WHALE: Towards Generalizable and Scalable World Models for Embodied Decision-making

Zhilong Zhang[1,2,3 *], **Ruifeng Chen**[1,2,3 *], **Junyin Ye**[1,2,3 *], **Yang Yu**[1,2,3 * †], **Yihao Sun**[1,2], **Haoxiang Ren**[2,3], **Xinghao Du**[1,2,3], **Pengyuan Wang**[1,2,3], **Jingcheng Pang**[1,2,3], **Kaiyuan Li**[1,2,3], **Tianshuo Liu**[1,2,3], **Haoxin Lin**[1,2], **Zhi-Hua Zhou**[1,2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
[3]Polixir Technologies, Nanjing, China

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. To facilitate effective decision-making, world models must be equipped with strong generalizability to support faithful imagination in out-of-distribution (OOD) regions, which present significant challenges for previous approaches. This paper introduces WHALE, a framework for learning generalizable world models with the behavior-conditioning technique, aiming to address the policy distribution shift, one of the primary sources of world model generalization errors. Building upon this, we instantiate WHALE as a scalable vision-based world model built on a spatial-temporal transformer architecture, designed to support high-fidelity imagination over long horizons. We further introduce WHALE-X, a 414M parameters world model pre-trained on 970K Open X-Embodiment trajectories, exhibiting promising scalability and generalizability in real-world manipulation tasks using minimal demonstrations.

1 Introduction

Human beings can envision an imagined world in their minds, predicting how different actions might lead to different outcomes [1, 2]. Inspired by this aspect of human intelligence, world models [3] are designed to abstract real-world dynamics and provide such *"what if"* predictions. 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 [4], off-policy evaluation [5], and offline reinforcement learning [6].

In the realm of embodied intelligence, interactive world models offer promise by reducing reliance on costly real-world interactions through predictive modeling of future observations and dynamics—thereby enabling efficient and effective policy evaluation and learning []. However, early world models were largely limited to simple tasks or narrow environments, raising concerns about their generalization and scalability to complex, real-world robotic scenarios [7, 8, 9]. Although recent efforts have used larger datasets and model parameters to improve generalization [10, 11, 12], these models still struggle to support robust, long-horizon decision-making. A fundamental challenge lies in the distribution shift that arises from the policy divergence during training and evaluation phases, which remains largely unaddressed.

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

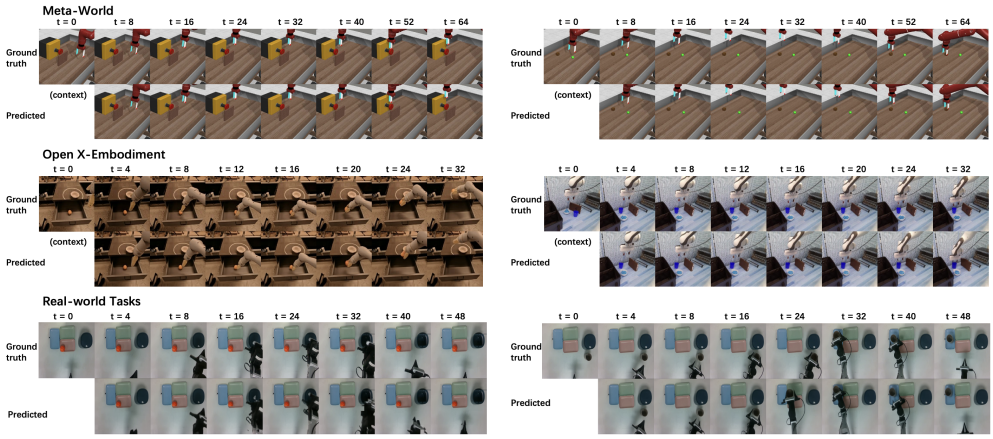


Figure 1: Qualitative evaluation on Meta-World, Open X-Embodiment, and our real-world tasks.

In this work, we first analyze how the world models effect the *value gap* of policy evaluation, the discrepancy between the policy value in the real world and estimated in world models due to distribution shift, revealing that standard world models fail to generalize when target policies differ from training-time behaviors. Even with perfect in-distribution fitting, the divergence between policy-induced trajectory distributions leads to significant extrapolation error, amplified by long-horizon rollouts. Previously, PCM [13] sought to address this issue by introducing a meta-dynamics model conditioned on evaluation policies, which it implemented through the regularization of RNN representations to recover policies. However, the limitation to the recurrent architecture and proprioceptive-state control prohibits its applicability to transformers and scalability to complex, high-dimensional embodied tasks.

To address this challenge, we introduce WHALE (**W**orld models with be**HA**vior-conditioning **LE**arning), a framework for learning scalable and generalizable interactive world models for embodied agents. WHALE extracts behavioral patterns from trajectories and embeds them into latent representations that are conditioned on the world model, enabling it to recognize policy-specific dynamics and adaptively mitigate distribution shifts during autoregressive rollouts. To instantiate this framework, we propose a scalable embodied world model based on a spatial-temporal Transformer architecture [14, 11], designed for accurate and coherent long-horizon imagination in visual control tasks. Our model is trained on a large-scale dataset of 970k real-world robotic manipulation trajectories from the Open X-Embodiment dataset [15], resulting in a 414M parameters world model that generalizes across diverse robots, tasks, and environments. This pre-trained world model serves as a simulator for evaluating real-world policies and demonstrates strong generalizability across multiple environments and robots.

Extensive experiments on both simulated benchmarks [16] and a real-world robotic platform demonstrate that WHALE outperforms existing methods in video prediction fidelity and value estimation accuracy. Furthermore, our ablation studies reveal consistent performance gains with increasing model capacity and data scale, highlighting the framework's excellent scalability and potential for continued improvement with larger resources.

The primary contributions of this work are as follows:

- We propose WHALE, a behavior-conditioned world model framework that embeds behavior latent representations to mitigate distribution shift.
- We instantiate the WHALE framework with a spatial-temporal Transformer-based architecture and pre-train a 414M parameters world model on 970k real-world demonstrations.
- We conduct extensive experiments to show the scalability and generalizability of WHALE across simulated and real-world tasks.

2 Related Works

Despite a long history [17, 18] and rich literature on environment models [19, 20, 21, 22, 23, 24, 25], the focus has primarily been on modeling transition dynamics within lower-dimensional proprioceptive state spaces until recently. [3] was the first to propose a general framework to model dynamics for high-dimensional visual observations, introducing the term "world models". This generic architecture soon achieved a series of notable successes in complex decision-making tasks [7, 26, 8, 27]. However, world models are supposed to answer "what if" questions: *"What will happen in the environment if the agent makes any possible decisions?"*, which must be highly out-of-distribution and has yet to be fully addressed.

A potential solution to this generalization issue is to collect more data to train large world models. Recently, advanced methods have leveraged modern action-conditioned video prediction models [28, 29] to model visual dynamics and pre-train from large-scale video experience data [11, 30, 31, 32, 12]. Despite the large amount, the available training data are normally collected by expert or near-expert policies, leading to low data coverage, posing challenges to reasoning decision outcomes for suboptimal policies in the learned world models [33]. Another line of work investigates the impact of learning methods on world model generalizability. For single-step maximum likelihood objectives, the autoregressive rollout suffers from policy divergence and compounding errors [20, 34, 35]. To overcome the limitations in the standard MLE learning, a series of improvements have been made [36, 37, 4, 38, 39]. Despite the successes in lower-dimensional tasks, scaling these methods to large amounts of high-dimensional visual data remains an open problem.

3 Foundations of World Model Learning

3.1 Problem Formulation

An markov decision process (MDP) [40] is 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 [41], 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: $V^\pi_{T^*} = \mathbb{E}_{\tau_H \sim (\pi, T^*)}[\sum_{t=1}^{H} r(s_t, a_t)]$, where trajectories $\tau_H = (s_1, a_1, \ldots, s_H, a_H)$ and rewards are obtained by rolling out policy π within dynamics T^*.

An environment model T can be explicitly learned from offline data to imitate the real transition T^*.

Given the value V^π_T estimated within the model T, the model error induces a value gap $|V^\pi_{T^*} - V^\pi_T|$ for the policy π. Since offline data are typically collected by a narrow range of (near-expert) policies, the learned models may struggle with unfamiliar outcomes of novel behaviors and are expected to generalize beyond the training experiences for counterfactual reasoning.

3.2 Generalizability of World Models

The common learning methods for 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.,

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

where (sub-)trajectory $\tau_h = (s_1, a_1, s_2, \ldots, s_h, a_h), 1 \le h \le H$ 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 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:

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

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. For world model learning, however, the distribution shift results from both the model prediction inaccuracy and the divergence between the target policy and behavior policies, exacerbating the evaluation inaccuracy:

$$\left| V_T^\pi - V_{T^*}^\pi \right| \leq 2R_{\max} \underbrace{H^2}_{\text{teacher-forcing}} \Big(\underbrace{\sqrt{2\, l_{\mathrm{KL}}(T; \Pi)}}_{\text{in distribution error}} + \underbrace{L \cdot W_1(d^\pi, d^\Pi)}_{\text{policy divergence}} \Big), \tag{1}$$

where a distribution shift term induced by the policy divergence * occurs in addition to the KL training loss, further amplified by an H^2 factor caused by the supervised teacher-forcing learning. Even if the world model perfectly models the training distribution, i.e. $l_{\mathrm{KL}}(T; \Pi) = 0$, the variation of the target policies could also significantly shift the trajectory simulation distribution to those large error areas, resulting in degenerative generalizability. Further detailed analysis can be found in Appendix A.

4 Learning Generalizable World Models for Embodied Decision-making

In this section, we introduce **WHALE**, a framework for learning scalable and generalizable world models. The section is organized as follows: we begin with the foundation of behavior-conditioning in Section 4.1, deriving an objective that encourages behavior embeddings to capture policy patterns. We then present the practical architecture of WHALE in Section 4.2. Finally, in Section 4.3, we describe the pre-training and fine-tuning pipeline for **WHALE-X**, a 414M parameters world model trained on large-scale real-world demonstrations.

4.1 Behavior-conditioning for Generalization

According to Eq (1), the generalization error of the world model primarily arises from error compounding caused by policy divergence. One solution to this policy generalization issue is to embed the behavior information into the world model, allowing the model to actively recognize the behavior patterns of the policies and adapt to the policy-induced distribution shift [13]. This adaptation effect has been shown to reduce model generalization error caused by policy divergence, i.e. the last term in Eq (1). For further analysis, please refer to Appendix A. Building upon behavior-conditioning, we introduce a learning objective to obtain behavior embeddings from training trajectories.

We would like to extract the decision patterns within training trajectories τ_H into a behavior embedding, reminiscent of the maximization of the evidence lower bound (ELBO) of the trajectory likelihood conditioned on the history τ_h [42, 43]:

$$\log P(\tau_H | \tau_h) \geq \mathbb{E}_{q_\phi(z|\tau_H)} \sum_{t=h}^{H} \log \pi_w(a_t | s_t, \tau_{t-1}, z) - D_{\mathrm{KL}}(q_\phi(z|\tau_H) || p_\psi(z|\tau_h)) + Const \tag{2}$$

where $q_\phi(z|\tau_H)$ is the posterior encoder, encoding the whole trajectory τ_H into a latent variable z; $p_\psi(z|\tau_h)$ denotes the prior predictor, allowing the prediction of z based on the history τ_h; $\pi_w(a_h|s_h, \tau_{h-1}, z)$ denotes the action decoder, recovering the decision action from the latent variable z and the up-to-date history (τ_{h-1}, s_h). The information bottleneck requires the learned variable z to effectively capture the decision pattern within the trajectory, embedding the information about the behavior policy. Following this argument, we propose to learn the behavior embedding by maximizing the ELBOs over H decision steps and adjusting the amount of KL constraints similar to β-VAE [44]:

$$\mathcal{L}(w, \phi, \psi) = \mathbb{E}_{\tau_H \sim \mathcal{D}} \sum_{h=1}^{H} \Big[\mathbb{E}_{q_\phi(z|\tau_H)} \log \pi_w(a_h | s_h, \tau_{h-1}, z) + \beta D_{\mathrm{KL}}(q_\phi(z|\tau_H) || p_\psi(z|\tau_h)) \Big] \tag{3}$$

*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 [13].

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 exhibit similar behavioral patterns and, consequently, similar representations.

The learned prior predictor p_ψ is then used to obtain behavior embeddings z_h from history τ_h for world model learning, where z_h serves as additional contexts for future prediction:

$$\mathbb{E}_{\tau_H \sim (\mu, T^*)} - \frac{1}{H} \sum_{h=1}^{H} \mathbb{E}_{z_h \sim p_\psi(\cdot|\tau_h)} \log T(s_{h+1}|\tau_h, z_h). \tag{4}$$

When rolling out target policies or executing action sequences within the learned world models, the prior predictor infers the latent behavior intentions from interaction history, adjusting the autoregressive generation process to the target distribution on the fly for future imagination adaptively.

4.2 Practical Implementation

In this section, we describe the practical implementation and general algorithm of WHALE, a scalable and generalizable world model built upon a Spatial-Temporal Transformer [45] backbone.

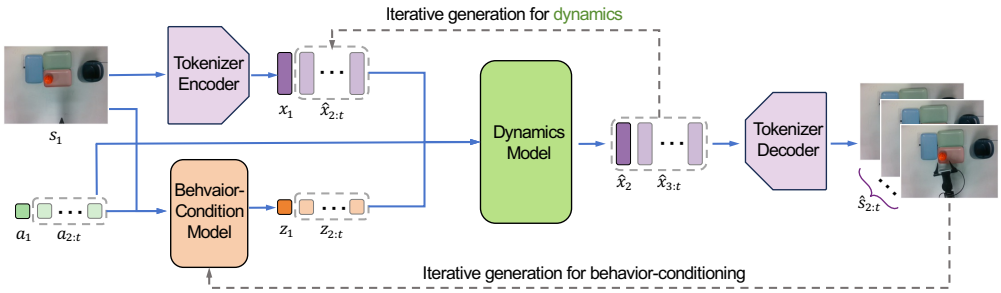


Figure 2: Model architecture of WHALE.

Figure 2 illustrates the model architecture of WHALE, consisting of three main components: video tokenizer, behavior-conditioning model, and dynamics model.

Video Tokenizer. The video tokenizer is implemented as a VQ-VAE [46], which compresses the pixel-based observations s into the discerete tokens x. It consists of an encoder and a decoder, with a normalized vector quantization (VQ) codebook. The model is jointly optimized using a reconstruction loss and a codebook regularization term.

Given an input observation sequence $s_{h:h+t}$, we first divide it into patches. Each patch is encoded by the encoder into a latent vector, and the nearest vector in the normalized codebook is selected via Euclidean distance to produce the discrete latent code z_q. These discrete codes are then passed through the decoder to generate the reconstructed output $s^{\text{rec}}_{h:h+t}$.

Behavior-Conditioning Model. The behavior-conditioning model consists of three core components: a posterior encoder, a prior predictor, and an action decoder. These modules are jointly optimized to maximize the behavior embedding objective (see Eq. 3). We also employ two-hot encoding to represent the latent behavior embeddings, enabling differentiable discretization and improving both the resolution and training stability of the learned representations [26].

Given an input trajectory $\tau_{1:H}$, we first split each observation s_t into patches. The posterior encoder takes the full sequence of patches and corresponding actions $\{(s_t, a_t)\}_{t=1}^{H}$ as input and encodes the posterior behavior embedding z_{post}. In contrast, the prior predictor operates using only the history up to time h (i.e., $\{(s_t, a_t)\}_{t=1}^{h}$ with $h < H$) to predict a behavior embedding z_{prior}. The action decoder π_w reconstructs the current action a_h using the visual patches from past observations $\{s_t\}_{t=1}^{h}$, the corresponding past actions $\{a_t\}_{t=1}^{h-1}$, and the prior behavior embedding z_{prior}.

Dynamics Model. The dynamics model is designed to predict future visual observations in the discrete latent space obtained from the video tokenizer. Given a history of encoded visual tokens $x_{1:h}$ and actions $a_{1:h}$, the model predicts the subsequent latent token x_{h+1}. At each timestep h,

the input to the dynamics model is constructed by combining embedded visual and action tokens: $\overline{x}_h = \mathbf{E}_{\text{vis}}(x_h) + \mathbf{E}_{\text{act}}(a_h)$, where $\mathbf{E}_{\text{vis}}$ and $\mathbf{E}_{\text{act}}$ denote learnable embedding tables for visual tokens and actions, respectively. To enable behavior-aware prediction, the model conditions on a high-level behavior embedding z_{prior}, generated by a behavior-conditioning model. This embedding is incorporated into the dynamics model through cross-attention applied to the sequence $\{\overline{x}_1, \ldots, \overline{x}_h\}$, allowing the model to modulate its predictions based on the intended policy.

During training, the model is optimized to minimize the cross-entropy loss between predicted $\hat{x}_{h+1}$ and ground-truth future tokens x_{h+1}. At inference time, the model unrolls autoregressively in the latent space: the predicted token $\hat{x}_{h+1}$ is fed back as input for the next step, enabling long-horizon visual prediction conditioned on both actions and behavioral intent.

4.3 Training Pipeline

Overall Algorithm. Our world model is trained in three stages: (1) the video tokenizer learns to compress observations into discrete tokens; (2) the behavior-conditioning model learns policy-aware latent embeddings from trajectories; (3) the dynamics model predicts future tokens autoregressively.

Pre-training. We introduce WHALE-X, a 414M parameters world model pre-trained on 970K real-world robot demonstrations from the Open X-Embodiment dataset. We list our used data mixture and weights in Table 8, all of which are used to pre-train the video tokenizer and behavior-conditioning model. To train a world model focused on tabletop tasks, we only use data related to tabletop tasks from the dataset (the bolded tasks in Table 8) to train the dynamics model.

Fine-tuning. The fine-tuning training pipeline follows a similar three-stage structure as pre-training: video tokenizer tuning, behavior-conditioning model tuning, and dynamics model training. The key difference lies in the video tokenizer stage: we freeze the encoder while only updating the decoder. This ensures that the pretrained discrete representation of states remains unchanged, preserving the pretrained world model's understanding of visual dynamics and enabling more stable and effective adaptation to downstream tasks.

5 Experiment

We conduct experiments on both simulation tasks and real-world tasks, which are primarily designed to answer the following key questions: (1) How does WHALE perform compared with other baselines on simulated tasks and real-world tasks? (2) Does the behavior-conditioning technique effectively improve the world model generalizability? (3) How is the scalability of WHALE? Does increasing the model capacity or pre-training data improve performance?

5.1 Simulation Tasks Experiments

Data. We conduct our simulated task experiments on the Meta-World [16] benchmark. We construct a training dataset with 60k trajectories collected from 20 tasks. For evaluation, we assess the learned world models on 200 held-out trajectories collected by 5 unseen policies per task to test world model generalizability to novel behavioral patterns. Detailed information about data collection can be found in Appendix C.1.

Baselines. We compare Whale against several world model learning baselines, including (1) **FitVid** [47], a variational-based world model that can fit large video datasets. (2) **MCVD** [48], a diffusion-based world model that can perform video generation conditioning on different subsets of video frames and actions. (3) **DreamerV3** [26], a recurrent world model that outperforms specialized methods across diverse control tasks. (4) **iVideoGPT** [12], a scalable transformer-based world model that achieved state-of-the-art results in video generation and embodied control tasks. Complete descriptions are provided in Appendix B.3.

Evaluation Metrics. We assess the performance of world models from two perspectives: (1) *Value estimation accuracy*. Verifies whether the model can correctly estimate the value of a given action sequence, in terms of Value Gap, Return Correlation, and Regret [5]. (2) *Video fidelity*. Measures the quality of video trajectory generation, in terms of FVD [49], PSNR [50], LPIPS [51], and SSIM [52]. More detailed information about evaluation metrics is provided in Appendix B.5.

Task Results. As shown in Table 1, WHALE achieves state-of-the-art performance in both value prediction accuracy and video fidelity under the from-scratch training setting, outperforming existing

world models across all metrics. Notably, WHALE achieves a value gap of 4.7 and regret@5 of 6.7 at 256×256, surpassing DreamerV3 and other strong baselines, while also attaining the lowest FVD and best SSIM, indicating superior visual realism. These results validate the effectiveness of its architecture design and behavior-conditioned dynamics modeling. When pre-training is introduced (blue-dashed rows), WHALE-X further enhances performance, achieving the smallest value gap (3.7) and the highest return correlation (0.86) among all models, while maintaining superior video fidelity. This demonstrates that pre-training on large-scale interaction data significantly enhances both predictive accuracy and generative quality, and underscores the importance of leveraging prior experience and model scale for building precise, high-fidelity world models.

Ablation Study. To assess the impact of behavior-conditioning, we compare WHALE with its ablated version (w/o bc) at 256×256 resolution. The results show that removing behavior-conditioning leads to a noticeable degradation in value prediction (e.g., value gap increases from 4.7 to 6.8 in the from-scratch setting) and a consistent drop in video fidelity (FVD rises from 25.0 to 27.6), confirming that conditioning dynamics on policy actions improves both accuracy and visual quality, further validating that modeling action-conditioned transitions is critical for capturing realistic and policy-relevant dynamics. The consistent gains across settings affirm behavior-conditioning as a core component of our framework.

Meta-World	#Params	Value Gap↓	Return Corr↑	Regret@5↓	FVD↓	PSNR↑	SSIM↑	LPIPS↓
64×64 resolution	*Scratch*	*Value accuracy*			*Video fidelity*			
FitVid	143M	18.2	0.64	22.0	154.6	23.7	90.3	6.5
MCVD	53M	20.6	0.72	12.2	272.8	**29.7**	92.3	4.0
DreamerV3	44M	**10.0**	0.70	16.5	142.7	27.6	92.1	4.3
iVideoGPT	63M	15.9	0.62	**7.2**	115.7	28.5	92.8	4.5
WHALE	51M	**10.3 ± 0.8**	**0.77±0.01**	**7.3±1.2**	**38.5±2.6**	28.8±0.0	**93.5±0.1**	**3.7±0.1**
256×256 resolution	*Scratch*	*Value accuracy*			*Video fidelity*			
DreamerV3	61M	8.5	0.69	14.5	92.8	24.2	89.9	8.6
WHALE (w/o bc)	61M	6.8±0.2	0.82±0.01	9.4±1.5	27.6±2.4	26.9±0.1	92.6±0.1	4.7±0.1
WHALE (ours)	63M	**4.7±0.1**	**0.83±0.01**	**6.7±0.6**	**25.0±3.4**	**27.6±0.2**	**94.2±0.1**	**4.4±0.1**
256×256 resolution	*Pre-trained*	*Value accuracy*			*Video fidelity*			
iVideoGPT	448M	13.1	0.77	8.1	568.7	24.1	89.9	11.2
WHALE-X (w/o bc)	398M	3.9	0.84	6.4	25.2	27.4	94.1	4.4
WHALE-X (ours)	414M	**3.4**	**0.87**	**1.8**	**23.8**	**27.7**	**94.4**	**4.2**

Table 1: Value prediction accuracy and video fidelity comparison on Meta-World benchmark.

5.2 Real-world Task Experiments

Platform. We evaluate the real-world generalization capabilities of WHALE using the ARX5 mobile manipulator, a platform equipped with a 6-DoF robotic arm and integrated onboard sensors. This setup introduces significant domain shifts compared to the simulation data used during pre-training—spanning robot morphology, camera viewpoints, and environmental layout—thereby presenting a rigorous test for sim-to-real transfer.

Tasks Design and Data Collection. Our experiments focus on three diverse manipulation tasks: *open trash bin*, *pick & place cup*, and *throw ball*. For fine-tuning, we collect 50 trajectories per task, consisting of 20 human teleoperation demonstrations and 30 self-collected trajectories generated by diverse policies, including Action Chunking Transformer (ACT) [53], Diffusion Policy (DP) [54], and π_0 [55]. During evaluation, we assess each learned world model using three previously unseen policies per task. For each policy, we perform 20 rollouts within the world model to generate long-horizon trajectories (more than **200 interactions** per trajectory). Detailed descriptions of the fine-tuning procedure, policy implementations, and data collection protocols are provided in Appendix B.2, Appendix B.4, and Appendix D, respectively.

Baselines. To ablate key components of our framework, we compare WHALE-X against three carefully designed baselines: (1) **iVideoGPT**: a state-of-the-art autoregressive world model pretrained on large-scale vision-language data; (2) **WHALE-X w/o Behavior-Conditioning**: an ablated variant that removes policy-aware conditioning from the dynamics model; (3) **WHALE-X from Scratch**: a version trained without pre-training.

Evaluation Metrics. We evaluate world models along two primary dimensions: *policy evaluation accuracy* and *video fidelity*. Since pixel-level rewards are not available in the real world, direct value

prediction cannot be assessed. Instead, we estimate value accuracy by simulating 20 rollouts per policy in the learned world model and computing the predicted success rates. We then measure how well these predictions correlate with the actual performance of the same policies in the real world, using two metrics: *Mean Maximum Rank Violation (MMRV)* and *Rank Correlation* [56]. Implementation details for these metrics are provided in Appendix B.5. For video fidelity, we compute standard perceptual and reconstruction metrics to assess the visual realism and temporal coherence of generated videos.

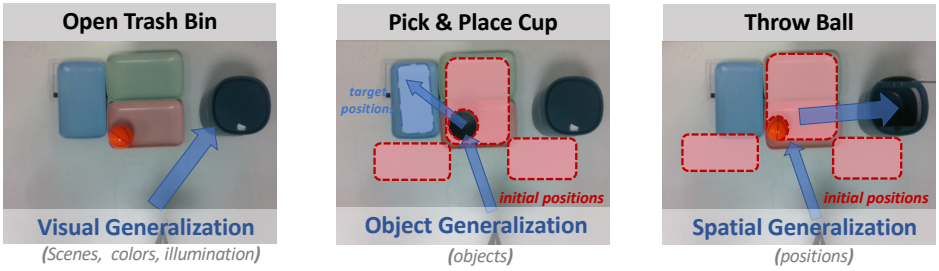


Figure 3: Real-world tasks illustration.

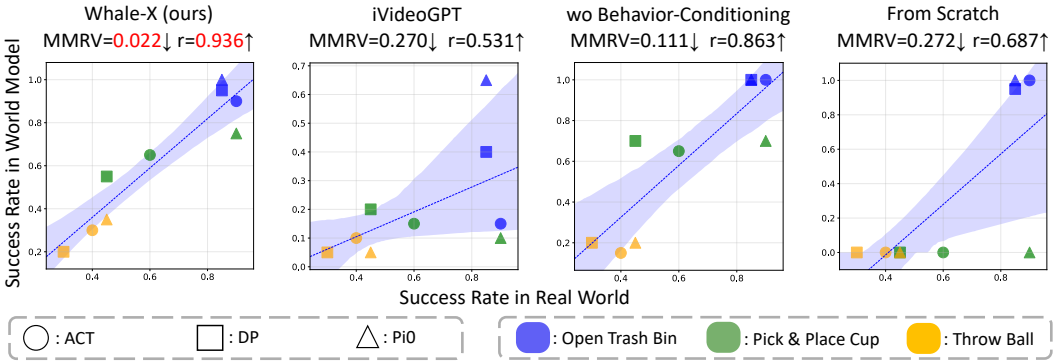


Figure 4: Comparison of World Model Variants on real-world policy evaluation tasks. The x-axis represents real-world success rates, while the y-axis shows success rates in the world model.

Task Results. WHALE-X demonstrates strong real-world performance across diverse manipulation tasks, achieving high fidelity in policy evaluation and robust generalization to unseen behaviors, as shown in Figure 4. The results highlight three key advantages:(1) **Superiority over iVideoGPT.** WHALE-X significantly outperforms the strong pre-trained baseline iVideoGPT, reducing MMRV by 82% and improving rank correlation by 76%. This substantial gain underscores the effectiveness of the framework design and implementation of WHALE-X. (2) **Critical Role of Behavior Conditioning.** Ablating behavior-conditioning leads to a dramatic 80% increase in MMRV, with WHALE-X w/o Behavior-Conditioning failing to reliably rank unseen policies. (3) **Necessity of Pre-training.** WHALE-X trained from scratch performs poorly, attaining less than half the rank correlation of its pretrained counterpart. This highlights the indispensable role of large-scale pre-training.

As shown in Table 10, WHALE-X also produces visually coherent and realistic video rollouts, outperforming all baselines in video fidelity metrics. This combination of high visual quality and accurate dynamics modeling enables reliable simulation within the learned world model.

Importantly, WHALE-X enables highly efficient policy evaluation: it simulates 20 rollouts in under 2 minutes on a single RTX-4090 GPU—achieving a **20× speedup** over real-world execution. This efficiency makes WHALE-X a practical tool for rapid policy screening and iterative development in real-world robotic systems.

5.3 Scaling Experiments

In this section, we aim to investigate the scaling behavior of WHALE-X. Specifically, we freeze the video tokenizer and behavior-conditioning model, adjusting only the model size and pre-training data size of dynamics models.

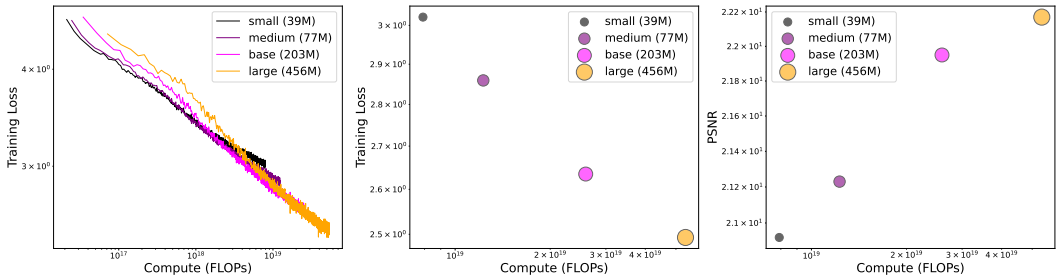


Figure 5: Scaling experiment results of WHALE-X. The leftmost plot shows the training loss curves for models with varying parameter sizes during the pre-training phase. The second plot presents the final training loss for all models after 300k pre-training steps. The third plot displays the PSNR after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

Pre-training Scaling Experiments. With a frozen video tokenizer and behavior-conditioning model, we pretrain four dynamics models ranging in size from 39M to 456M parameters. The results, presented in the first two plots of Figure 5, demonstrate that WHALE-X exhibits strong scalability. Specifically, increasing either the amount of pretraining data or the model size consistently leads to a reduction in training loss. Moreover, we observe that the training loss of WHALE-X approximately follows a log-linear relationship with FLOPs.

Fine-tuning Scaling Experiments. To this end, we fine-tune a series of dynamics models and show the PSNR results in the rightmost plot in Figure 5. The results indicate that after fine-tuning, the larger model demonstrates a larger PSNR value on test data, highlighting the promising scalability of WHALE-X for real-world tasks.

6 Discussions and Limitations

In this paper, we introduce WHALE, a framework of world model learning that incorporates the behavior-conditioning technique to enhance OOD generalization. Building on this foundation, we present a scalable ST-transformer-based implementation and pre-train a 414M parameters WHALE-X on large-scale robot data to assist robot manipulation. WHALE enables high-fidelity imagination and accurate policy evaluation, even in novel scenarios, thereby facilitating downstream control tasks.

Failure Case Analysis. Although WHALE-X exhibits strong generalizability, generative world models inevitably encounter hallucinations and other types of failure cases. In this work, we systematically categorize these failures into three distinct types: (1) **Object Errors**, which result in missing objects, unrealistic deformations, inconsistent scene layouts, or disrupted temporal continuity; (2) **Dynamics Errors**, where the model's predicted transitions visibly violate physical constraints, leading to implausible movements of robotic arms; (3) **Visual Errors**, involving the generation of blurry, incorrect, or visually implausible images by the world model. The distribution of these failure types is illustrated in Figure 6. Among them, *Dynamics Errors* constitute the largest proportion of WHALE-X's failure cases.

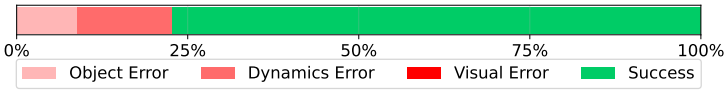


Figure 6: Failure case analysis of WHALE-X.

Computation Resources. We conduct all experiments on RTX 4090 GPUs. Pre-training WHALE-X takes around 2000 GPU hours, and fine-tuning WHALE-X requires an additional 24 GPU hours. During inference, WHALE-X runs at a speed of approximately 20 steps per second on a single RTX 4090 GPU. Additional details on computational resources can be found in Appendix F.

Limitations and Future Works. One limitation is that 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. Moreover, we mention that although the generalizability of WHALE 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

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

[2] 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.

[3] David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018.

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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, 2024.

[10] 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.

[11] 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.

[12] 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.

[13] 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*, 2024.

[14] 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.

[15] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[16] 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 *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[17] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

[18] Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[19] 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.

[20] 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.

[21] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2019.

[22] 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.

[23] 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.

[24] Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022.

[25] Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, Honolulu, HI, 2023.

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

[27] 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.

[28] 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.

[29] 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.

[30] Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, Chi Zhang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

[31] 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*, 2023.

[32] 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.

[33] 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.

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

[35] Nathan Lambert, Kristofer Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models. *arXiv preprint arXiv:2203.09637*, 2022.

[36] Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michel L Littman. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320*, 2019.

[37] Benjamin Eysenbach, Alexander Khazatsky, Sergey Levine, and Russ R Salakhutdinov. Mismatched no more: Joint model-policy optimization for model-based rl. *Advances in Neural Information Processing Systems*, 35:23230–23243, 2022.

[38] 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*, 2024.

[39] 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*, 2024.

[40] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

[42] 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.

[43] 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.

[44] 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.

[45] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

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

[47] 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.

[48] 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.

[49] 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.

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

[51] 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*, pages 586–595, 2018.

[52] 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.

[53] 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.

[54] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[55] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[56] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *8th Annual Conference on Robot Learning*.

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

[58] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

[59] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[60] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023.

[69] 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.

[70] 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.

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

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

[73] 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.

[74] 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.

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

[76] 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.

[77] 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.

[78] 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.

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

[80] 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.

[81] 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.

[82] 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)*, page 7, Paris, France, 2020.

[83] 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.

[84] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023.

[85] 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. 2023.

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

[87] 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*, pages 991–1002. PMLR, 2022.

[88] 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*, 2024.

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

[90] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

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

A Analysis of behavior-conditioning

In this section, we provide some theoretical explanations about why behavior-conditioning mechanism helps mitigate the generalization error caused by the policy divergence. The analysis is mainly adapted from [13].

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.,

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

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 behavior-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 behavior-conditioning has an upper bound:*

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

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:*

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

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 behavior-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 [13], justify the underlying rationale.

B Implementation Details

B.1 Implementation Details of WHALE

Video Tokenizer. In this work, we adopt a tokenizer based on VQ-VAE [46] as the encoder to discretize observations into tokens and train a dynamics model at the token level. 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. Here we show the architecture and training hyperparameters of the video tokenizer as shown in Table 2. We train three different video tokenizers in total.

Table 2: Hyperparameter of video tokenizers.

Component	Parameter	WHALE$_{(64\times64)}$	WHALE$_{(256\times256)}$	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

Behavior-conditioning Model. The behavior-conditioning model comprises a CNN-based visual encoder v_θ, ST-Transformer-based posterior model q_ϕ, prior model p_ψ, and reconstruction model π_ω. Given an input image sequence, v_θ first converts it into tokens by patchifying the images. These tokens are then processed by q_ϕ, p_ψ, and π_ω, which produce the posterior representations z_H, prior representations z_h, and reconstructed actions a_h, respectively.

For behavior embeddings, we employ two-hot encoding due to its strong expressive capacity and stable training process, as noted in [7]. The model architecture and training hyperparameters of the behavior-conditioning model are shown in Table 3. We also train three different behavior embedding models for WHALE-X. Additionally, we also observe overfitting in the behavior-conditioning 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 The key distinction from standard dynamics model learning is that WHALE additionally incorporates a behavior-conditioning 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 behavior-conditioning $z_h = p_\psi(\tau_h)$:

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

Table 4 and Table 5 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 [11].

B.2 Fine-tuning Details of WHALE-X

For fine-tuning all pre-trained models, we first update the video tokenizer for 5000 gradient steps while keeping the encoder network fixed. After that, we update the behavior-conditioning model for 1000 gradient steps, and finally, we update the dynamics model for 5000 gradient steps. For training

Table 3: Hyperparameter of behavior-conditoning models.

Component	Parameter	WHALE$_{(64\times64)}$	WHALE$_{(256\times256)}$	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 4: Model hyperparameter of dynamics models.

Model	#Parameters (dynamics only)	num_layers	num_heads	d_model
WHALE (64)	26M	12	8	512
WHALE (256)	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 5: Trainig 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

models from scratch, the video tokenizer, behavior-conditioning model, and dynamics model are all updated for 10,000 gradient steps.

B.3 Implementation Details of Baselines

Baselines for model evaluation We compare WHALE against several world model learning baselines, including (1) **FitVid** [47], a variational-based world model that can fit large diverse video datasets. (2) **MCVD** [48], a diffusion-based world model that can perform video generation conditioning on different subsets of video frames and actions. (3) **DreamerV3** [26], a recurrent world model that outperforms specialized methods across diverse control tasks. (4) **iVideoGPT** [12], a scalable transformer-based world model that achieved state-of-the-art results in video generation and embodied control tasks.

Specifically, we use the official implementation of VP2 [57] 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 6 and Table 7, respectively.

Table 6: Hyperparameters for DreamerV3.

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 7: Hyperparameters for iVideoGPT.

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

B.4 Implementation Details of Real-world Policies

Action Chunking with Transformer (ACT) [58]. ACT is a generative imitation learning model designed to address the challenges of long-horizon, fine-grained manipulation tasks. We use the official codebase [†]. Our backbone consists of a 4-layer Transformer encoder and a 7-layer Transformer decoder, each employing 8 attention heads and a feedforward dimension of 3,200. The model processes action sequences in chunks of 30 timesteps and utilizes a latent space dimension of 32 for variational inference. We train the model using the AdamW optimizer with a learning rate of 4×10^{-5}, weight decay of 1×10^{-4}, and a batch size of 32. The training procedure runs for 500 epochs to ensure convergence.

Diffusion Policy (DP) [59]. DP is a generative imitation learning approach that formulates action prediction as a conditional denoising diffusion process. We use the official code release [‡], employing a U-Net backbone with downsampling dimensions $[256, 512, 1024]$ and diffusion step embedding dimension of 128. The model is trained for 100 diffusion steps and uses 16 DDIM sampling steps during inference. We train the model using the AdamW optimizer with a learning rate of 1×10^{-4}, a cosine learning rate scheduler, and a batch size of 128 for 300 epochs.

OpenPI (π_0) [60]. π_0 is the state-of-the-art Vision-Language-Action (VLA) model that enables direct policy learning from visual observations and natural language commands, eliminating the need for explicit state estimation while maintaining strong generalization across diverse tasks. We use the official code release [§], processing action sequences in 30-step chunks and maintaining all other

[†] `https://github.com/tonyzhaozh/act`

[‡] `https://github.com/real-stanford/diffusion_policy`

[§] `https://github.com/Physical-Intelligence/openpi`

hyperparameters from the original implementation while applying LoRA fine-tuning to adapt the model to our specific dataset.

B.5 Implementation Details of Evaluation Metrics

B.5.1 Evaluation metrics for model evaluation.

The metrics we use for model evaluation are defined as follows:

Absolute Error is defined as the difference between the value and the estimated value of a policy:

$$\text{AbsErr} = |V^\pi - \hat{V}^\pi|, \tag{6}$$

where V^π is the true value of the policy and $\hat{V}^\pi$ is the estimated value of the policy.

Rank correlation measures the correlation between the ordinal rankings of the value estimates and the true values, which can be written as:

$$\text{RankCorr} = \frac{\text{Cov}(V^\pi_{1:N}, \hat{V}^\pi_{1:N})}{\sigma(V^\pi_{1:N})\sigma(\hat{V}^\pi_{1:N})}, \tag{7}$$

where $1:N$ denotes the indices of the evaluated policies.

Regret@k is the difference between the value of the best policy in the entire set, and the value of the best policy in the top-k set (where estimated values choose the top-k set). It can be defined as:

$$\text{Regret @k} = \max_{i \in 1:N} V^\pi_i - \max_{j \in \text{topk}(1:N)} V^\pi_j, \tag{8}$$

where topk$(1:N)$ denotes the indices of the top K policies as measured by estimated values $\hat{V}^\pi$.

Mean Maximum Rank Violation (MMRV) is a metric that quantifies the worst-case ranking inconsistency between real-world and simulated policy evaluations by averaging the maximum performance-weighted ranking errors across all policies.

$$\text{RankViolation}(i,j) = |R_i - R_j| \cdot \mathbb{1}\left[(R_{\mathcal{S},i} < R_{\mathcal{S},j}) \neq (R_i < R_j)\right] \tag{9}$$

$$\text{MMRV}(R, R_\mathcal{S}) = \frac{1}{N}\sum_{i=1}^{N} \max_{1 \leq j \leq N} \text{RankViolation}(i,j) \tag{10}$$

C Data Preparation

C.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.

C.2 Pre-training Data

Follow [61], our pre-training dataset collection includes 27 datasets, with a total scale of 970k demonstrations, as shown in Table 8.

D Real-world Task Design

D.1 Hardware Setup

Our hardware setup is shown in Figure 7. For the embodiment, we use the ARX5 robotic platform, which is similar to Aloha [53] and includes two master arms and two puppet arms. We only use the right arm in our experiment. For the vision sensor, a Realsense D435i camera is mounted above the desk to capture RGB image observations.

Table 8: WHALE-X Pre-training Dataset Mixture.

WHALE-X Pre-training Dataset Mixture	Percentage
Fractal [62]	12.7%
Kuka [63]	12.7%
Bridge[64, 65]	13.3%
Taco Play [66, 67]	3.0%
Jaco Play [68]	0.4%
Berkeley Cable Routing [69]	0.2%
Roboturk [70]	2.3%
Viola [71]	0.9%
Berkeley Autolab UR5 [72]	1.2%
Toto [73]	2.0%
Language Table [74]	4.4%
Stanford Hydra Dataset [75]	4.4%
Austin Buds Dataset [76]	0.2%
NYU Franka Play Dataset [77]	0.8%
Furniture Bench Dataset [78]	2.4%
UCSD Kitchen Dataset [79]	<0.1%
Austin Sailor Dataset [80]	2.2%
Austin Sirius Dataset [81]	1.7%
DLR EDAN Shared Control [82]	<0.1%
IAMLab CMU Pickup Insert [83]	0.9%
UTAustin Mutex [84]	2.2%
Berkeley Fanuc Manipulation [85]	0.7%
CMU Stretch [86]	0.2%
BC-Z [87]	7.5%
FMB Dataset [88]	7.1%
DobbE [89]	1.4%
DROID [90]	10.0%

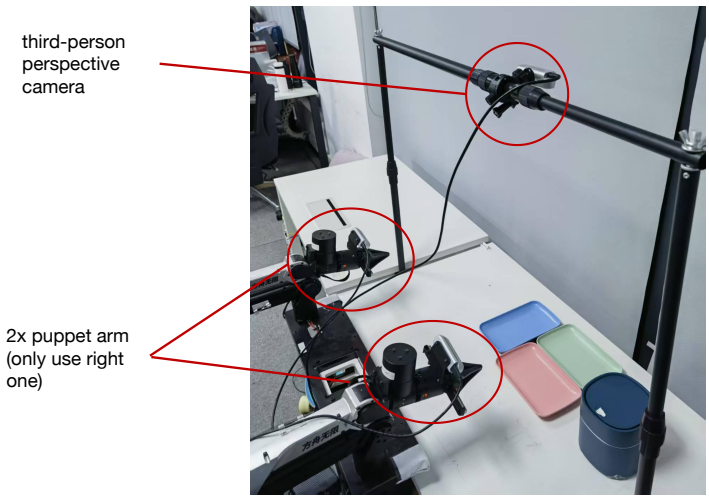


Figure 7: The illustration of our robotics platform used for physical robot evaluation.

D.2 Details of Tasks

The training dataset for policy learning comprises three tasks: **Open Trash Bin**, **Pick & Place Cup**, and **Throw Ball**.

Open Trash Bin: In this task, the robot arm needs to reach the trash bin and press a specific spot on the lid to open it. The robot must accurately reach and press the lid from its initial pose.

Pick & Place Cup: The robot must first reach and pick up a cup placed at a randomly sampled location within a predefined region, and then place it at a specified target location. Both the initial and target positions are sampled from multiple predefined candidate positions, introducing variability in both motion planning and execution.

Throw Ball: This task requires the robot to execute a two-step motion. First, it reaches a randomly placed ball, picks it up, and then moves to a predefined throw zone to release it. This setup evaluates both manipulation precision and dynamic coordination.

To evaluate generalization, we introduce three types of unseen task variations, each designed to challenge the model along a different axis:

Figure 8: The illustration of initial train state

Figure 9: The illustration of initial test state

Visual Generalization: This setting introduces unseen visual distractors, including changes in background and illumination, to evaluate the robustness of the model's visual perception. The visual generalization experiment is conducted based on the Open Trash Bin task. As illustrated in the left figure of Figure 8 and Figure 9, distractor plates may have different colors, and the lighting conditions differ from those seen during training.

Object Generalization: In this variant, we replace objects with alternatives that are visually and physically different from those used during training, while preserving the task semantics. The experiment is conducted based on the Pick & Place Cup task. As showing in the mid figure of Figure 8 and Figure 9, the cup is replaced with one of a different type, color, or shape to test the model's ability to generalize across object instances.

Spatial Generalization: This setting involves perturbing the spatial configuration of the initial and goal object locations. Objects are placed in positions not encountered during training, challenging the policy to generalize to new spatial layouts and reachability conditions. The spatial generalization experiment is conducted based on the Throw Ball task, where the initial position of the ball is significantly different from the training scenarios as illustrated in the right figure of Figure 8 and Figure 9.

D.3 Data Overview

The overall composition of data used in real-world experiments is summarized in Table 9. For each task, we collected 20 teleoperated demonstration trajectories and an additional 30 policy rollout

Table 9: The meta Information of data used in physical robot evaluation.

Entry	Value
# Episodes	330(150 for fine-tuning, 180 for testing)
Average horizon	200
Data Collection 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	Joint angle (qpos)
Action semantics	(q1, q2, q3, q4, q5, q6, the gripper state)
Control frequency	15Hz
Has suboptimal?	Yes(some failure data for fine-tuning)
Has camera calibration?	No

trajectories, which include both successful and failed attempts. 10 rollouts were collected using each of the three policies: ACT, DP, and π_0. This results in a total of 150 training trajectories across all tasks. Importantly, the training data does not include any trajectories from the testing policy checkpoint or from task configurations involving visual, object, or spatial generalization. During evaluation, we used policy checkpoints that were not seen during training to assess WHALE-X's generalization performance under these unseen conditions.

E Additional Experimental Results

E.1 Video Fidelity Results in Real-world Tasks

The results are shown in Table 10. WHALE-X achieves the best image quality with the highest average PSNR (21.95 dB), outperforming all baselines across three tasks (19.36, 20.92, and 21.13 dB), highlighting the benefits of its enhanced architecture and behavior-conditioning.

Table 10: Peak Signal-to-Noise Ratio (PSNR) comparison across different tasks and models.

Model	Open Trash Bin	Pick & Place Cup	Throw Ball	Average
From Scratch	18.38	19.90	19.80	19.36
wo Behavior-Conditioning	21.89	21.11	20.39	21.13
iVideoGPT	16.59	16.33	16.68	16.53
WHALE-X (ours)	**23.02**	**21.66**	**21.17**	**21.95**

E.2 Qualitative Evaluation

Qualitative Evaluation on Simulated Task. Figure 10 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.

Qualitative Evaluation on Open X-Embodiment Dataset. Figure 11 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.

Qualitative Evaluation on Real-world Task Figure 12- 17 show the qualitative evaluation results of WHALE-X on Real-world Tasks. WHALE-X demonstrates strong generalizability in terms of motion, visualization, and task combination.

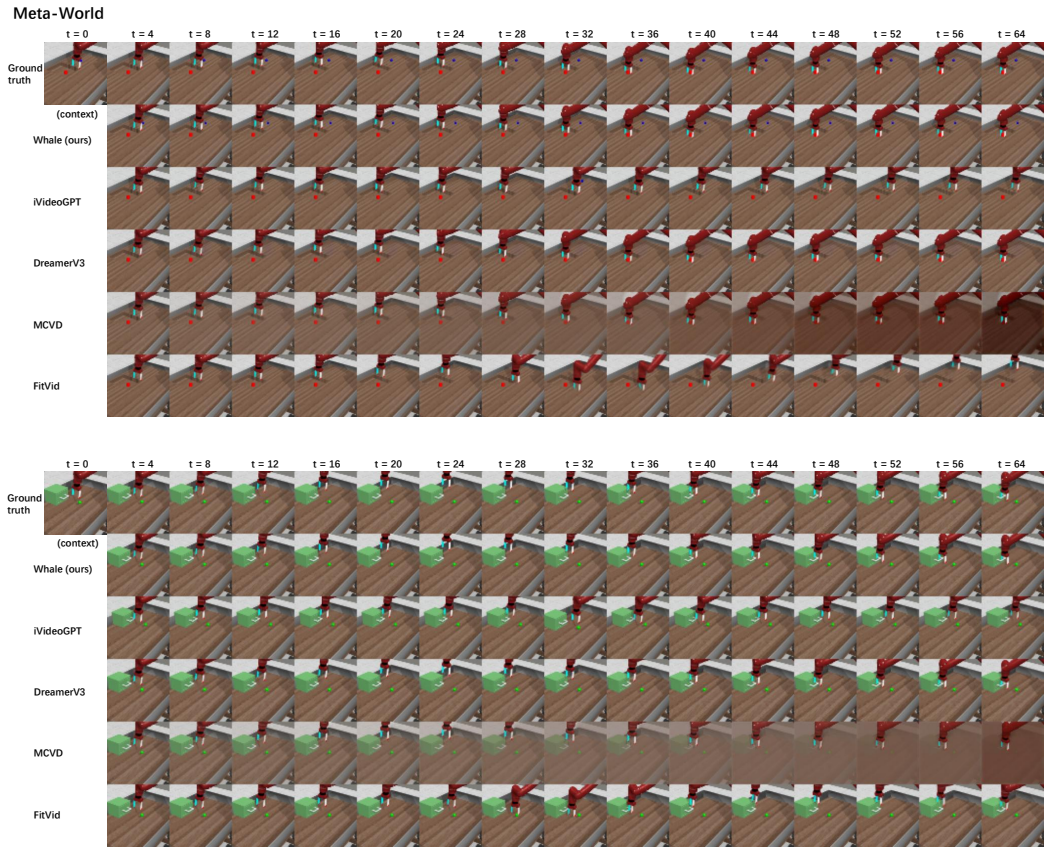


Figure 10: Additional qualitative evaluation on the Meta-World dataset.

E.3 Behavior Embedding Visualization

To verify whether the learned behavior embedding has captured policy modes, we perform t-SNE [91] to visualize the representations corresponding to different tasks and policies. Figure 19a shows that different policies for the same task can be distinguished by the learned behavior embedding. Notably, the embedding of the noisy expert policy appears to be a linear interpolation between the expert policy and the noisy policy, indicating that the behavior-conditioning models the policies reasonably. Figure 19b shows that the expert policies for different tasks can also be distinguished, while Figure 19c shows the random policies for different tasks cannot. This distinction indicates that our learned embedding is more inclined toward policy representation rather than task representation.

F Computational Resources

Pre-training. All experiments are conducted on RTX 4090 GPUs for both training and inference. For simulation tasks, approximately 80 GPU hours are required: 48 GPU hours for tokenizer training, 8 GPU hours for behavior-conditioning model training, and 24 GPU hours for dynamics model training. Pre-training the WHALE-X model demands substantially higher computational resources,

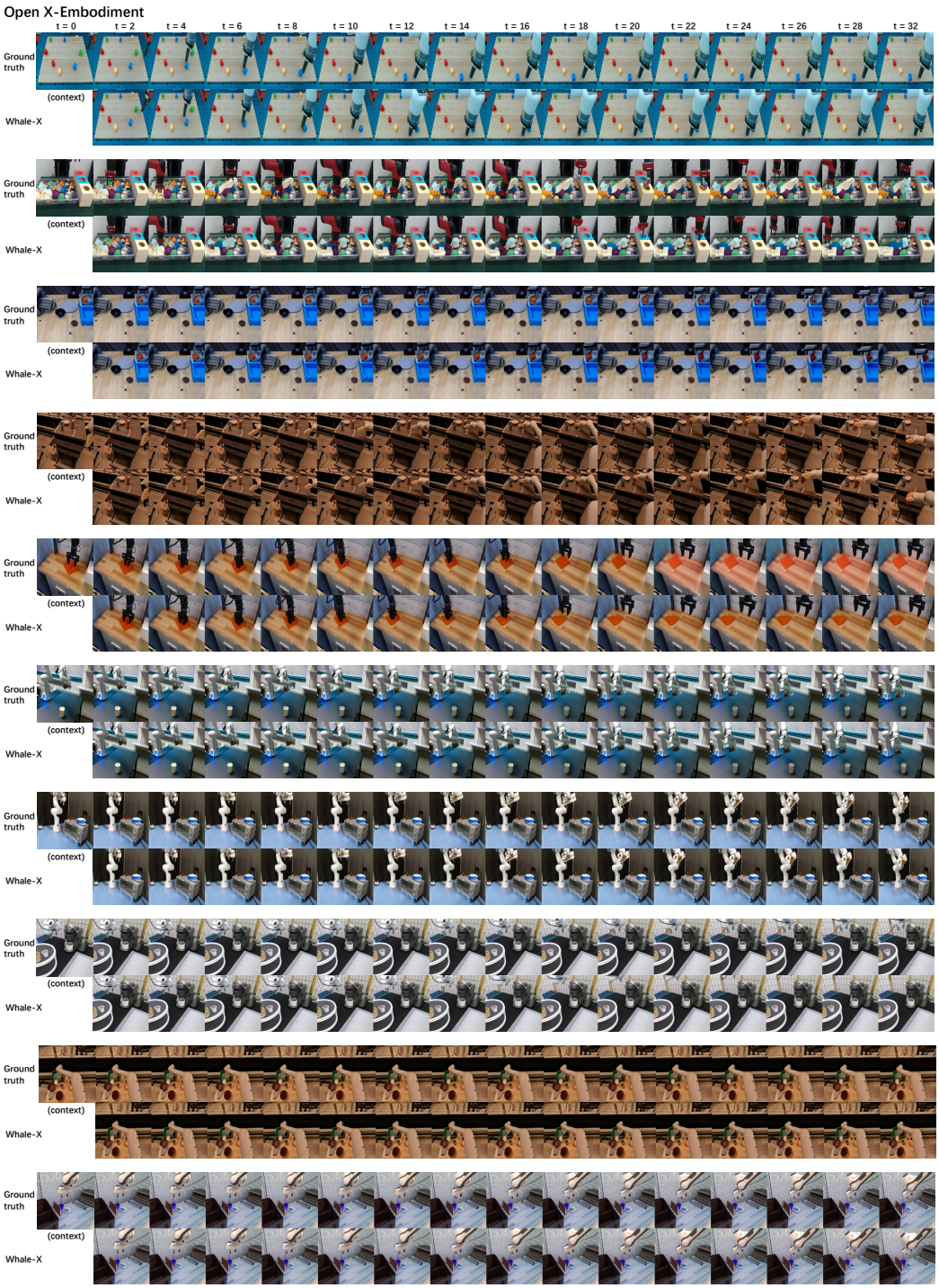


Figure 11: Additional qualitative evaluation on the Open X-Embodiment dataset.

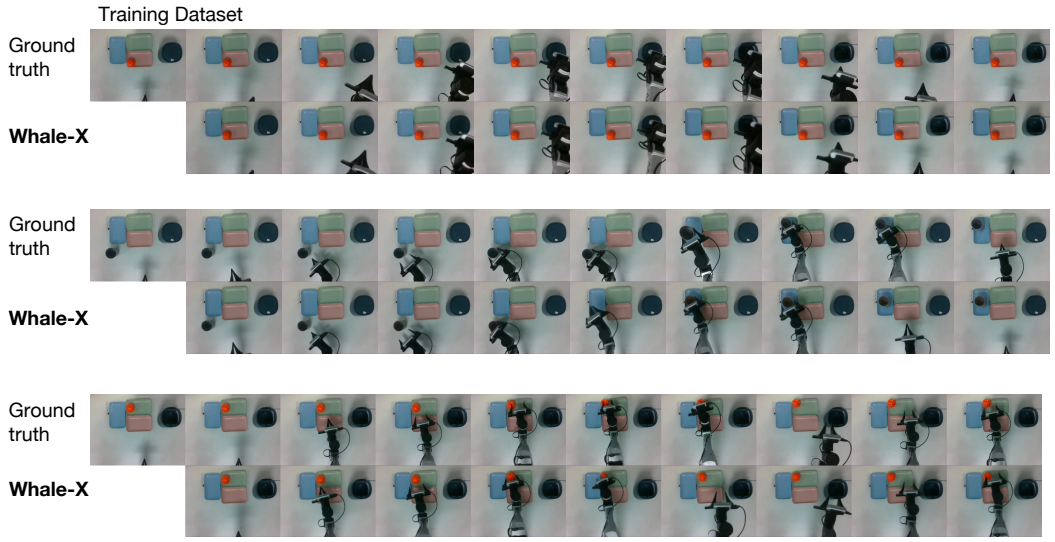


Figure 12: Additional qualitative evaluation on the Real-world tasks. The images show the rollout results of the ACT across three tasks from top to bottom: open trash bin, pick&place cup, and throw ball. For each task, we compare the rollout result in WHALE-X with the corresponding real trajectory. The rollout lengths vary across tasks, with the shortest around 100 timesteps and the longest up to 300.

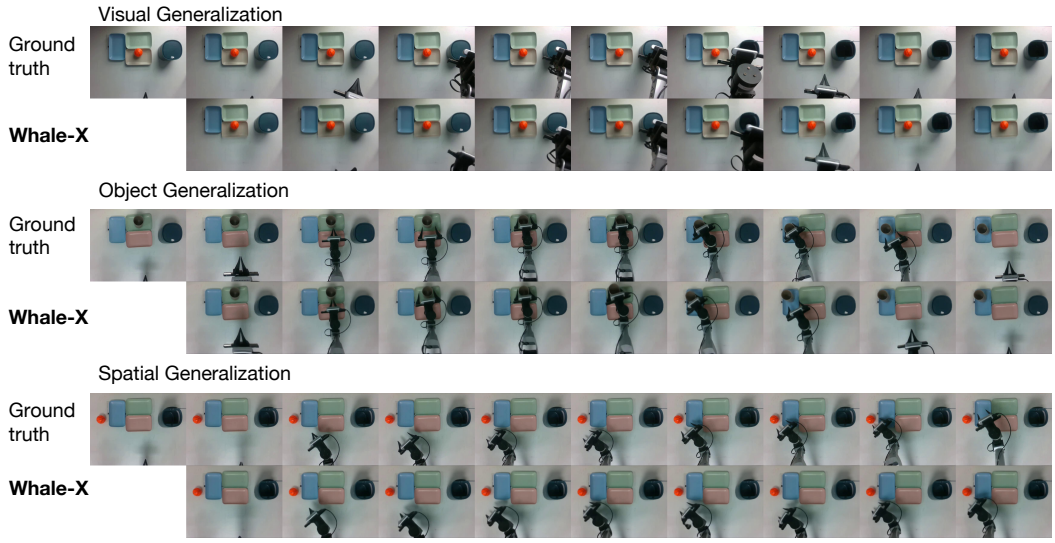


Figure 13: Additional qualitative evaluation on the Real-world tasks. The images illustrate the WHALE-X's evaluation capabilities across three unseen settings. From top to bottom, they correspond to visual generalization, object generalization, and spatial generalization. For each setting, we compare the rollout results of ACT in WHALE-X with ground truth. The rollout lengths range from approximately 100 to 300 timesteps.

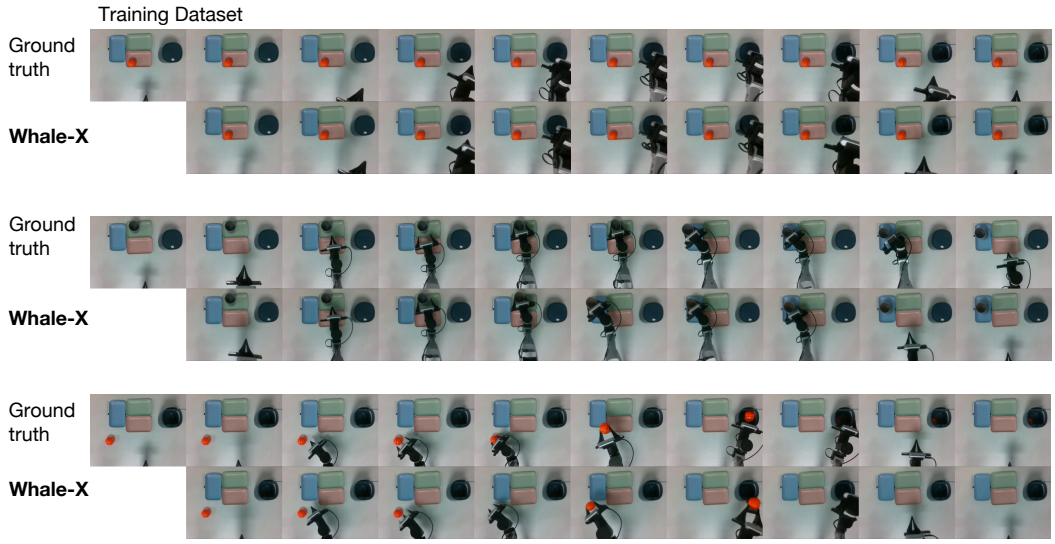


Figure 14: Additional qualitative evaluation on the Real-world tasks. The images show the rollout results of the Diffusion Policy across three tasks from top to bottom: open trash bin, pick&place cup, and throw ball. For each task, we compare the rollout result in WHALE-X with the corresponding real trajectory. The rollout lengths vary across tasks, with the shortest around 100 timesteps and the longest up to 300.

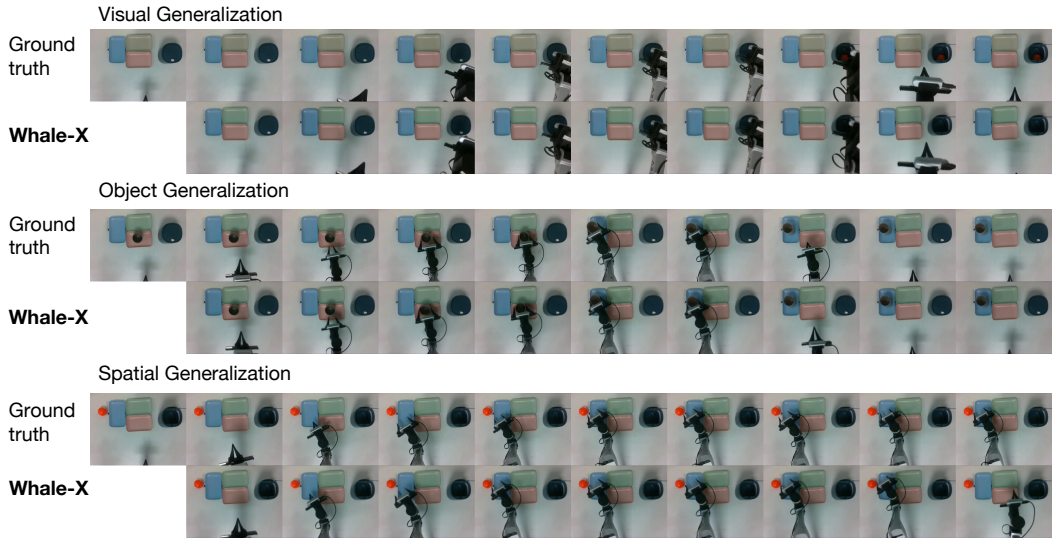


Figure 15: Additional qualitative evaluation on the Real-world tasks. The images illustrate the WHALE-X's evaluation capabilities across three unseen settings. From top to bottom, they correspond to visual generalization, object generalization, and spatial generalization. For each setting, we compare the rollout results of Diffusion Policy in WHALE-X with ground truth. The rollout lengths range from approximately 100 to 300 timesteps.

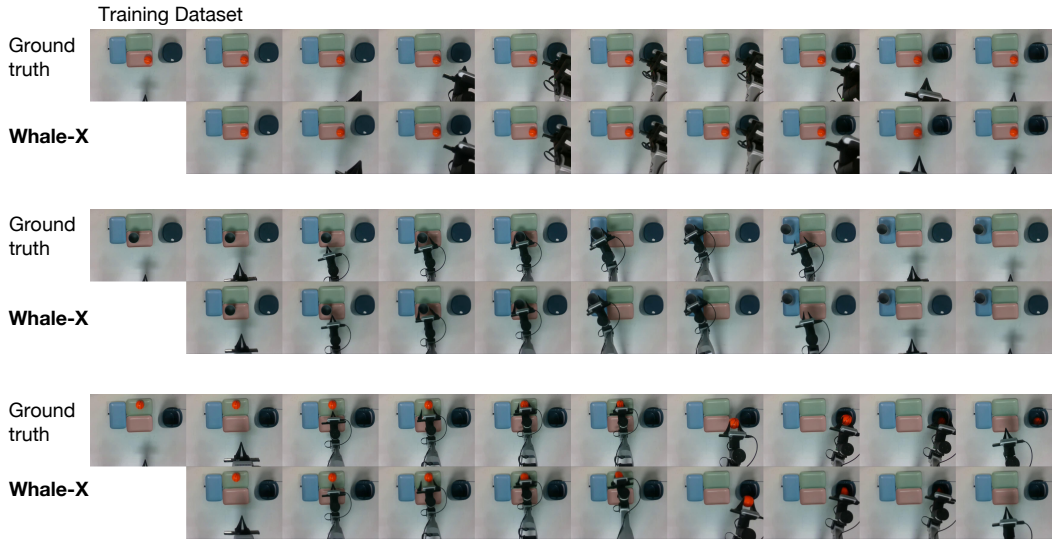


Figure 16: Additional qualitative evaluation on the Real-world tasks. The images show the rollout results of the π_0 across three tasks from top to bottom: open trash bin, pick&place cup, and throw ball. For each task, we compare the rollout result in WHALE-X with the corresponding real trajectory. The rollout lengths vary across tasks, with the shortest around 100 timesteps and the longest up to 300.

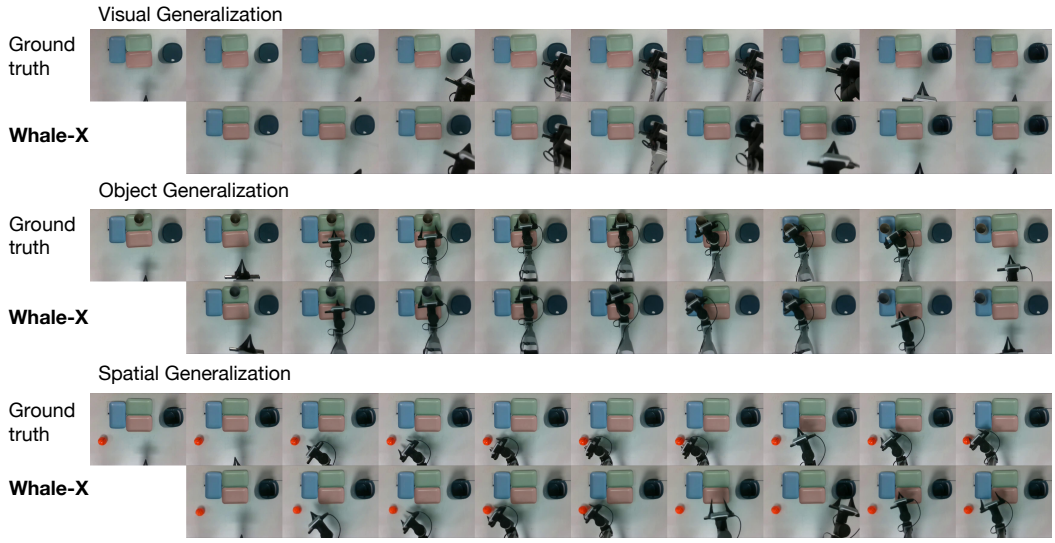


Figure 17: Additional qualitative evaluation on the Real-world tasks. The images illustrate the WHALE-X's evaluation capabilities across three unseen settings. From top to bottom, they correspond to visual generalization, object generalization, and spatial generalization. For each setting, we compare the rollout results of π_0 in WHALE-X with ground truth. The rollout lengths range from approximately 100 to 300 timesteps.

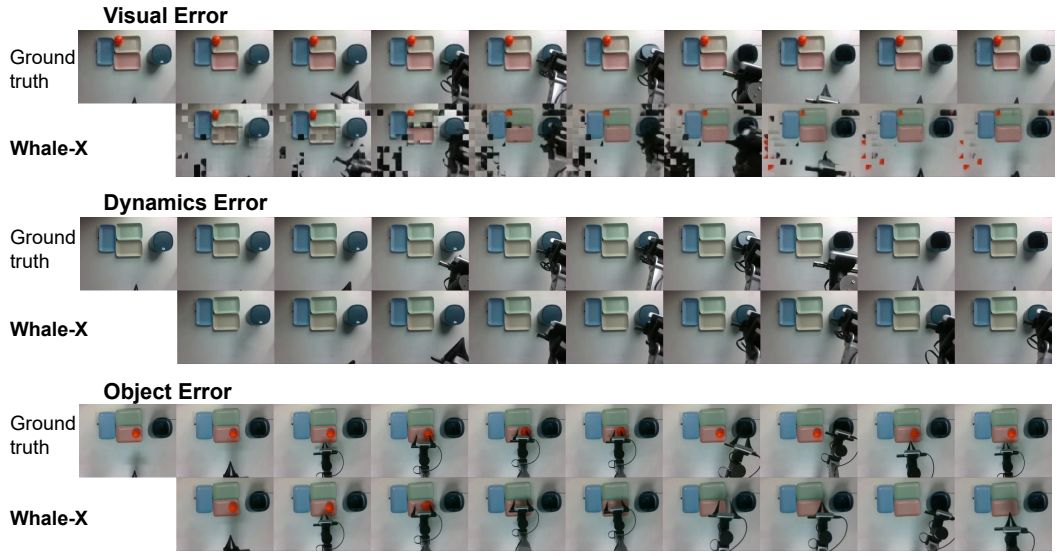
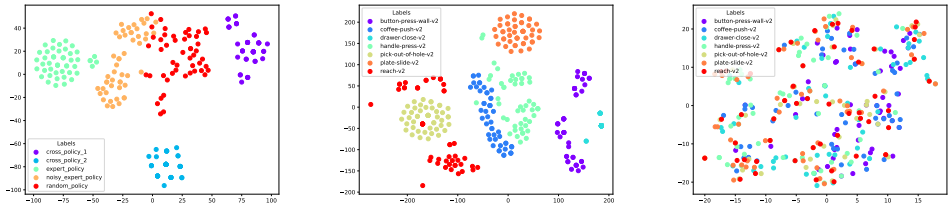


Figure 18: Failure cases on the Real-world tasks. The figure illustrates generalization errors in WHALE-X's rollout, showing three distinct failure cases from top to bottom: visual error, dynamics error and object error, with each case presented alongside ground truth for comparison.



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

Figure 19: The behavior embedding visualization via t-SNE [91]. The different colors denote different policies in the same task (19a) and expert policies in different tasks (19b) or random policies in different tasks (19c).

totaling around 2000 GPU hours: 1152 GPU hours for tokenizer training, 192 GPU hours for behavior-conditioning model training, and 576 GPU hours for dynamics model training.

Fine-tuning. Fine-tuning WHALE-X involves 20,000 gradient steps within our environment, requiring approximately 16 GPU hours in total. This includes 8 GPU hours for fine-tuning the tokenizer, 1 GPU hour for the behavior-conditioning model, and 8 GPU hours for the dynamics model.

Inference. WHALE-X achieves efficient inference performance, benefiting from the parallel decoding structure of the ST-Transformer architecture, reaching an inference speed of 19.8 steps per second.

We further summarize and compare the computational requirements for fine-tuning and inference of the WHALE-X *dynamics model* across different model scales in Table 11.

Table 11: Computational resources for WHALE-X with different model sizes.

Model Size	39M	77M	203M	456M
GPU hours (20000 steps)	~3	~4	~8	~16
Inference Speed (frames/sec)	31.7	27.5	19.8	13.4

G Broader Impacts

This work advances the development of scalable and generalizable world models for embodied decision-making, with potential benefits across a range of applications of robotics applications. By addressing core challenges in generalization, our proposed WHALE framework may enable efficient and more reliable deployment of decision-making agents in real-world settings. Moreover, our large-scale pre-trained model, WHALE-X, highlights the promise of scaling embodied world models through cross-domain datasets, contributing toward more generalizable world models for robotics.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state the contribution and scope of this paper in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations of this work are discussed in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the full set of assumptions and proof in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We present all implementation details for reproducing the main experimental results of this paper in Appendix B.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We submit the code for reproducing the main experimental results in the supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details are described in Section 4.2 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard deviation over 3 random seeds for all experiments in this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We describe the information on the computer resources for running the experiments in Appendix F.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper aims to build a more generalizable world model learning for embodied decision-making and conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts of this work in Appendix G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The models proposed in this paper are only oriented to the domain of embodied decision-making and have no risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should descriviorzbe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original paper that produced the codebase and dataset, and provide the corresponding URLs.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for writing and editing, and does not impact the core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.