the real environment.

A common scenario involves abundant pre-collected experience data, but direct interaction with the environment is either prohibited or costly, necessitating value estimation and optimization to be performed offline. In this scenario, an environment model T can be explicitly learned from the offline data and used to generate a simulated experience for value estimation and optimization. Assume that V^{π}_{T} is the value estimated within the model T, the environment model error induces a value gap $|V^{\pi}_{T^*} - V^{\pi}_{T}|$ for the policy π. If the model is globally accurate, the value gap will diminish for any policy. However, offline experiences are often collected by a narrow range of policies (e.g., near-expert policies), and the learned environment models have to generalize beyond the training experiences to evaluate diverse policies.

2.2 World Models for Visual Control

Real-world control tasks often involve high-dimensional visual observations and a partially observable nature. These visual control environments can be further described by a partially observable Markov decision process (POMDP) (Åström, 1965) specified by tuple $(\mathcal{S}, \mathcal{O}, \phi, \mathcal{A}, r, T^*, \gamma, H, \rho_0)$, where the agent receives visual observation $o_t = \phi(s_t)$ at each step, only containing incomplete information of s_t, and executes an action based on history observations $a_t \sim \pi(\cdot|o_{1:t})$. The environment then transitions into the next state s_{t+1} according to $T^*(\cdot|s_t, a_t)$ and provides the agent the next observation $o_{t+1} = \phi(s_{t+1})$ and a reward signal $r(s_t, a_t)$. The agent must predict future outcomes and make decisions based on historical observations due to incomplete information, making the learning of general environment models in visual domains a significant challenge.

World models (Ha & Schmidhuber, 2018) are proposed as a general framework of learning visual dynamics. A vision module learns an abstract, compressed representation of high-dimensional image observations $z_t = E_\theta(o_t)$, a memory model tries to predict the future representations based on the history $P_\theta(z_{t+1}|z_{1:t}, a_{1:t})$, compressing what happens over time, and a decoder recovers the observation and reward predictions from the predicted representation $\hat{o}_{t+1}, \hat{r}_{t+1} = D_\theta(z_{t+1})$. The combination of vision and memory modules enables efficient autoregressive future predictions, allowing agents to plan or learn policies through model imaginations for visual control. Advanced approaches (Hafner et al., 2020; 2023; Babaeizadeh et al., 2021; Gupta et al., 2023; Wu et al., 2024) largely retain this architecture, but replace the encoder, decoder, and latent dynamics with different model architectures (e.g. transformer with video tokenizer and detokenizer). However, these works have not emphasized the generalizability of world models, which is crucial for sequential decision-making but has not been well addressed by previous approaches (Schubert et al., 2023).

3 Scalable World Model with Enhanced Generalizability

The common learning methods for autoregressive world models regard the transition learning as a standard supervised learning problem, minimizing the negative log-likelihood (NLL) of the single-step transition probabilities over the pre-collected trajectories in a teacher-forcing manner, i.e.,

$$\min_T \mathbb{E}_{\mu\sim\Pi}\mathbb{E}_{\tau_H\sim(\mu,T^*)}\frac{1}{H}\sum_{h=1}^{H} -\log T(o_h|\tau_{h-1}) \iff \min_T l_{\mathrm{KL}}(T;\Pi),$$

where (sub-)trajectory $\tau_h = (o_1, a_1, o_2, \ldots, o_h, a_h), 1 \le h \le H$ is generated by interaction of a behavior policy μ with the real dynamics T^*, and behavior μ is assumed to be sampled from a behavior policy distribution Π. Minimizing the NLL equals to minimizing the KL divergence loss $l_{\mathrm{KL}}(T;\Pi) = \mathbb{E}_{\mu\sim\Pi}\mathbb{E}_{\tau_H\sim(\mu,T^*)}\frac{1}{H}\sum_{h=1}^{H} D_{\mathrm{KL}}(T^*(\cdot|\tau_{h-1}), T(\cdot|\tau_{h-1}))$. The learned world models are usually utilized to evaluate any target policy π by simulating trajectories in an autoregressive manner:

$$V_T^\pi = \mathbb{E}_{\tau_H\sim(\pi,T)}\Big[\sum_{t=1}^{H} r(o_t, a_t)\Big],$$

where the trajectory simulation distribution deviates from the training distribution. In classical sequential modeling tasks like sentence generation and translation, the distribution shift from teacher-forcing training to autoregressive generation diminishes as the model accuracy improves, which therefore does not lead to significant negative impacts. In the world model learning, however, the distribution shift results from both the model inaccuracy and the policy divergence, exacerbating the evaluation inaccuracy:

$$\left|V_T^\pi - V_{T^*}^\pi\right| \le 2R_{\max}\underbrace{H^2}_{\text{AutoReg}}\Big(\underbrace{\sqrt{2\,l_{\mathrm{KL}}(T;\Pi)}}_{\text{Train Error}} + \underbrace{L\cdot W_1(d^\pi, d^\Pi)}_{\text{Policy Divergence}}\Big), \tag{2}$$

where a distribution shift term induced by the policy divergence [1] occurs in addition to the KL training loss, further amplified by an H^2 factor caused by the autoregressive generation. Even if the world

[1]Here $W_1(d^\pi, d^\Pi)$ is the Wasserstein-1 distance between the π-induced trajectory distribution $d^\pi(\tau)$ and the behavior trajectory distribution $d^\Pi(\tau) = \mathbb{E}_{\mu\sim\Pi}[d^\mu(\tau)]$, and L is the Lipschitz constant of model loss w.r.t. the trajectory, adapted from Chen et al. (2024a).

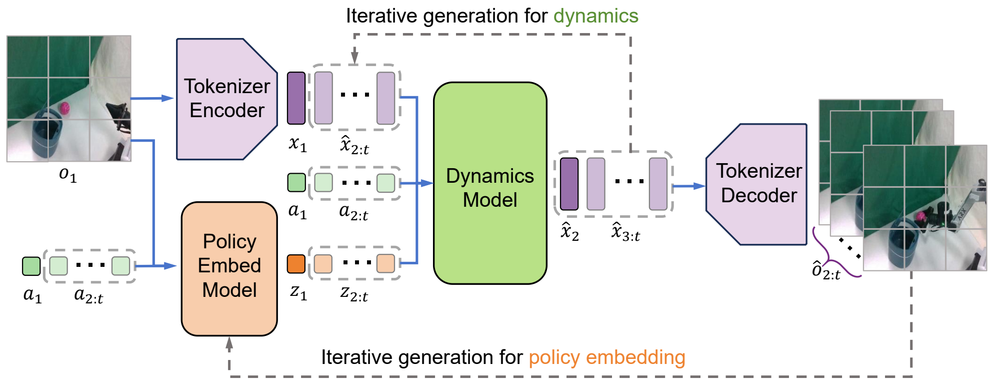


Figure 1: The overall architecture of Whale. The policy embedding model encodes the observation and action subsequences into policy embeddings z_i, which are then passed to the dynamics model along with observation tokens and actions to generate the next token predictions $\hat{x}_{i+1}$. The predicted observation tokens are subsequently fed into the dynamics model for further predictions autoregressively and decoded into observation predictions to obtain later policy embeddings.

model perfectly fits the training transitions, i.e. $l_{\mathrm{KL}}(T;\Pi) = 0$, the variation of the policies could also significantly shift the trajectory distribution to those large error areas, resulting in degenerative generalizability.

One possible solution to this policy generalization issue is to embed the policy information into the world model, allowing the model to actively recognize and adapt to the policy-induced distribution shift (Chen et al., 2024a). This adaptation effect has been shown to reduce model generalization error caused by policy divergence, i.e. the last term in Eq (2). For further analysis, please refer to Appendix A. Furthermore, we devise a simple trick to facilitate long-horizon model rollout for embodiment tasks, effectively alleviating the autoregressive error amplification. Building on these concepts, we propose Whale, a scalable embodied world model with enhanced generalizability.

3.1 Overall Model Architecture

In Figure 1, we illustrate the architecture of the Whale. Specifically, Whale comprises three main components: policy embedding model, video tokenizer, and dynamics model. Inspired by previous works (Bruce et al., 2024), these modules utilize a spatial-temporal transformer (ST-transformer) architecture. Within this framework, each token is designed to attend only to other tokens in the current frame and those at corresponding positions in prior frames. Additionally, Whale is capable of generating all tokens for the next frame in parallel at one time. These designs significantly simplify the computational demands from a quadratic to a linear dependency relative to sequence length, reducing both the memory usage and computational costs of the model training while increasing model inference speed.

3.2 Policy Embedding Learning

We would like to extract the decision patterns within training trajectories τ_H into a policy embedding, reminiscent of the maximization of the evidence lower bound (ELBO) of the trajectory likelihood conditioned on the history τ_h (Venkatraman et al., 2024; Yang et al., 2023; Ajay et al., 2021):

$$\log P(\tau_H|\tau_h) \geq \mathbb{E}_{q_\phi(z|\tau_H)} \sum_{t=h}^{H} \log \pi_w(a_t|o_t, \tau_{t-1}, z) - D_{\mathrm{KL}}(q_\phi(z|\tau_H)||p_\psi(z|\tau_h)) + Const, \quad (3)$$

where $q_\phi(z|\tau_H)$ denotes the posterior encoder, encoding the whole trajectory τ_H into a latent variable z; $\pi_w(a_h|o_h, \tau_{h-1}, z)$ denotes the decoder, which recovers the decision action from the latent variable z and the up-to-date history (τ_{h-1}, o_h); $p_\psi(z|\tau_h)$ denotes the prior predictor, which allows the prediction of z based on the history τ_h. The information bottleneck requires the learned variable z to effectively capture the decision pattern within the trajectory, embedding the information about the corresponding behavior policy. Following this argument, we propose to learn the policy embedding by maximizing the ELBOs over H decision steps and adjusting the amount of KL constraints similar

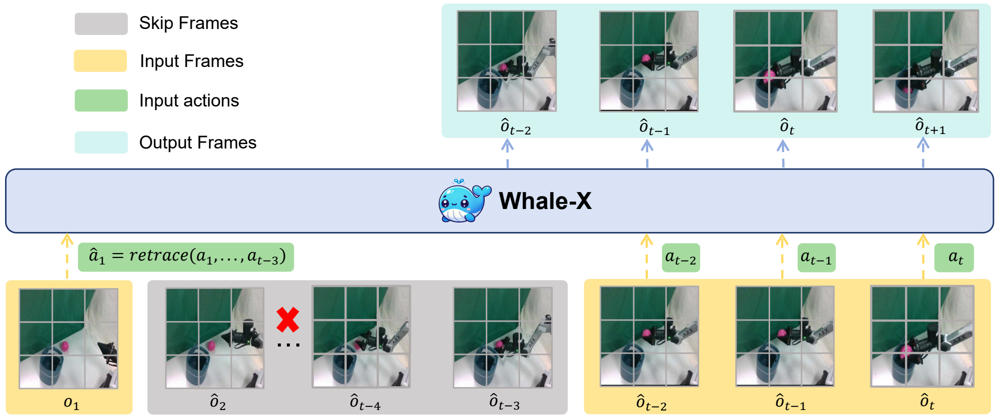


Figure 2: The illustration of the retracing rollout. Here, the retrace action $\hat{a}_1$ can produce an equal effect of the robot's arm as executing $a_1, \cdots, a_{t-3}$ sequentially from o_1, thus effectively reducing the compounding error of the robot's arm generated by world models.

to β-VAE (Higgins et al., 2017):

$$\mathcal{L}(w, \phi, \psi) = \mathbb{E}_{\tau_H \sim \mathcal{D}}\Big[\mathbb{E}_{q_\phi(z|\tau_H)}\big[-\sum_{h=1}^{H} \log \pi_w(a_h|o_h, \tau_{h-1}, z)\big] + \beta \sum_{h=1}^{H} D_{\mathrm{KL}}(q_\phi(z|\tau_H)||p_\psi(z|\tau_h))\Big], \tag{4}$$

here the KL terms constrain the embedding predictions from sub-trajectories up to each time step h, encouraging them to approximate the posterior encoding. This ensures that the representation remains policy-consistent, meaning that trajectories generated by the same policy should have similar representations, as suggested in the previous analysis.

3.3 World Model Learning

World models typically consist of an observation encoder that encodes the raw observation into a compact representation and a dynamics model that predicts future transitions within this representation space (Ha & Schmidhuber, 2018). In this work, we adopt a tokenizer based on VQ-VAE (Van Den Oord et al., 2017) as the encoder to discretize observations into tokens and train a dynamics model at the token level.

Specifically, the video tokenizer e_θ is composed of an encoder E_θ and a decoder D_θ, where the encoder E_θ compresses video input into a sequence of tokens, while the decoder D_θ is capable of reconstructing the original video from these tokens. This tokenizer is trained with the standard VQ-VAE loss $\mathcal{L}_{\text{tok}}(\theta)$, which is a combination of a L_1 reconstruction loss, a codebook loss, and a commitment loss.

After training the tokenizer, we embed the policy information into the dynamics model learning process. The key distinction from standard dynamics model learning is that Whale additionally incorporates a policy embedding z_h inferred by the prior predictor p_ψ. In this phase, for each input trajectory segment τ_H, the video tokenizer first converts it into a sequence of tokens $x_H = ((x_1^{(1)}, \cdots, x_1^{(N)}), (x_2^{(1)}, \cdots, x_2^{(N)}), \cdots, (x_H^{(1)}, \cdots, x_H^{(N)}))$, where $x_i^{(j)}$ represents the j-th token of the i-th frame. Consequently, the training objective of the dynamics model is to maximize the log-likelihood of the tokens x_{h+1} for the next frame s_{h+1}, conditioned on the history tokens $x_{0:h}$, history actions $a_{0:h}$ and the policy embedding $z_h = p_\psi(\tau_h)$:

$$\mathcal{L}_{\text{dyn}}(\theta) = \mathbb{E}_{\tau_H \sim \mathcal{D}}\big[-\sum_{h=1}^{H} \log P_\theta(x_{h+1}|x_{1:h}, a_{1:h}, z_h)\big], \tag{5}$$

Intuitively speaking, Whale does not only accept history as a direct feature to predict transitions but also infers the latent decision intention from the history to enable test-time adaptation to the induced distribution shift.

3.4 Retracing Rollout for Compounding Error Reduction

Model imagination involves rolling out a policy or executing an action sequence step-by-step within the world model. As highlighted in Eq (2), this process suffers from error compounding during test

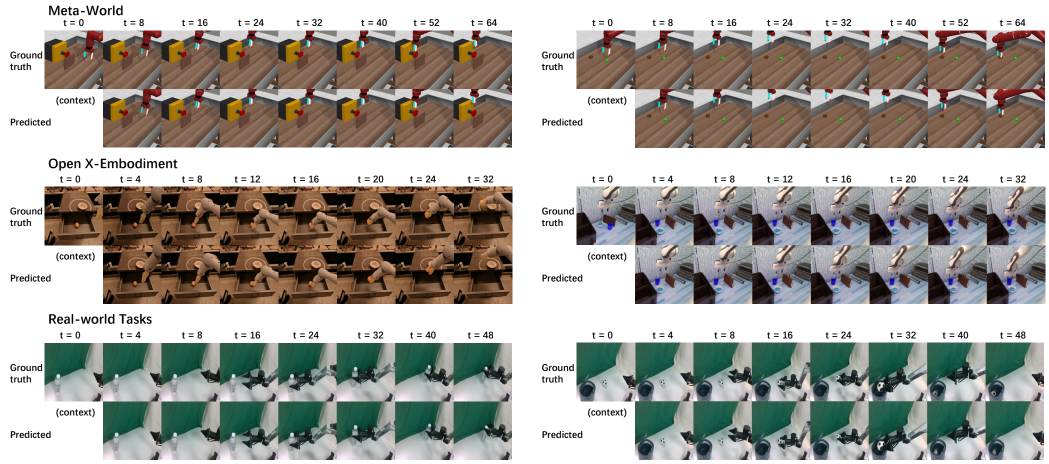


Figure 3: Qualitative evaluation: long-horizon video generation results of Whale on Meta-World, Open X-Embodiment, and our Real-world tasks.

time due to the shift from teacher-forcing training objective to autoregressive generation, resulting in a quadratic increase in model error as the decision horizon H extends. To mitigate this issue, we propose a simple but effective technique, termed **retracing rollout**, as depicted in Figure 2. Specifically, if the model imagination begins from a real initial observation o_1 with a context length assumed to be 4, the context for standard autoregressive rollout to predict $\hat{o}_{t+1}$ at timestep t is $(\hat{o}_{t-3}, a_{t-3}, \hat{o}_{t-2}, a_{t-2}, \hat{o}_{t-1}, a_{t-1}, \hat{o}_t, a_t)$. Nevertheless, if the decision horizon significantly exceeds the context length, the prediction context for later observations will consist entirely of model-generated images and actions. This causes early prediction errors to accumulate, leading to increasingly inaccurate subsequent predictions, a phenomenon commonly referred to as *compounding error*. To mitigate this issue, our retracing rollout instead use the context $(o_1, \hat{a}_1, \hat{o}_{t-2}, a_{t-2}, \hat{o}_{t-1}, a_{t-1}, \hat{o}_t, a_t)$, which fixes the first frame to be the initial real observation o_1, and relabel the corresponding action $\hat{a}_1$ to produces an equivalent effect on the robot's end-effector as executing the skipped actions $a_1, \ldots, a_{t-3}$ sequentially, starting from o_1.

Benefiting from the semantic structure of the action space in embodied control, the action-retracing operation is computationally feasible for end-effector pose control. For instance, in the Open X-embodied dataset, the action space is defined by a 7-dimensional vector that controls the end-effector. The first three dimensions represent the changes in the gripper position (Δx, Δy, Δz), the next three represent the changes in wrist orientation (Δroll, Δpitch, Δyaw), and the final dimension determines whether the gripper opens or closes. Therefore, the retrace action can be directly computed using Eq (6), where $a_i^{(j)}$ represents the value of the j-th dimension of the action a_i.

$$retrace(a_1, \cdots, a_t) = \left(\sum_{i=1}^{t} a_i^{(0)}, \sum_{i=1}^{t} a_i^{(1)}, \sum_{i=1}^{t} a_i^{(2)}, \sum_{i=1}^{t} a_i^{(3)}, \sum_{i=1}^{t} a_i^{(4)}, \sum_{i=1}^{t} a_i^{(5)}, a_t^{(6)} \right). \quad (6)$$

The retracing rollout intuitively offers two key advantages. First, in long-horizon rollouts, it acts as a "fast track" connecting the initial observation to the predicted target, reducing the prediction error in the pose of the robot's arm. Second, by consistently incorporating the real initial observation o_1 into the model's context, the retracing rollout significantly improves the coherence and consistency of the generated trajectories. Notably, the retracing rollout operates without any modifications to the training process, making it a plug-and-play solution, offering both flexibility and ease of implementation.

4 Experiment

We conduct extensive experiments on both simulated tasks and real-world tasks. The experimental design is primarily designed to answer the following key questions:

- How does Whale perform compared with other baselines on simulated tasks? Are policy-conditioning and retracing-rollout techniques effective? (Section 4.1)

- How does Whale perform on real-world tasks? Can Whale benefit from pre-training on internet-scale data? (Section 4.2)
- How is Whale's scalability? Does increasing the model capacity or pre-training data improve performance on real-world tasks? (Section 4.3)

4.1 SIMULATION EVALUATION

Experiment Setups We conduct our simulated task experiments on the Meta-World (Yu et al., 2019) benchmark, which offers a diverse set of vision-based manipulation tasks. In this experiment, we construct a training dataset with 60k trajectories collected from 20 tasks. The model learning algorithms are required to use all the data for training from scratch. During evaluation, given an initial observation and a sequence of actions, the world model should reconstruct the corresponding video trajectories. More detailed information about data collection can be found in Appendix D.1.

Baselines We compare Whale against several world model learning baselines, including (1) **FitVid** (Babaeizadeh et al., 2021), a variational-based world model that can fit large diverse video datasets. (2) **MCVD** (Voleti et al., 2022), a diffusion-based world model that can perform video generation conditioning on different subsets of video frames and actions. (3) **DreamerV3** (Hafner et al., 2023), a recurrent world model that outperforms specialized methods across diverse control tasks. (4) **iVideoGPT** (Wu et al., 2024), a scalable transformer-based world model that achieved state-of-the-art results in video generation and embodied control tasks. Complete descriptions and implementation details are provided in Appendix B.2.

Evaluation Metrics The evaluation scenarios are divided into two categories: *seen policies* and *unseen policies*. Specifically, *seen policies* involve tasks and action sequences that both appear in the training set, *unseen policies* refer to tasks from the training set with action sequences generated by unseen policies. Moreover, we assess the performance of world models from two perspectives: 1) *Video fidelity*. Measures the quality of video trajectory generation, in terms of Fréchet Video Distance (FVD) (Unterthiner et al., 2018), Peak Signal-to-noise Ratio (PSNR) (Huynh-Thu & Ghanbari, 2008), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), and Structural Similarity Index Measure (SSIM) (Wang et al., 2004). 2) *Value estimation accuracy*. Verifies whether the model can correctly estimate the value of a given action sequence, in terms of Value Gap.

Comparison Results Table 1 presents the results for video fidelity and value estimation in the unseen policies setting. Our analysis shows that Whale outperforms all other methods across every metric related to video fidelity, with a notable advantage in FVD. [2] Furthermore, the value estimation results demonstrate that Whale consistently matches or surpasses the baselines in minimizing the value gap for both seen and unseen policies, emphasizing its superior accuracy in value estimation. The remaining evaluation and visualization results can be found in Appendix C and F.1.

Meta-World	#Params	FVD↓	PSNR↑	SSIM↑	LPIPS↓	Value Gap ↓
			unseen policies & 64×64 resolution			
FitVid	143M	154.6	23.7	90.3	6.5	11.1
MCVD	53M	272.8	29.7	92.3	4.0	15.9
DreamerV3	44M	142.7	27.6	92.1	4.3	**5.3**
iVideoGPT	63M	115.7	28.5	92.8	4.5	6.4
Whale (ours)	51M	**33.0±1.4**	**29.8±0.0**	**94.4±0.0**	**3.2±0.0**	**5.6±0.3**
			unseen policies & 256×256 resolution			
DreamerV3	61M	112.4	26.2	91.7	8.5	7.5
Whale (ours)	63M	**28.2±3.6**	**29.2±0.2**	**95.0±0.1**	**4.3±0.1**	**5.0±0.2**

Table 1: Performance comparison on Meta-World benchmark with various models.

[2]In the 64×64 resolution, retracing rollout was omitted due to inconsistencies in object appearance within the real videos, offering no added benefit in this context. However, at higher resolutions, retracing rollout led to a marked improvement, as demonstrated in Table 2 and Table 3.

Ablation Study To validate the effectiveness of policy-conditioned and retracing rollout techniques, we conduct comprehensive ablation experiments under the unseen policies setting at a resolution of 256x256, as presented in Table 4 and Figure 4. The results show that the policy-conditioned method effectively identifies and represents test policies, generating more realistic video trajectories and reducing the value gap by **37%**. Additionally, we find that retracing rollout provides significant improvements: without altering the training process, the FVD of trajectories generated by retracing rollout is only **33%** of that produced by standard autoregressive rollout, while the value gap is reduced to **50%** of the original. These findings demonstrate that the policy-conditioning and retracing-rollout mechanisms significantly enhance the generalizability of world models. For more results, please refer to Appendix C.

Meta-World	FVD↓	PSNR↑	SSIM↑	LPIPS↓	Value Gap ↓
unseen policies & 256×256 resolution					
Whale (w/o retracing-rollout)	84.2±5.7	24.3±0.3	92.0±0.2	6.9±0.3	10.0±0.6
Whale (w/o policy-conditioning)	32.0±0.4	28.9±0.2	94.6± 0.1	4.6±0.1	7.9±0.2
Whale (ours)	**28.2±3.6**	**29.2±0.2**	**95.0±0.1**	**4.3±0.1**	**5.0±0.2**

Table 2: Ablation Study of Whale on Meta-World benchmark.

4.2 Physical Robot Evaluation

Pre-training. We present Whale-X, a 414M parameter world model pre-trained on 970K real-world robot demonstrations from Open X-Embodiment datasets. We use the entire dataset to pre-train both the policy embedding model and the video tokenizer, selectively using a subset of the data to pre-train the dynamics model. Whale-X serves as a foundational embodied world model for evaluating real-world behaviors, capable of generating realistic and controllable video trajectories that align with the given actions, as shown in Figure 3. Additional details on the pre-training process and the generated results can be found in Appendix D.2 and Appendix F.3, respectively.

Experiment Setups. To evaluate the out-of-distribution generalizablitiy of Whale-X in the physical world, we conduct comprehensive real-world experiments on ARX5 robotic platform. The evaluation tasks differ significantly from the pre-training data, in terms of the robotic platform, camera angles, and background visual information, posing considerable challenges for world models.

We carefully collect a limited dataset for fine-tuning, consisting of 60 trajectories for each of the four tasks: *open bin*, *push plate*, *throw ball*, and *move bottle*. Following this, we designed several challenging unseen tasks for testing, with a focus on evaluating the model from the perspectives of *visual generalization*, *motion generalization*, and *task generalization* perspectives. Further details on the data collection process can be found in Appendix E.

Evaluation Metrics. Given an initial frame and a sequence of subsequent actions, world models should autoregressively generate future video trajectories. For a visual world model to be effective in decision-making, it needs to focus more on reasoning about the consequences of actions than on reconstructing irrelevant visual information like backgrounds. Thus we introduce the *consistency rate* to assess whether the differences in reconstructed object positions, interactive object states, and robot arm positions fall within an acceptable range compared to the ground truth. We use the multimodal large model GPT-4o (Achiam et al., 2023) for this evaluation through multiple rounds of Q&A. Details of the prompts and the evaluation process can be found in the Appendix H, with results presented in Figure 4. In addition, we employ several video fidelity metrics, similar to those in Section 4.1, to assess the quality of video generation by the world models.

Task Results Whale-X model shows a clear advantage in our real-world experiments. Specifically, as shown in Figure 4, the quantitative results indicate that: 1) Whale-X improves consistency by **63%** and **30%** compared to models without policy-conditioning and retracing-rollout respectively, demonstrating that these mechanisms significantly enhance the OOD generalizability; and 2) Whale-X, pre-trained on 970k samples, achieved much higher consistency rate than models trained from scratch, highlighting the benefits of pre-training on large-scale internet data. Furthermore, the evaluation of video generation quality aligns with these consistency rate findings as illustrated in

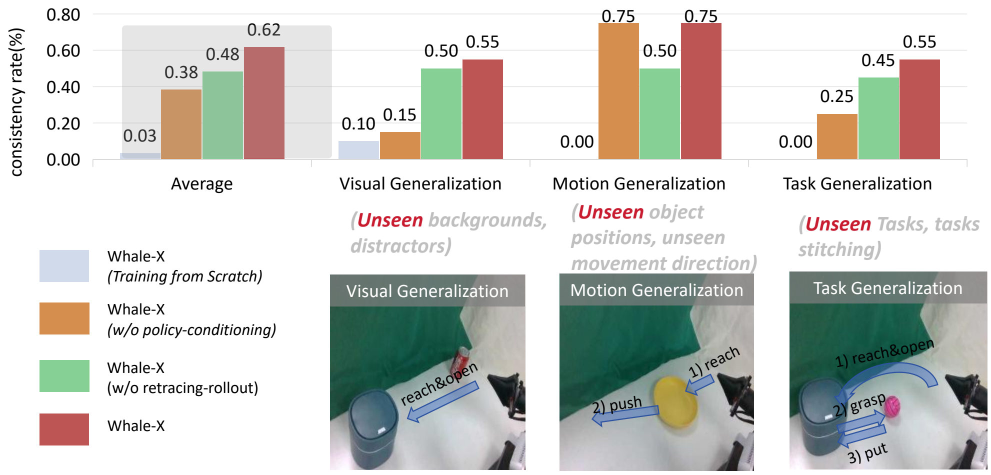


Figure 4: The results of physical robot evaluation on unseen scenarios. The row above shows the bar chart of the consistency rate, and the row below represents the tasks used for testing. The experiments demonstrate that Whale-X exhibits good generalization performance in unseen scenarios, and both the proposed policy-conditioning and retracing-rollout can enhance the model's performance.

Real-world Tasks	PSNR↑	SSIM↑	LPIPS↓
unseen tasks & 256×256 resolution			
Whale-X (training from scratch)	20.0	74.9	37.0
Whale-X (w/o retracing-rollout)	21.9	79.6	30.3
Whale-X (w/o policy-conditioning)	21.4	79.0	31.2
Whale-X (ours)	**22.3**	**80.5**	**29.6**

Table 3: Video Fidelity of Whale-X on real-world tasks.

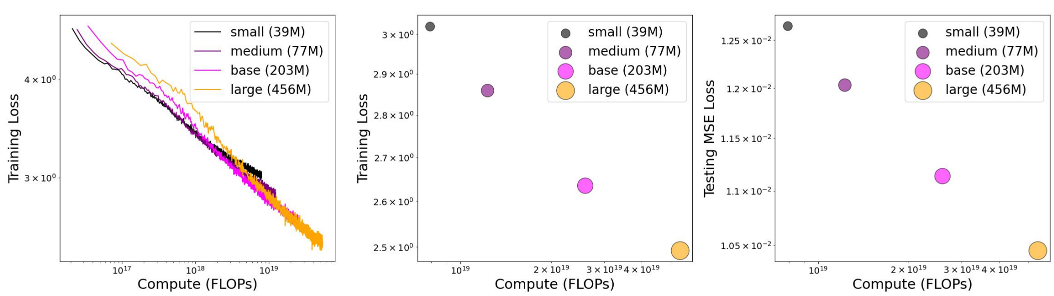


Figure 5: Scaling Experiment Results of Whale-X. The leftmost plot shows the training loss curves for models with varying parameter sizes during the pre-training phase. The second plot presents the final training loss for all models after 300k pre-training steps. The third plot displays the test loss after fine-tuning. The legend in the figure indicates the parameter number of the dynamics model.

Table 3: both policy-conditioning and retracing-rollout techniques boost OOD generalizability and significantly outperform models lacking pre-training.

4.3 Scaling Experiments

In this section, we aim to investigate the scaling behavior of Whale-X. Specifically, We freeze the video tokenizer and policy embedding model, adjusting only the model size and pre-training data size of dynamics models, considering the impact of model size and data size for the pre-training and fine-tuning phases.

Pre-training Scaling Experiments. With a freezed video tokenizer and policy embedding model, we train four dynamics models ranging from 39M to 456M parameters during the pre-training phase, with results shown in the first two plots of Figure 5. These results demonstrate that Whale-X exhibits strong scalability, as increasing either the pre-training data or the number of parameters reduces the training loss. Notably, the training loss of Whale-X follows a log-linear relationship with FLOPs, which can guide the design of larger models and appropriate data ratios for future experiments.

Fine-tuning Scaling Experiments. Apart from the scalability in the pre-training stage, it is also worth verifying whether a larger model can exhibit better performance during the fine-tuning phase. To this end, we fine-tune a series of dynamics models and show the test mean-squared-error losses in the leftmost plot in Figure 5. The results indicate that after fine-tuning, the larger model demonstrates lower loss on test data, highlighting promising scalability of Whale-X for real-world tasks.

5 Related Works

Learning accurate dynamics models has been a long-standing challenge in sequential decision-making. Many works focused primarily on learning transition models in lower-dimensional proprioceptive state spaces from the perspective of model architecture (Chua et al., 2018; Zhang et al., 2021; Janner et al., 2021; Chen et al., 2024b) or learning objective (Xu et al., 2020; Chen et al., 2023; Luo et al., 2024a; Lin et al., 2024). The environment model learning provides benefits for downstream tasks, especially model-based reinforcement learning (Janner et al., 2019; Yu et al., 2020; 2021; Sun et al., 2023). Recent research interest has shifted towards learning environment models for high-dimensional image-based tasks (Hafner et al., 2020; Babaeizadeh et al., 2021; Yang et al., 2024), commonly referred to as world models (Ha & Schmidhuber, 2018).

Some recent model-based RL algorithms leverage latent imagination for more efficient and accurate rollouts (Hafner et al., 2020; 2021; 2023; Hansen et al., 2022; Schrittwieser et al., 2020), but they become more complex by tightly coupling model and policy learning. Advanced methods leverage modern action-conditioned video prediction models (Oh et al., 2015; Kaiser et al., 2020) to model the visual dynamics and pre-train from large-scale video experience data (Mendonca et al., 2023b; Wu et al., 2023). Various models have been adopted in these methods, including RNNs (Villegas et al., 2019; Hafner et al., 2020; Babaeizadeh et al., 2021), diffusion models (Voleti et al., 2022), and transformers (Gupta et al., 2023; Wu et al., 2024). These interactive models generate videos under the control of the executed actions, with the goal of capturing real visual dynamics for various decision strategies. However, these works have not emphasized the generalizability of world models, which is crucial for sequential decision-making but has not been well addressed by previous approaches (Schubert et al., 2023). In contrast, our work focuses on the world model generalizability from a perspective of evaluation accuracy and utilizing model adaptation to policies and retracing rollout to mitigate the generalization error in scalable world models.

6 Discussions and Limitations

We introduce Whale, a scalable and generalizable embodied world model that incorporates the policy-conditioning mechanism and retracing-rollout technique to enhance out-of-distribution generalization, and pre-train a 414M-parameter Whale-X on large-scale real-world robot data to assist physical robot manipulation. As a powerful world model with strong generalizability and promising scalability, Whale enables high-fidelity imagination and accurate value estimation, even in novel scenarios, thereby facilitating downstream control tasks.

Limitations and future work. Although Whale-X marks significant progress, there remains substantial room for further improvement in future work. One limitation is the lack of diversity in real-world robotic data, typically collected by a narrow range of policies (e.g. near-optimal policies). This poses significant challenges to the generalization of world models. Additionally, we found that the quality of reward models with visual input plays a crucial role in accurate value estimation, which remains an unsolved challenge for future research. Lastly, we have to mention that although Whale's generalization capability has significantly improved compared with previous methods, it remains limited for zero-shot transfer in the face of the diversity and complexity of unseen real-world tasks. Integrating existing prior knowledge into the data-driven world model learning process could enable broader generalization, presenting a valuable avenue for long-term research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: offline primitive discovery for accelerating offline reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. *arXiv preprint arXiv:2402.15391*, 2024.

Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. `https://sites.google.com/view/berkeley-ur5/home`.

Ruifeng Chen, Xiong-Hui Chen, Yihao Sun, Siyuan Xiao, Minhui Li, and Yang Yu. Policy-conditioned environment models are more generalizable. In *Forty-first International Conference on Machine Learning*, 2024a. URL `https://openreview.net/forum?id=g9mYBdooPA`.

Ruifeng Chen, Chengxing Jia, Zefang Huang, Tian-Shuo Liu, Xu-Hui Liu, and Yang Yu. Offline transition modeling via contrastive energy learning. In *Forty-first International Conference on Machine Learning*, 2024b.

Xiong-Hui Chen, Yang Yu, Zhengmao Zhu, Zhihua Yu, Chen Zhenjun, Chenghe Wang, Yinan Wu, Rong-Jun Qin, Hongqiu Wu, Ruijin Ding, et al. Adversarial counterfactual environment model learning. *Advances in Neural Information Processing Systems*, 36:70654–70706, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023. URL `https://github.com/clvrai/clvr_jaco_play_dataset`.

Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Ziyu Wang, Alexander Novikov, Mengjiao Yang, Michael R. Zhang, Yutian Chen, Aviral Kumar, Cosmin Paduraru, Sergey Levine, and Tom Le Paine. Benchmarks for deep off-policy evaluation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. In *ICLR*, 2023.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL `http://arxiv.org/abs/1803.10122`.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

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

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pp. 991–1002. PMLR, 2022.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *CoRR*, abs/2406.09246, 2024.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Haoxin Lin, Yu-Yan Xu, Yihao Sun, Zhilong Zhang, Yi-Chen Li, Chengxing Jia, Junyin Ye, Jiaji Zhang, and Yang Yu. Any-step dynamics model improves future predictions for online and offline reinforcement learning. *arXiv preprint arXiv:2405.17031*, 2024.

Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.

Fan-Ming Luo, Tian Xu, Xingchen Cao, and Yang Yu. Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv preprint arXiv:2307.08927*, 2023.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024b.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

Ju Ma, Juan Zhao, and Yao Hou. Spatial-temporal transformer networks for traffic flow forecasting using a pre-trained language model. *Sensors*, 24(17):5502, 2024. URL `https://doi.org/10.3390/s24175502`.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. *CoRR*, abs/1811.02790, 2018. URL `http://arxiv.org/abs/1811.02790`.

Gerrit W Maus, Jason Fischer, and David Whitney. Motion-dependent representation of space in area mt+. *Neuron*, 78(3):554–562, 2013.

Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023a.

Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023b.

Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

Nora Nortmann, Sascha Rekauzke, Selim Onat, Peter König, and Dirk Jancke. Primary visual cortex represents the difference between past and present. *Cerebral Cortex*, 25(6):1427–1440, 2015.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. `https://octo-models.github.io`, 2023.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Joern Vogel. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7, Paris, France, 2020.

Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=WuBv9-IGDUA`.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Ingmar Schubert, Jingwei Zhang, Jake Bruce, Sarah Bechtle, Emilio Parisotto, Martin Riedmiller, Jost Tobias Springenberg, Arunkumar Byravan, Leonard Hasenclever, and Nicolas Heess. A generalist dynamics model for control. *arXiv preprint arXiv:2305.10912*, 2023.

Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. URL `https://openreview.net/forum?id=PwqiqaaEzJ`.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pp. 33177–33194. PMLR, 2023.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. *arXiv preprint arXiv:2304.13723*, 2023.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Jialong Wu, Haoyu Ma, Chaoyi Deng, and Mingsheng Long. Pre-training contextualized world models with in-the-wild videos for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are scalable world models. *arXiv preprint arXiv:2405.15223*, 2024.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33:15737–15749, 2020.

Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens Dataset. August 2023.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024.

Yiqin Yang, Hao Hu, Wenzhe Li, Siyuan Li, Jun Yang, Qianchuan Zhao, and Chongjie Zhang. Flow to control: Offline reinforcement learning with lossless primitive discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10843–10851, 2023.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2019. URL http://proceedings.mlr.press/v100/yu20a.html.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, virtual event, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems 34 (NeurIPS'21)*, virtual event, 2021.

Michael R Zhang, Thomas Paine, Ofir Nachum, Cosmin Paduraru, George Tucker, Mohammad Norouzi, et al. Autoregressive dynamics models for offline policy evaluation and optimization. In *International Conference on Learning Representations*, 2021.

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*, pp. 586–595, 2018.

Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark, 2023.

Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingyu Ding, Wei Zhan, and Masayoshi Tomizuka. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. 2023a.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023b.

A ANALYSIS OF POLICY CONDITIONING

In this section, we provide some theoretical explanations about why policy-conditioning mechanism helps mitigate the generalization error caused by the policy divergence. The analysis is mainly adapted from Chen et al. (2024a).

First, we introduce an assumption on the smoothness of a well-trained dynamics model:

Assumption A.1 *For the learned dynamics model T, the point-wise total-variation model error $D_{\mathrm{TV}}[T^*(\cdot|\tau_h), T(\cdot|\tau_h)]$ is L-Lipschitz with respect to the trajectory inputs, i.e.,*

$$\left| D_{\mathrm{TV}}[T^*(\cdot|\tau_h^1), T(\cdot|\tau_h^1)] - D_{\mathrm{TV}}[T^*(\cdot|\tau_h^2), T(\cdot|\tau_h^2)] \right| \le L \cdot D(\tau_h^1, \tau_h^2),$$

where $D(\cdot, \cdot)$ is some kind of distance defined on the trajectory space.

Assumption A.1 measures the local extrapolation ability of a world model. Based on this assumption, the value gaps of common dynamics model T without a policy-conditioning mechanism can be controlled:

Proposition A.2 *Under Assumption A.1, for any policy π, the value gap of common dynamics model T without policy conditioning has an upper bound:*

$$\left| V_T^\pi - V_{T^*}^\pi \right| \le 2R_{\max} H^2 \Big(\underbrace{\sqrt{2\, l_{\mathrm{KL}}(T; \Pi)}}_{\text{Train Error}} + \underbrace{L \cdot W_1(d^\pi, d^\Pi)}_{\text{Policy Divergence Error}} \Big),$$

where $W_1(d^\pi, d^\Pi)$ is the Wasserstein-1 distance between the π-induced trajectory distribution $d^\pi(\tau)$ and the behavior trajectory distribution $d^\Pi(\tau) = \mathbb{E}_{\mu\sim\Pi}[d^\mu(\tau)]$.

Proposition A.2 shows that the generalization of common dynamics model T solely relies on its point-level smoothness over the trajectory inputs, resulting in an inevitable extrapolation error of the policy distribution. In contrast, a policy-conditioned dynamics model $T(\cdot)$, which yields adapted dynamics model $T(\pi)$ for some policy π, takes a further step to reduce the policy distribution extrapolation error:

Proposition A.3 *Under Assumption A.1, for any policy π, the value gap of policy-conditioned dynamics model $T(\cdot)$ has an upper bound:*

$$\left| V_{T(\pi)}^\pi - V_{T^*}^\pi \right| \le 2R_{\max} H^2 \Big(\underbrace{\sqrt{2\, l_{\mathrm{KL}}(T; \Pi)}}_{\text{Train Error}} + \underbrace{L \cdot W_1(d^\pi, d^\Pi) - C(\pi, \Pi)}_{\text{Reduced Policy Divergence Error}} \Big),$$

where the adaptation gain $C(\pi, \Pi) := \mathbb{E}_{\mu\sim\Pi}\mathbb{E}_{\tau\sim d^\pi} D_{\mathrm{TV}}[T^*, T(\mu)](\tau) - \mathbb{E}_{\tau\sim d^\pi} D_{\mathrm{TV}}[T^*, T(\pi)](\tau)$ summarizes the policy adaptation effect.

Proposition A.3 explains the benefit brought by policy-conditioning: a positive adaptation gain $C(\pi, \Pi)$, which quantifies the advantage of the policy adaptation effect. The key insight is that when testing on an unseen policy π within some effective region, the model $T(\pi)$, customized for π, should exhibit a smaller model error under the target trajectory distribution d^π compared to models $T(\mu)$ trained on behavior policies $\mu \in \Pi$, which mitigates the generalization error caused by the policy extrapolation. Although it is challenging to rigorously analyze the adaptation gain $C(\pi, \Pi)$ due to the complexity of neural networks and the optimization process, qualitative discussions and empirical evidence, as shown in Chen et al. (2024a), justify the underlying rationale.

B IMPLEMENTATION DETAILS

B.1 IMPLEMENTATION DETAILS OF WHALE

Video Tokenizer. Here we show the architecture and training hyperparameter of the video tokenizer as shown in Table 4. We train three different video tokenizers in total, and our model architecture and training parameter selection are based on the design of Bruce et al. (2024).

Component	Parameter	Meta-World$_{(64\times64)}$	Meta-World$_{(256\times256)}$	Whale-X$_{(256\times256)}$
Encoder	num_layers	4	12	12
	d_model	512	512	512
	num_heads	8	8	8
Decoder	num_layers	8	16	20
	d_model	512	512	1024
	num_heads	8	8	16
Codebook	num_codes	1024	1024	2048
	patch_size	4	16	16
	latent_dim	32	32	32
	beta	0.25	0.25	0.25
Optimizer	type	AdamW	AdamW	AdamW
	max_lr	3e-4	3e-4	3e-4
	min_lr	3e-4	3e-4	3e-5
	β_1	0.9	0.9	0.9
	β_2	0.9	0.9	0.9
	weight_decay	1e-4	1e-4	0
	warmup_steps	10k	10k	5k
	batch_size	32	32	64
	training_steps	100k	150k	300k

Table 4: Hyperparameter of video tokenizers.

Policy Embedding Model. The model architecture and training hyperparameters of the policy embedding model are shown in Table 5. We also train three different policy embedding models. We use two-hot encoding for the practical implementation of our policy embedding similar in Hafner et al. (2020). Additionally, We also observe overfitting in the policy embedding model during pre-training, prompting the use of the early-stop technique. As a result, the checkpoint at 50k is selected as the final model for Whale-X.

Dynamics model Table 6 and Table 7 present the hyperparameters of the dynamics model. We train a total of 6 different dynamics models. The architecture design and training hyperparameters of our dynamics model are also referred to Bruce et al. (2024).

B.2 Implementation Details of Baselines

We use the official implementation of VP2 (Tian et al., 2023) for both FitVid and MCVD. For DreamerV3, we retain only the world model learning component. Additionally, we use the official implementation of iVideoGPT as described in their original paper, but with a reduced number of parameters. The detailed hyperparameters for DreamerV3 and iVideoGPT are provided in Table 8 and Table 9, respectively.

C Additional Experiments Results

Benchmark results. The omitted benchmark results on simulated tasks are shown in Table 10. This table presents the evaluation results of trajectories generated by the world model, conditioned on action sequences produced by policies seen in the training dataset.

Ablation studies. The omitted ablation studies results on simulated tasks are shown in Table 11, showcasing both retracing rollout and policy conditioning consistently effective in seen policies setting.

Component	Parameter	Meta-World$_{(64\times64)}$	Meta-World$_{(256\times256)}$	Whale-X$_{(256\times256)}$
Posterior	num_layers	8	8	12
	d_model	512	512	768
	num_heads	8	8	12
	patch_size	8	32	32
Prior	num_layers	4	4	8
	d_model	512	512	512
	num_heads	4	4	8
	patch_size	8	32	32
Policy	num_layers	8	8	12
	d_model	512	512	768
	num_heads	8	8	12
	log_std	[-2, 5]	[-2, 5]	[-2, 5]
	patch_size	8	32	32
Embedding	category_size	16	16	16
	class_size	16	16	16
Optimizer	type	AdamW	AdamW	AdamW
	max_lr	3e-4	3e-4	3e-4
	min_lr	3e-5	3e-5	3e-5
	β_1	0.9	0.9	0.9
	β_2	0.9	0.9	0.9
	weight_decay	1e-4	1e-4	1e-4
	warmup_steps	5k	5k	5k
	batch_size	64	64	64
	training_steps	100k	100k	50k

Table 5: Hyperparameter of policy embedding models.

Model	#Parameters (dynamics only)	num_layers	num_heads	d_model
Whale-Meta64	26M	12	8	512
Whale-Meta256	26M	12	8	512
Whale-X-small	39M	18	8	512
Whale-X-medium	77M	16	16	768
Whale-X-base	204M	24	16	1024
Whale-X-large	456M	24	12	1536

Table 6: Model hyperparameter of dynamics models.

Parameter	Value
max_lr	3e-5
min_lr	3e-6
β_1	0.9
β_2	0.9
weight_decay	0
warmup_steps	5k
batch_size	64
training_steps	300k

Table 7: Trainig hyperparameter of dynamics models.

Hyperparameters	Values
# Parameters	44M
Dynamics hidden	1024
Dynamics deterministic	1024
Dynamics stochastic	32
Dynamics discrete	32
CNN depth	64
CNN kernel size	4
MLP layers	5
MLP units	1024
Actionvation	SiLU
Train batch size	32
Train batch length	8

Table 8: Hyperparameters for DreamerV3.

Hyperparameters	Values
# Parameters	63M
Down blocks	3
Down layers per block	2
Down channels	[64, 128, 256]
Up blocks	3
Up layers per block	3
Up channels	[256, 128, 64]
Embedding dim	64
Codebook size	8192
Actionvation	SiLU
Transformer hidden dim	512
Transformer hidden layers	6
Attention Heads	8
Feedforward dim	1024

Table 9: Hyperparameters for iVideoGPT.

Meta-World	FVD ↓	PSNR↑	SSIM↑	LPIPS↓	Value Gap ↓
seen policies & 64×64 resolution					
FitVid	193.2	23.7	90.3	6.4	9.7
MCVD	271.7	30.1	92.8	3.8	12.2
DreamerV3	145.8	28.3	92.8	4.0	**4.4**
iVideoGPT	122.0	30.4	93.3	4.4	**4.5**
Whale (ours)	**28.4±1.1**	**31.3±0.01**	**95.3±0.03**	**2.9±0.05**	**4.7 ± 0.47**
seen policies & 256×256 resolution					
DreamerV3	105.1	26.8	92.1	8.2	6.2
Whale (ours)	**25.2± 3.0**	**30.1±0.2**	**95.4±0.1**	**4.0±0.1**	**3.9 ± 0.3**

Table 10: Benchmark results in seen policies setting.

D Data Preparation

D.1 Simulated Data

We select a total of 20 tasks from the MetaWorld benchmark. Each task includes a training set of 3,000 trajectories and a test set of 1,500 trajectories. Specifically, for each task, we use six different policies to collect the training set: expert policy, random policy, two suboptimal policies with different levels of Gaussian noise, and two cross-environment policies. Additionally, three unseen policies are used to gather the testing data. The world models are trained on the full training dataset, followed by a thorough evaluation using the testing data.

D.2 Pre-training Data

We pretrain our Whale-X model on the Open X-Embodiment dataset (Collaboration et al., 2023) (OpenX). The full OpenX dataset consists of more than 70 individual robot datasets, with more than 2M robot trajetories, that were pooled into a coherent and easy-to-use data format in a large community effort. We list our used data mixture in Table 12, mostly following OpenVLA (Kim et al., 2024) and Octo (Octo Model Team et al., 2023).

To train a world model focused on tabletop tasks, we extract data related to tabletop tasks from the dataset that features similar camera positions (the bolded tasks in Table 12) to train the dynamics model, while the video tokenizer and policy condition model are trained on the full OpenX dataset.

Meta-World	FVD↓	PSNR↑	SSIM↑	LPIPS↓	Value Gap ↓
		seen policies & 256×256 resolution			
Whale (wo retracing rollout)	63.4±23.6	25.5±0.7	92.8±0.5	6.2±0.6	7.5±1.0
Whale (wo policy-conditioned)	28.4±1.1	29.5±0.2	95.0±0.1	4.5±0.1	4.9±0.1
Whale	**25.2±3.0**	**30.1±0.2**	**95.4±0.1**	**4.0±0.1**	**3.9±0.3**

Table 11: Ablation Study of Whale in seen policies setting.

Whale-X Pre-training Dataset Mixture	**Percentage**
Fractal (Brohan et al., 2022)	12.7%
Kuka (Kalashnikov et al., 2018)	12.7%
Bridge(Ebert et al., 2021; Walke et al., 2023)	13.3%
Taco Play (Rosete-Beas et al., 2022; Mees et al., 2023)	3.0%
Jaco Play (Dass et al., 2023)	0.4%
Berkeley Cable Routing (Luo et al., 2023)	0.2%
Roboturk (Mandlekar et al., 2018)	2.3%
Viola (Zhu et al., 2023b)	0.9%
Berkeley Autolab UR5 (Chen et al.)	1.2%
Toto (Zhou et al., 2023)	2.0%
Language Table (Lynch et al., 2023)	4.4%
Stanford Hydra Dataset (Belkhale et al., 2023)	4.4%
Austin Buds Dataset (Zhu et al., 2022)	0.2%
NYU Franka Play Dataset (Cui et al., 2022)	0.8%
Furniture Bench Dataset (Heo et al., 2023)	2.4%
UCSD Kitchen Dataset (Yan et al., 2023)	<0.1%
Austin Sailor Dataset (Nasiriany et al., 2022)	2.2%
Austin Sirius Dataset (Liu et al., 2023)	1.7%
DLR EDAN Shared Control (Quere et al., 2020)	<0.1%
IAMLab CMU Pickup Insert (Saxena et al., 2023)	0.9%
UTAustin Mutex (Shah et al., 2023)	2.2%
Berkeley Fanuc Manipulation (Zhu et al., 2023a)	0.7%
CMU Stretch (Mendonca et al., 2023a)	0.2%
BC-Z (Jang et al., 2022)	7.5%
FMB Dataset (Luo et al., 2024b)	7.1%
DobbE (Shafiullah et al., 2023)	1.4%
DROID (Khazatsky et al., 2024)	10.0%

Table 12: Whale-X Pre-training Dataset Mixture.

E Real-world Task Design

E.1 Hardware Setup

Our hardware setup is shown in Figure 6. For the embodiment, we use the ARX5 robotic platform, which is similar to Aloha (Fu et al., 2024) and includes two master arms and two puppet arms. Data is collected via teleoperation and we only use the right arm in our experiment. For the vision sensor, a Realsense D435i camera is mounted above the left side of the platform to capture RGB image observations.

E.1.1 Details of Tasks

The training data set used for finetuning consists of 4 tasks: **Move Bottle**, **Open Bin**, **Push Plate**, and **Throw Ball**.

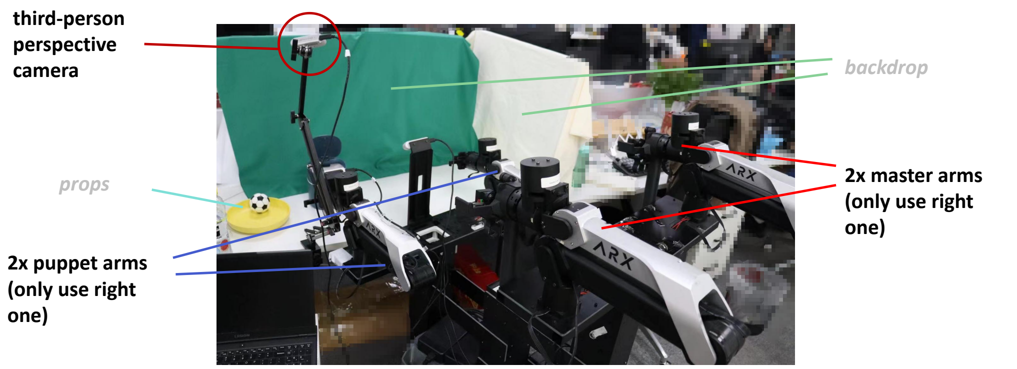
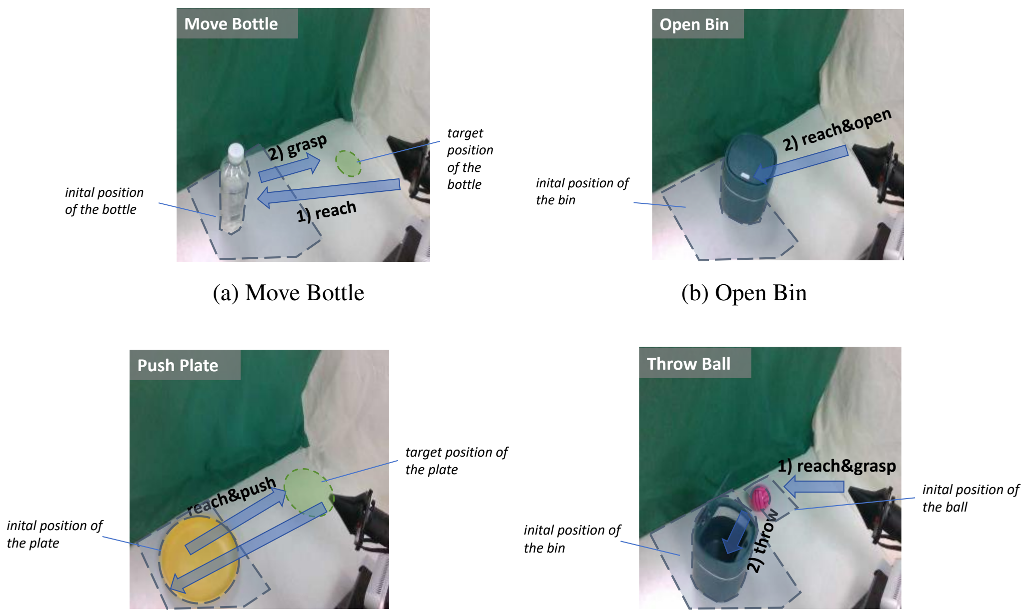


Figure 6: The illustration of our robotics platform used for physical robot evaluation.



(a) Move Bottle

(b) Open Bin

(c) Push Plate

(d) Throw Ball

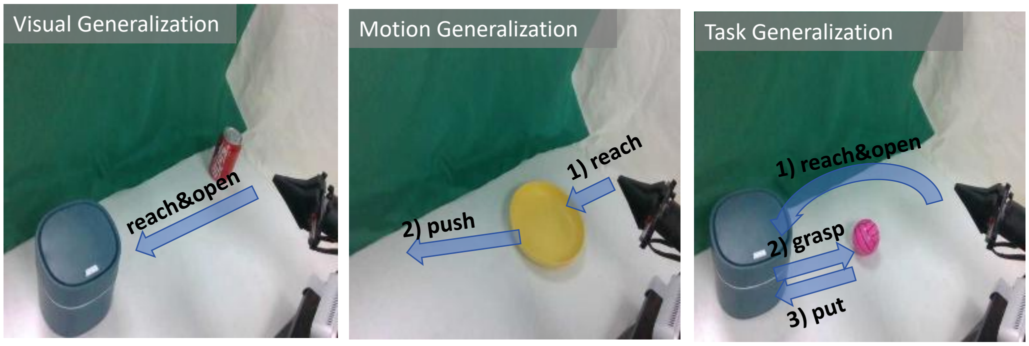
Figure 7: The illustration of Training Tasks.

Move Bottle: The robot arm must first grasp the bottle and then move it to a specific area on the right side of the table. The bottle's initial position is somewhat random, varying within a range of two bottle widths around the location shown in the figure, while its target position remains fixed.

Open Bin: In this task, the robot arm must press a small white area on the trash bin's lid to open it. The initial position of the trash bin has some randomness, and it may vary within a 5 cm range around the position shown in the figure. Additionally, the orientation of the trash bin may have a random variation of about 10 degrees relative to its square alignment.

Push Plate: In this task, the robot arm must push the plate from the left side of the table to the right with appropriate force and angle. The challenge lies in the fact that the plate may rotate or shift during the pushing process. The initial position of the plate has some randomness, varying within a 5 cm range around the position shown in the figure. The robot arm needs to push the plate at a distance of approximately 20 cm.

Throw Ball: In this task, the robot arm needs to make a two-stage decision: 1) move to the ball's location and grasp it; 2) move to the trash bin's opening and release the gripper. The initial position of the trash bin has some randomness, varying within a 5 cm range around the position shown in the figure.



(a) Visual Generalization (b) Motion Generalization (c) Task Generalization

Figure 8: The illustration of Generalization Tasks.

Visual Generalization: In this unseen scenario, we introduced several visual distractors not encountered during the fine-tuning phase, including a soda can, a plate, a ball, and a pencil, based on the Open Bin task. This task is designed to test the robustness of the world model's visual representation and its generalization in visual perception.

Motion Generalization: In this unseen scenario, based on the Push Plate task, we changed the specific task from pushing left to right in the fine-tuning data to pushing right to left. This task is designed to evaluate the model's ability to generalize environment transition modeling when facing an unseen action distribution, or even a completely reversed action distribution.

Task Generalization: In this unseen scenario, we combined two tasks from the fine-tuning phase—Open Bin and Throw Ball—into a new two-stage task. In this task, the robot arm must first open the bin and then place the ball inside. This task is designed to test the model's generalization ability to new tasks, as well as its capability to model long-horizon actions.

E.2 Data Overview

Entry	Value
# Episodes	300(240 for fine-tuning, 60 for testing)
Average horizon	30
Data Collect Method	Human teleoperation using the master arm
Scene Type	Table top
Robot Morphology	Single arm
Camera resolution	640x480
# Cameras	1
Action dimension	7
Action space	EEF position
Action semantics	(Δx, Δy, Δz, Δroll, Δpitch, Δyaw, the gripper state)
Control frequency	5Hz
Has suboptimal?	Yes(some failure data for fine-tuning)
Has camera calibration?	No

Table 13: The meta Information of data used in physical robot evaluation.

F Qualitative Evaluation

F.1 Qualitative Evaluation on Simulated Task

Figure 9 shows the results of Whale and baselines after rolling out 64 steps in two different tasks. Notably, this qualitative evaluation is highly challenging and presents significant complexities. First, the evaluation rollout horizon is set to 64, exceeding that used in prior works, which imposes substantial demands on the generalizability and robustness of world models. Moreover, the variations between adjacent frames are subtle in the Meta-World environment, requiring world models to learn the semantics of actions from these minimal changes. In each image, the first row represents the real trajectory, while the others show the generated trajectories. It can be observed that Whale not only generates high-fidelity videos but also accurately restores the robot arm's pose. DreamerV3 is the baseline closest to Whale, but its generated trajectory still loses key information, such as the blue marker representing the target point. The other baselines fail to accurately model the robot arm's pose changes from the subtle variations between adjacent frames.

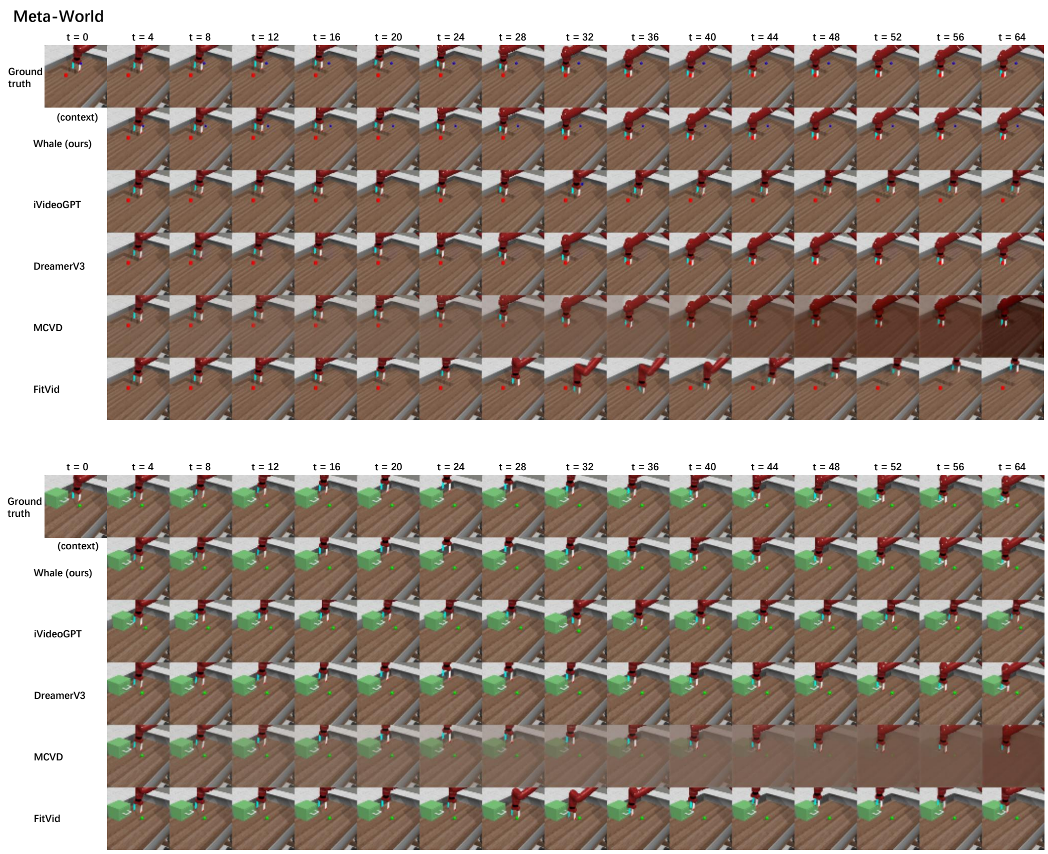


Figure 9: Additional qualitative evaluation on the Meta-World dataset.

F.2 Qualitative Evaluation on Open X-Embodiment Dataset

Figure 10 shows the qualitative evaluation results of Whale-X on Open X-Embodiment dataset. Whale-X demonstrates a remarkable ability to generate high-fidelity, action-conditioned trajectories. Moreover, with the aid of retracing-rollout and policy-conditioning techniques, Whale-X consistently delivers highly accurate predictions of the robotic arm's pose.

F.3 Qualitative Evaluation on Real-world Task

Figure 11 shows the qualitative evaluation results of Whale-X on Real-world Tasks. Whale-X demonstrates strong generalizability in terms of motion, visualization, and task combination.

G Policy Embedding Analysis

In this section, we conduct experiments to visualize the policy embeddings via t-SNE (Van der Maaten & Hinton, 2008) in order to verify whether our method can learn reasonable representations.

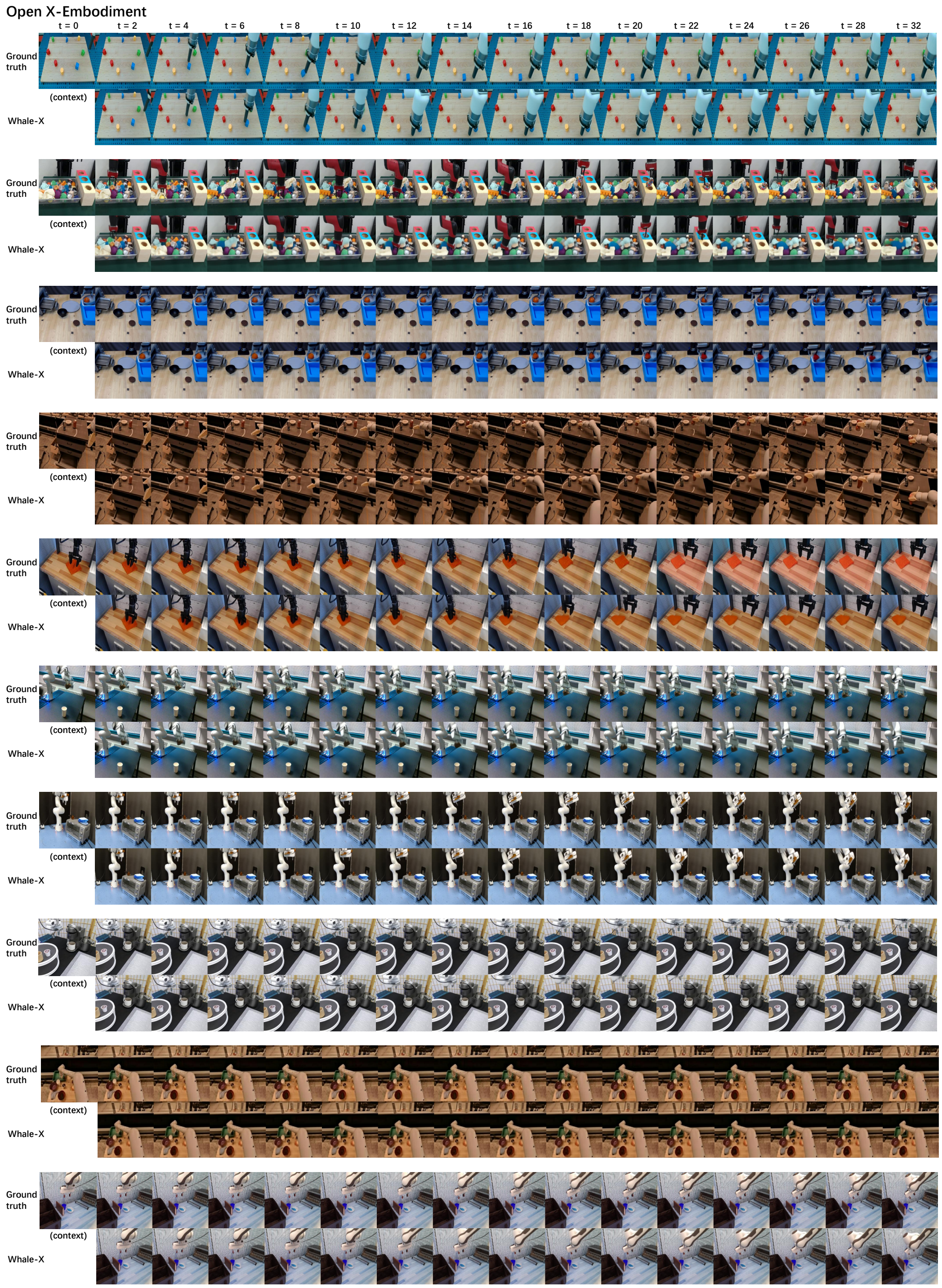


Figure 10: Additional qualitative evaluation on the Open X-Embodiment dataset.

Figure 12a shows that dfferent policies for the same task can be distinguished by the learned policy embeddings. Figure 12b shows that the expert policies for different tasks can also be distinguished, while Figure 12c shows the random policies for different tasks cannot. This distinction indicates that our learned embedding is more inclined towards policy representation rather than task representation.

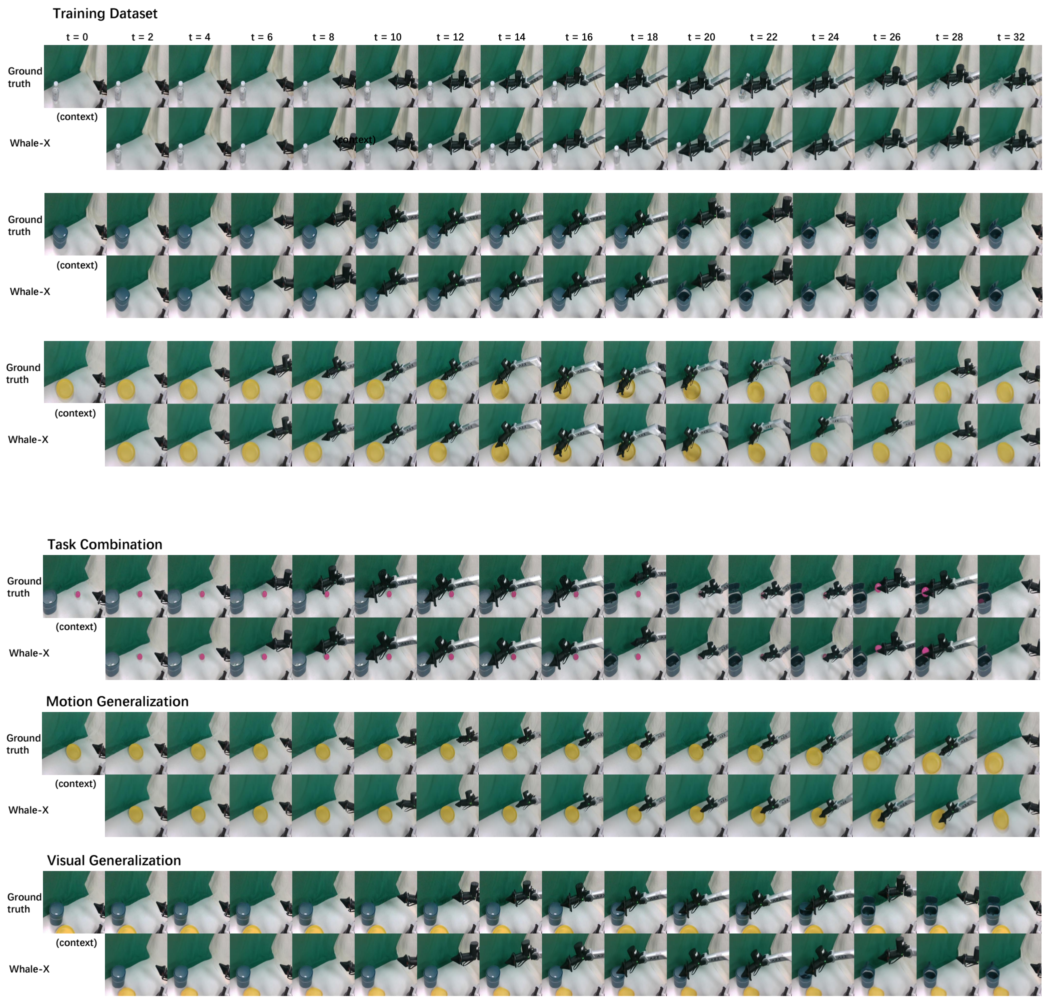
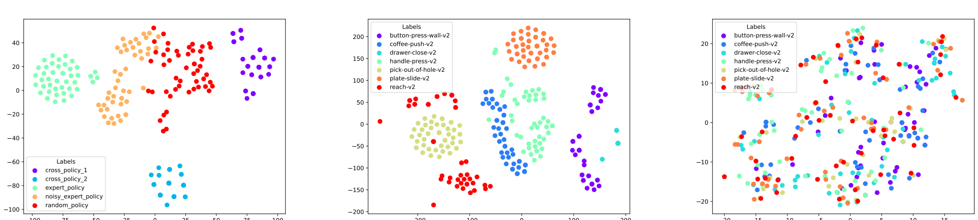


Figure 11: Additional qualitative evaluation on the Real-world tasks.



(a) Same task different policies (b) Different tasks expert policies (c) Different tasks random policies

Figure 12: The policy embedding visualization via t-SNE (Van der Maaten & Hinton, 2008). The different colors denotes different policies in the same task (12a) and different tasks' expert policies (12b) or random policies (12c). The separability validates the ability of the embeddings learned by our method to represent different policies.

H GPT-4O EVALUATION DETAILS

H.1 Q&A EXAMPLE

We use the large vision language model GPT-4o for evaluation in the physical robot experiment. Generally, we input the real final frame and the model-generated final frame to GPT-4o, using natural language dialogue to enable GPT-4o to assess whether the generated errors in key information such as the robot arm's position and the status of interactive objects fall within an acceptable range, thus determining whether the generated results are consistent with reality. Figure 13 shows one of our dialogue examples with GPT-4o. We use multi-turn dialogue to enable the model to easily process and infer information from images.

H.2 ALL GPT-4O PROMPTS

Table 14 contains all the prompts we used with GPT-4o for evaluation in unseen scenarios. The prompts evaluate various criteria by listing factors such as the robotic arm's position and the status of interactive objects.

Task	Prompt
Visual Generalization	Here are two images. The first image is the last frame of a real scene, and I will provide you with another image predicted by a model. The task is to open the trash bin under an unseen background. The trash bin is on the left side of the desk and is closed at the beginning. You need to determine if the two images are consistent based on the following criteria: 1) You can see the inside of the trash bin. 2) Is the predicted image clear? When all these criteria are satisfied, we call the predicted image is consistent with the real one. Now I will show you the real image.
Motion Generalization	Here are two images. The first image is the last frame of a real scene, and I will provide you with another image predicted by a model. You need to determine if the two images are consistent based on the following criteria: 1) Is the plate's position on the left side of the image? 2) Does the plate disappear in the predicted image? 3) Is the predicted image clear? 4) Is the robot arm still present in the predicted image? 5) Does the position of the robot arm in the predicted image match that of the real image? When all these criteria are satisfied, we call the predicted image is consistent with the real one. Now I will show you the real image.
Task Generalization	Here are two images. The first image is the last frame of a real scene, and I will provide you with another image predicted by a model. You need to determine if the two images are consistent based on the following criteria: 1) You can see the inside of the trash bin. 2) The ball should be simply missing and not on the desk. 3) Is there any severe distortion in the predicted image? 4) Is the position of the robot arm in the predicted image not far away from that of the real image? When all these criteria are satisfied, we call the predicted image is consistent with the real one. Now I will show you the real image.

Table 14: The prompt used for 3 unseen tasks.

H.3 MORE EVALUATION RESULTS

Figure 14 15 16 17 show the evaluation results for Whale-X, Whale-X (w/o policy conditioning), Whale-X (w/o retracing-rollout), and Whale-X (training from scratch) on the Visual Generalization task. Figure 18 19 20 21 show the evaluation results for Whale-X, Whale-X (w/o policy conditioning), Whale-X (w/o retracing-rollout), and Whale-X (training from scratch) on the Motion Generalization task. Figure 22 23 24 25 show the evaluation results for Whale-X, Whale-X (w/o policy conditioning), Whale-X (w/o retracing-rollout), and Whale-X (training from scratch) on the Task Generalization task.

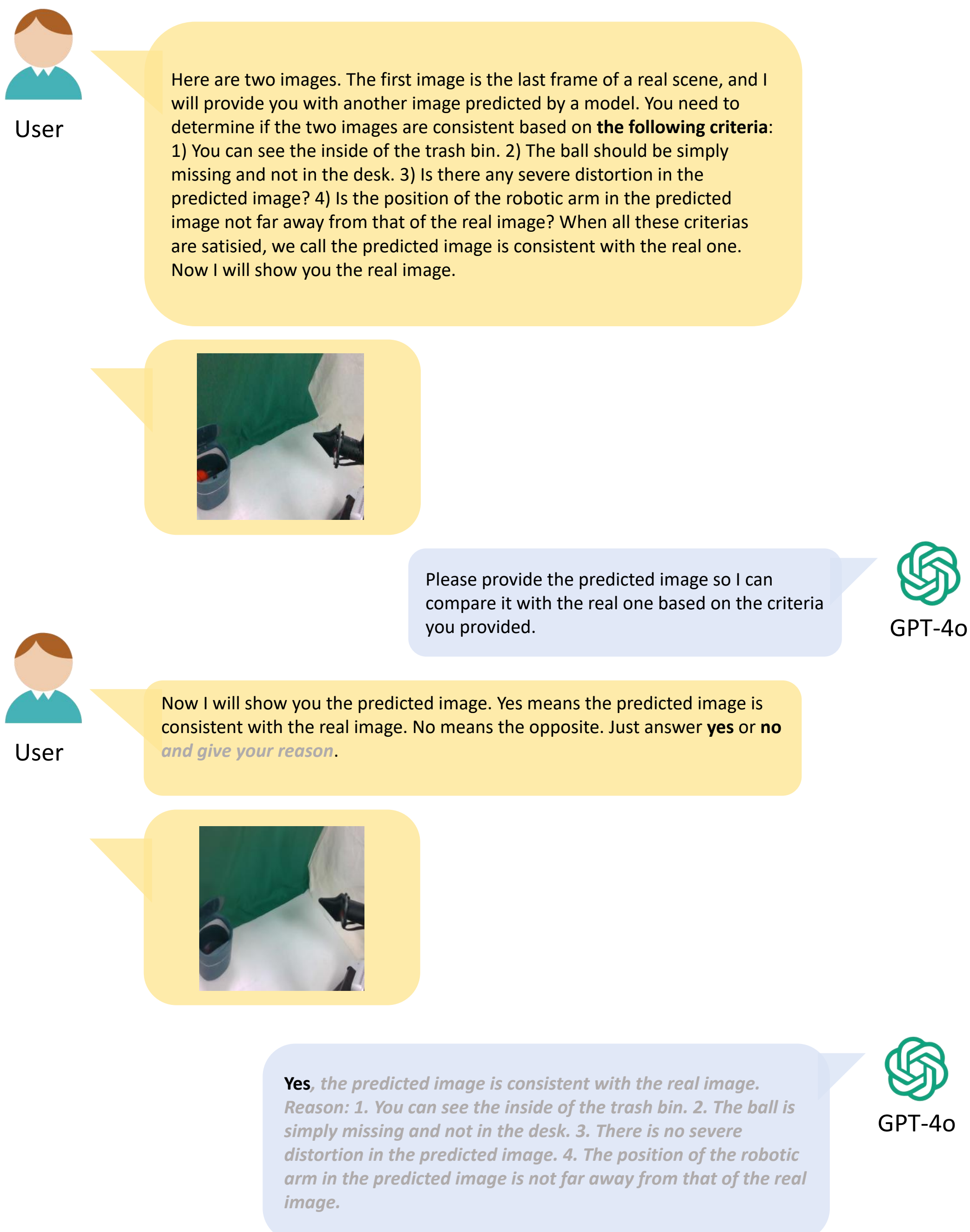


Figure 13: The illustration of a Q&A example using GPT-4o for evaluating the world model's consistency rate.

I COMPUTATIONAL RESOURCES

Our models for simulated Meta-World tasks are trained and evaluated on a single RTX 4090 GPU platform, while Whale-X for real-world robot manipulation is trained and evaluated using 8 RTX 4090 GPUs. On simulated tasks, it is approximately 2 days for tokenizer training, 8 hours for policy embedding model training, and 1 day for dynamics model training, totaling around 3 days. Pre-

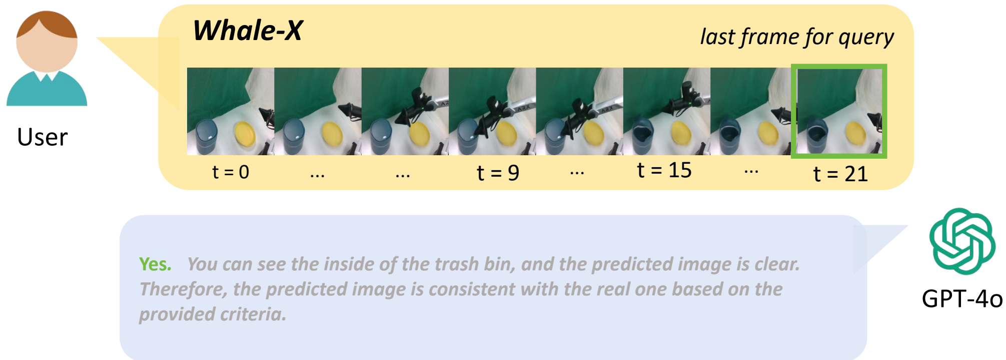


Figure 14: The example of GPT-4o evaluation for Whale-X on the Visual Generalization Task.

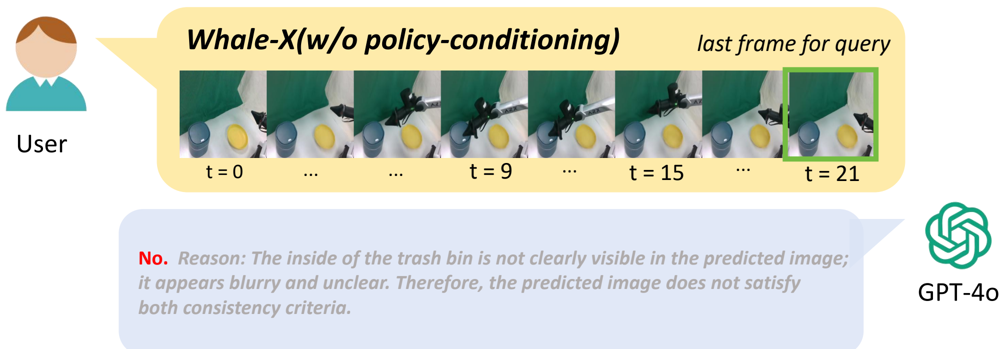


Figure 15: The example of GPT-4o evaluation for Whale-X(w/o policy-conditioning) on the Visual Generalization Task.

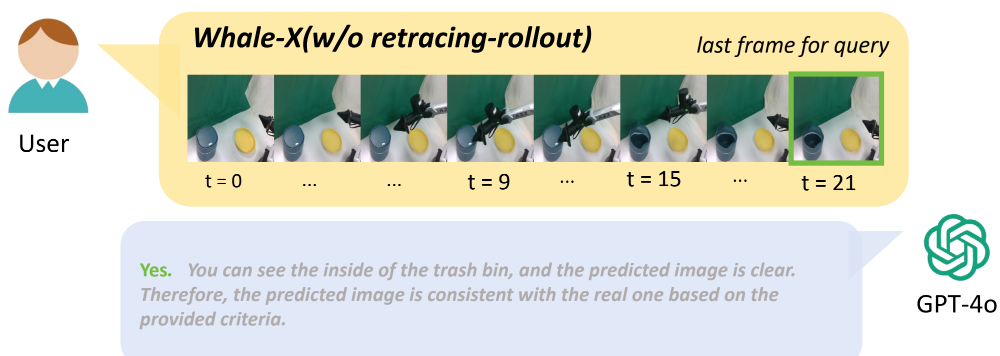


Figure 16: The example of GPT-4o evaluation for Whale-X(w/o retracing-rollout) on the Visual Generalization Task.

training a Whale-X-base model using 8 RTX-4090 GPUs takes about 10 days in total. Specifically, tokenizer training requires 6 days, dynamics model training takes 3 days, and policy embedding model training takes less than 1 day.

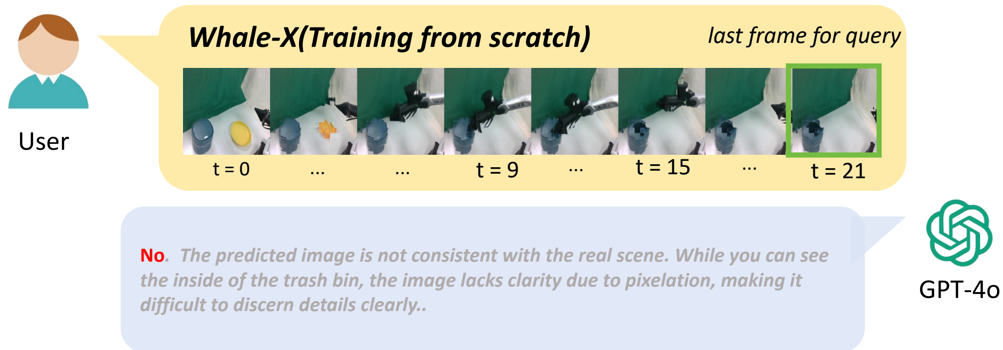


Figure 17: The example of GPT-4o evaluation for Whale-X(Training from scratch) on the Visual Generalization Task.

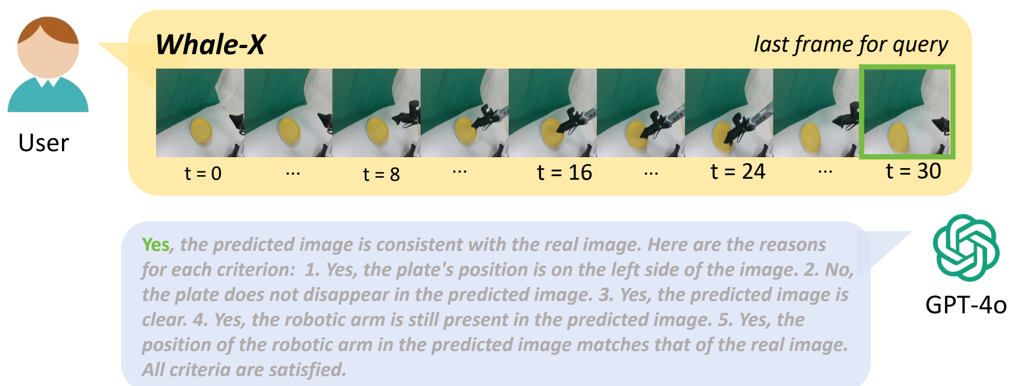


Figure 18: The example of GPT-4o evaluation for Whale-X on the Motion Generalization Task.

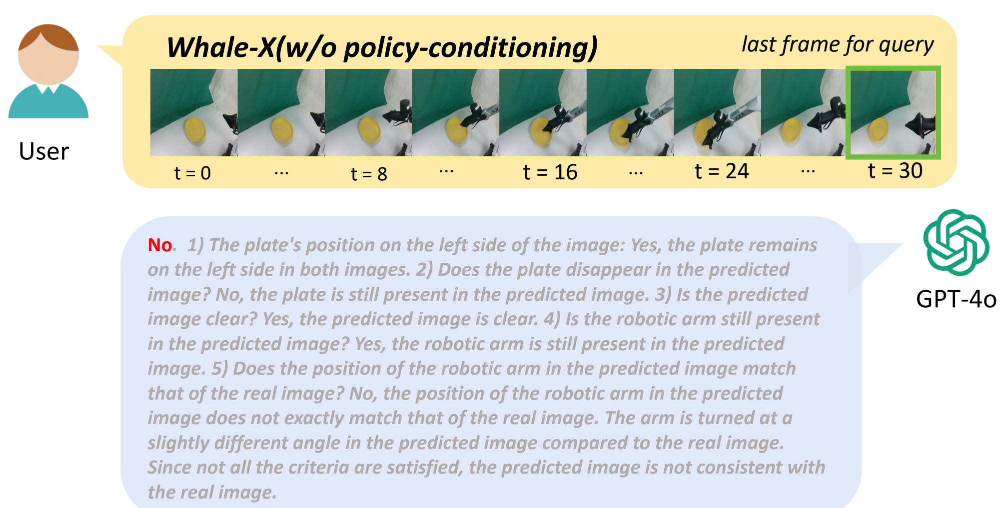


Figure 19: The example of GPT-4o evaluation for Whale-X(w/o policy-conditioning) on the Motion Generalization Task.

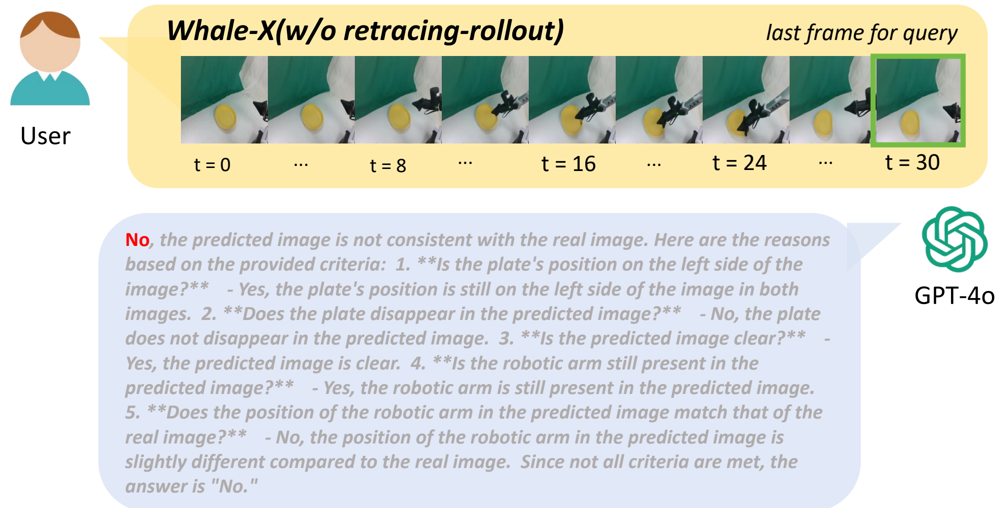


Figure 20: The example of GPT-4o evaluation for Whale-X(w/o retracing-rollout) on the Motion Generalization Task.

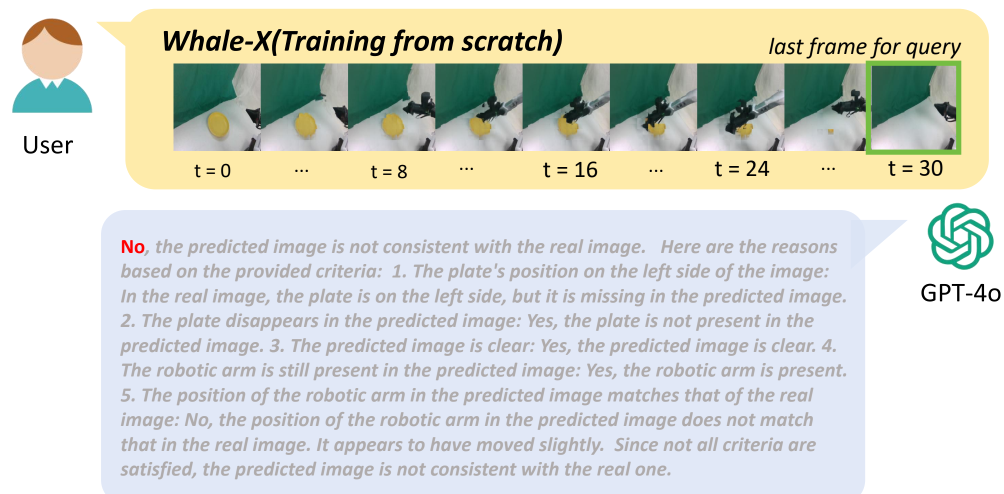


Figure 21: The example of GPT-4o evaluation for Whale-X(Training from scratch) on the Motion Generalization Task.

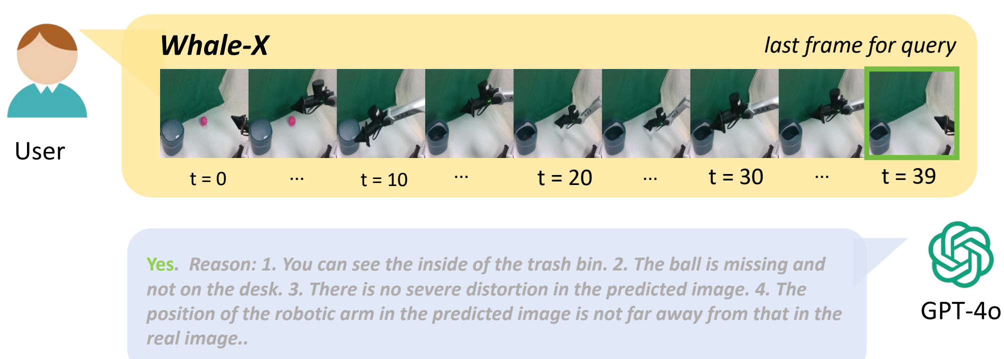


Figure 22: The example of GPT-4o evaluation for Whale-X on the Task Generalization Task.

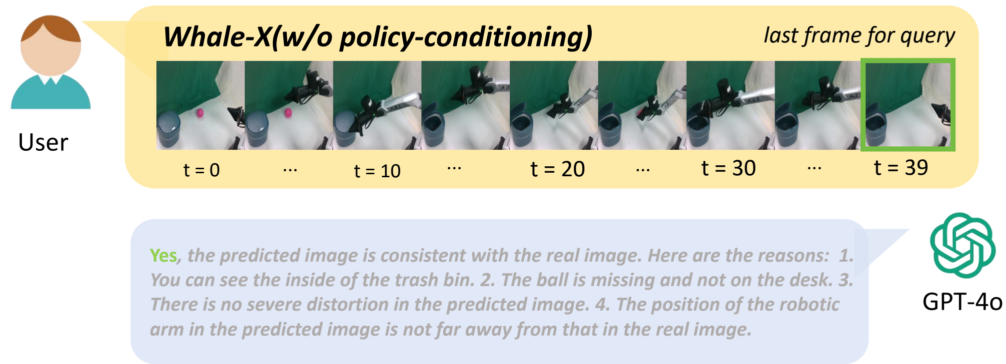


Figure 23: The example of GPT-4o evaluation for Whale-X(w/o policy-conditioning) on the Task Generalization Task.

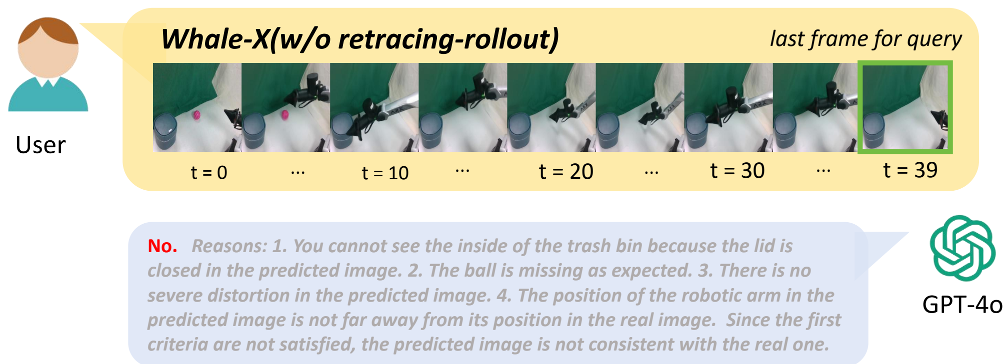


Figure 24: The example of GPT-4o evaluation for Whale-X(w/o retracing-rollout) on the Task Generalization Task.

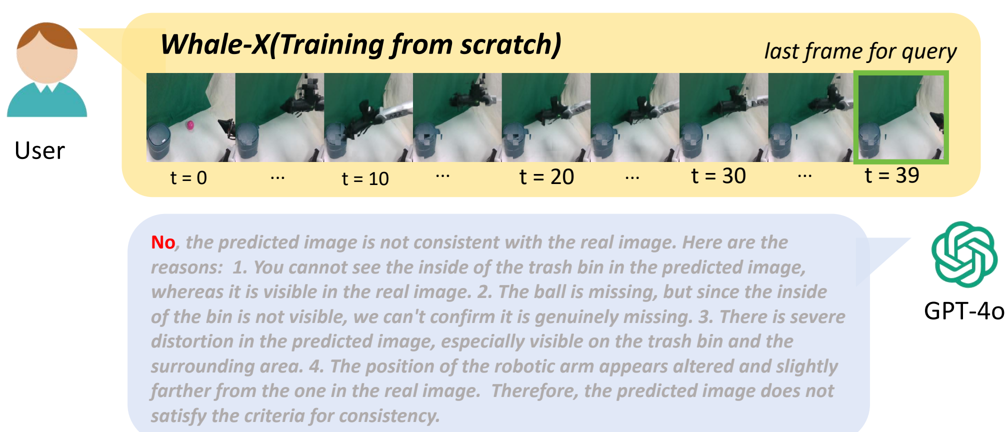


Figure 25: The example of GPT-4o evaluation for Whale-X(Training from scratch) on the Task Generalization Task.