LaDi-WM: A Latent Diffusion-based World Model for Predictive Manipulation

Yuhang Huang
National University of
Defense Technology

Jiazhao Zhang
Peking University

Shilong Zou
National University of
Defense Technology

Xinwang Liu
National University of
Defense Technology

Ruizhen Hu[*]
Shenzhen University

Kai Xu[*]
National University of
Defense Technology

Abstract: Predictive manipulation has recently gained considerable attention in the Embodied AI community due to its potential to improve robot policy performance by leveraging predicted states. However, generating accurate future visual states of robot-object interactions from world models remains a well-known challenge, particularly in achieving high-quality pixel-level representations. To this end, we propose LaDi-WM, a world model that predicts the latent space of future states using diffusion modeling. Specifically, LaDi-WM leverages the well-established latent space aligned with pre-trained Visual Foundation Models (VFMs), which comprises both geometric features (DINO-based) and semantic features (CLIP-based). We find that predicting the evolution of the latent space is easier to learn and more generalizable than directly predicting pixel-level images. Building on LaDi-WM, we design a diffusion policy that iteratively refines output actions by incorporating forecasted states, thereby generating more consistent and accurate results. Extensive experiments on both synthetic and real-world benchmarks demonstrate that LaDi-WM significantly enhances policy performance by 27.9% on the LIBERO-LONG benchmark and 20% on the real-world scenario. Furthermore, our world model and policies achieve impressive generalizability in real-world experiments. *The source code will be public at: https://github.com/GuHuangAI/LaDiWM.*

Keywords: World models, Latent diffusion model, Robotic manipulation

1 Introduction

Predictive manipulation [1, 2, 3] has recently gained considerable attention in Embodied AI due to its potential to enhance robot policy performance by leveraging predicted future states. These predictive states, which represent anticipated interactions between robots and objects, have been widely received as beneficial for action planning which is error-prone especially for long-term tasks [4, 5]. To obtain predictive states, most existing methods employ world models [3, 6] to predict visual frames. However, pixel-level visual prediction often exhibits limited performance and generalizability. Another line of research advances world models by predicting latent representations [7, 8, 4], which simplifies model learning and demonstrates strong generalizability. However, the latent space in existing works is typically optimized for image reconstruction, thus failing to capture the geometric and semantic information that is crucial for successful manipulations.

Inspired by latent-space world models, we propose LaDi-WM, which is designed to efficiently model the correlation of geometric and semantic information in the latent space of Visual Foundation Models (VFMs). Our method incorporates both DINO-based [9, 10] and CLIP-based [11, 12, 13] latent representations, capturing geometric and semantic information, respectively. To effectively
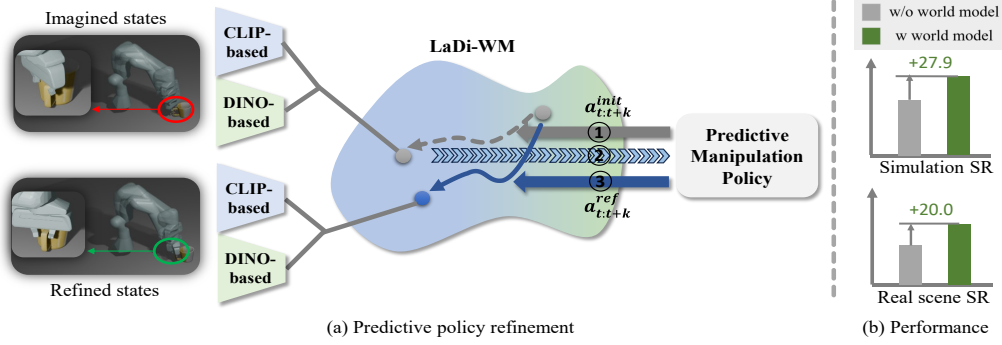
---

[*]Co-corresponding author.

Figure 1: Our method consists of a latent diffusion-based world model and a corresponding predictive manipulation policy. (a): ① Given an initial action sequence generated by the policy, our world model produces imagined future states. ② These imagined states provide efficient guidance for the policy model, ③ yielding refined actions for improved manipulation. (b): Benefiting from our world model, we achieve significant improvements in both simulation and real-world performance.

model their dynamic correlations, we introduce an interactive diffusion process that performs cross-attention between *clean latent components* decomposed from noisy latent representations of geometry and semantics. We find that modeling the dynamic interaction between geometry and semantics with latent space diffusion leads to better alignment between the two, and hence to fast learning convergence and strong cross-scene and cross-task generalibility.

Based on LaDi-WM, we propose an imagination-guided diffusion policy that features *iterative action refinement*. As illustrated in Fig. 1 (a), we train a diffusion policy within the latent space of both DINO and CLIP. This policy is trained to output a sequence of actions at a time. We predict for each action an imagined future state using the pretrained LaDi-WM. These imagined states are then fed into the policy model to produce a refined sequence of actions. Such refinement is guided by multi-step imaginations, yielding higher quality actions. The refinement can be iterated for multiple times until convergence. We provide a discussion on the convergence of the iterative refinement and show that it progressively improves the accuracy and consistency of action predictions.

We have conducted extensive experiments on both synthetic benchmarks and real-world environments. Benefiting from the efficient future guidance by LaDi-WM, our method achieves the highest success rate among the state-of-the-art methods. In particular, our approach significantly improves over the behavior cloning baseline by 27.9% in the synthetic experiment and 20.0% in the real-world environment, demonstrating the superior potential of LaDi-WM in guiding policy learning.

2 Related Work

World models can be regarded as future prediction models that forecast future environmental states based on historical observations and action controls. Current studies on world models can be split into two types, which learn the dynamics in the raw image space and the latent space, respectively. Typical world models [14] employ recurrent neural networks [15, 16] and convolution networks [17] to learn environmental dynamics as future image prediction. Advanced methods [3, 18, 19] explore the use of diffusion models to build world models and learn dynamics in pixel space; however, the computational expense associated with image reconstruction and the overhead of diffusion models limit their applicability. In contrast, the Dreamer series of methods [7, 8, 20] builds upon recurrent neural networks and explores learning imagination in the latent space. Similarly, IRIS [6] and TDMPC2 [5] also learn latent dynamics but utilize transformer and MLP architectures for world modeling. To enhance the representation ability, DINO-WM [4] leverages the pretrained DINO model to extract the latent representation containing rich geometric information. In this paper, we propose the adoption of latent diffusion to establish a more generalizable world model, involving both geometric and semantic information, powered by pretrained foundation models.
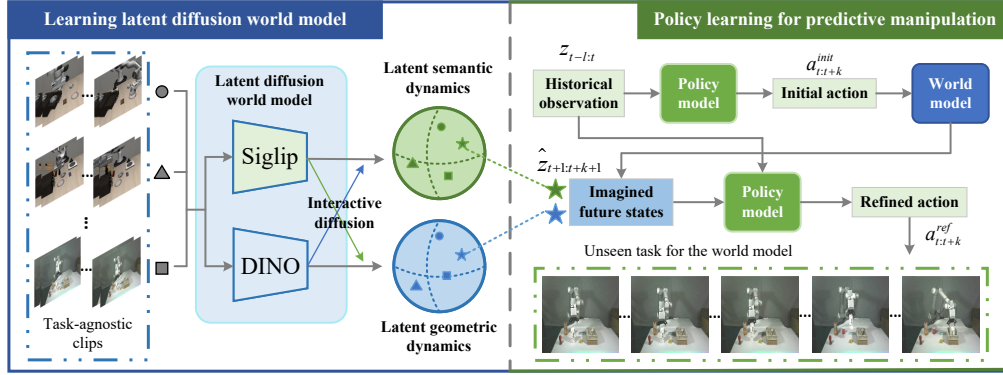
Figure 2: Overall framework of the proposed method. We first train LaDi-WM on task-agnostic clips, which benefits cross-task generalization. The trained world model is used to generate future imagined states, which serve as effective guidance and input to the policy network, improving the manipulation performance significantly.

World models for predictive policy learning. World modes can interact with the policy to generate future dynamics, which can be used to facilitate policy learning. Early world models are usually utilized to support reinforcement learning. IRIS [6] and the Dreamer series [7, 8, 20] propose learning policies in the imagined world to solve simple game tasks. TDMPC series methods [5] and DINO-WM [4] combine world models with model predictive control, optimizing action sequences based on a goal image. However, reinforcement learning [21, 22] requires dense reward, limiting its application in relatively hard tasks such as manipulation. Apart from reinforcement learning, another line of work lets the policy predict the action according to the future imagined observations. AVDC [2] employs a language-driven video prediction model to generate future frames and estimate future optical flow to calculate corresponding robotic transformations. Its performance depends largely on the accuracy of the estimated flow, which is not robust for manipulation tasks. ATM [1] introduces an instruction-conditioned prediction model to predict future point trajectories and utilizes the prediction as the input to the policy network for behavior cloning. The task-specific prediction limits its generalization, and the point representation ignores the detailed information, which may constrain performance improvement. In this paper, our world model is designed to deal with task-agnostic prediction for powerful generalization and learns more detailed information containing both geometry and semantics, resulting in a significant improvement in manipulation success rate.

3 Method

As illustrated in Fig. 2, our method involves two components: a latent diffusion-based world model for imagining the future states, and an imagination-guided policy for action planning, with details provided in Sec. 3.1 and Sec. 3.2, respectively.

3.1 Latent Diffusion-based World Model

We propose to utilize the latent diffusion to construct the world model, aiming to capture the diverse generalizable dynamics of the environments. Benefiting from the powerful ability of diffusion models for modeling multimodal distribution, our world model can handle complex environmental dynamics. Moreover, the diffusion process is conducted in a generalizable latent space that is governed by the pretrained foundation models, containing both geometric and semantic information.

Specifically, we adopt DINO [10] and Siglip [13] to extract both geometric and semantic latent codes, which are crucial for executing robotic tasks. The latent states extracted via the foundation models can be expressed by:

$$\mathbf{z}_t = [\mathbf{z}_t^D; \mathbf{z}_t^S] = [f_{dino}(I_t); f_{sigl}(I_t)], \tag{1}$$

3

where $f_{dino/sigl}$ represents the DINO and Siglip networks, I_t and $\mathbf{z}_t$ are the image observation and latent state at time t.

After obtaining the latent states $\mathbf{z}_t$, the goal of the world model is to predict future latent states given historical latent states $\mathbf{z}_{t-l:t}$ and action control $a_{t:t+k}$:

$$\mathbf{z}_{t+1:t+k+1} \sim p_{\boldsymbol{\theta}}(\mathbf{z}_{t+1:t+k+1}|\mathbf{z}_{t-l:t}, a_{t:t+k}), \tag{2}$$

where p is the transition probability distribution, and $\boldsymbol{\theta}$ is the parameter of the world model. To simplify the following formulation, we replace the future latent states $\mathbf{z}_{t+1:t+k+1}$ with $\mathbf{z}_{t+1:}$.

Interactive diffusion modeling. We propose to utilize the diffusion model to model the transition probability. However, the two types of codes follow different distributions, which implies that it is inappropriate to apply the one diffusion process to learn the two types of dynamics. On the other hand, a separate learning framework may overlook the relationships between high-level semantics and low-level geometry, resulting in inefficient dynamic modeling. To this end, we propose a novel interactive diffusion process that enables two latent codes to interact with each other, leading to more effective dynamic modeling. Moreover, inspired by Huang et al. [23], we decompose the clean latent component from the noisy codes and leverage the clean latent component to conduct interaction.

In more detail, the forward diffusing process consists of an attenuation process of the latent states and a noise-increasing process. To formulate the forward process, we introduce an additional superscript n to represent the diffusion step, i.e., we use $\mathbf{z}_{t+1:,0} = \mathbf{z}_{t+1:}$ and $\mathbf{z}_{t+1:,n}$ to denote the clean latent states and noisy states. Considering a continuous-time diffusion equation with $n \in [0, 1]$, the forward process for the DINO latent code can be represented by:

$$\mathbf{z}_{t+1:,n}^D = \mathbf{z}_{t+1:,0}^D + \int_0^n \boldsymbol{C}_n^D \mathrm{d}n + \sqrt{n}\boldsymbol{\eta}^D, \quad \boldsymbol{\eta}^D \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\mathbf{z}_{t+1:,0}^D + \int_0^n \boldsymbol{C}_n^D \mathrm{d}n$ represents the attenuation process of the latent states while $\sqrt{n}\boldsymbol{\eta}^D$ denotes the growth of the normal gaussian noise, and $\mathbf{z}_{t+1:,n}^D$ is the noisy latent code. Essentially, $\boldsymbol{C}_n^D$ is the gradient of the latent code attenuation, which can be solved by $\mathbf{z}_{t+1:,n}^D + \int_0^1 \boldsymbol{C}_n^D \mathrm{d}n = \mathbf{0}$. According to the derivation in [23], the attenuation process is governed by $\boldsymbol{C}_n^D$, thus we can use $\boldsymbol{C}_n^D$ as the clean component of $\mathbf{z}_{t+1:,n}^D$. Similarly, we can denote the clean component of $\mathbf{z}_{t+1:,n}^S$ as $\boldsymbol{C}_n^S$.

Following the derivation in Huang et al. [23], the conditional probability used in the reversed process can be expressed by:

$$q(\mathbf{z}_{t+1:,n-\Delta n}^D|\mathbf{z}_{t+1:,n}^D, \mathbf{z}_{t+1:,0}^D) = \mathcal{N}(\mathbf{z}_{t+1:,n-\Delta n}^D, \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

$$\boldsymbol{\mu} = \mathbf{z}_{t+1:,n}^D + \Delta t\mathbf{z}_{t+1:,0}^D - \frac{\Delta n}{\sqrt{n}}\boldsymbol{\eta}^D, \tag{4}$$

$$\boldsymbol{\Sigma} = \frac{\Delta n(n - \Delta n)}{n}\mathbf{I}.$$

In the reversed process, the clean latent code $\mathbf{z}_{t+1:,0}^D$ and $\boldsymbol{\eta}$ are not accessible, hence we need to use the parameterized $p_{\boldsymbol{\theta}}(\mathbf{z}_{t+1:,n-\Delta n}^D|\mathbf{z}_{t+1:,n}^D)$ to approximate $q(\mathbf{z}_{t+1:,n-\Delta n}^D|\mathbf{z}_{t+1:,n}^D, \mathbf{z}_{t+1:,0}^D)$. In this way, the denoising network outputs $\mathbf{z}_{t+1:,\boldsymbol{\theta}}^D$ and $\boldsymbol{\eta}_{\boldsymbol{\theta}}^D$ to parameterize $\mathbf{z}_{t+1:,0}^D$ and $\boldsymbol{\eta}^D$ simultaneously. Note that we perform a similar implementation to predict the future Siglip latent code $\mathbf{z}_{t+1:,0}^S$.

To let the two types of latent codes interact with each other in the diffusion process, we first leverage the decomposition network $f_{\boldsymbol{\theta}_1}$ and $f_{\boldsymbol{\theta}_2}$ to estimate the clean components $\boldsymbol{C}_n^D$ and $\boldsymbol{C}_n^S$. The outputs can be expressed via:

$$\boldsymbol{C}_{\boldsymbol{\theta}}^D = f_{\boldsymbol{\theta}_1}(\mathbf{z}_{t+1:,n}^D, n, \mathbf{z}_{t-l:t}^D, a_{t:t+k}),$$
$$\boldsymbol{C}_{\boldsymbol{\theta}}^S = f_{\boldsymbol{\theta}_2}(\mathbf{z}_{t+1:,n}^S, n, \mathbf{z}_{t-l:t}^S, a_{t:t+k}). \tag{5}$$

Here, $\mathbf{z}_{t-l:t}^D$ and $\mathbf{z}_{t-l:t}^S$ are the historical latent codes and $a_{t:t+k}$ is the given action control, which serves as the conditional inputs. Then, $\boldsymbol{C}_{\boldsymbol{\theta}_1}^D$ and $\boldsymbol{C}_{\boldsymbol{\theta}_2}^S$ are fed into the denoising network $f_{\boldsymbol{\theta}_3}$ and $f_{\boldsymbol{\theta}_4}$, generating the parameterized $\mathbf{z}_{t+1:,\boldsymbol{\theta}}^D$ and $\boldsymbol{\eta}^D$ for recovering future latent codes.

$$\mathbf{z}_{t+1:,\boldsymbol{\theta}}^D, \boldsymbol{\eta}_{\boldsymbol{\theta}}^D = f_{\boldsymbol{\theta}_3}(\mathbf{z}_{t+1:,n}^D, n, \boldsymbol{C}_{\boldsymbol{\theta}}^S, a_{t:t+k}),$$
$$\mathbf{z}_{t+1:,\boldsymbol{\theta}}^S, \boldsymbol{\eta}_{\boldsymbol{\theta}}^S = f_{\boldsymbol{\theta}_4}(\mathbf{z}_{t+1:,n}^S, n, \boldsymbol{C}_{\boldsymbol{\theta}}^D, a_{t:t+k}). \tag{6}$$

We adopt the diffusion transformer [24] architecture to construct our world model, and more details can be found in Appendix A.

3.2 Predictive Policy Learning

Based on the world model, we propose a predictive policy learning framework to efficiently solve manipulation tasks by imitation learning. Specifically, given an initial action generated by the policy model, the world model can imagine the future latent states based on historical states. The imagined future states indicate the possible outcomes caused by the initial action, which provides effective guidance for the policy model. Utilizing the imagined guidance as additional input, the policy model can predict the action precisely and refine itself iteratively during testing time:

$$a_{t:t+k} = \pi_{\boldsymbol{\theta}}(\mathbf{z}_{t-l:t}, \mathbf{e}_{t-l:t}, \hat{\mathbf{z}}_{t+1:t+k+1}, \mathcal{T}), \tag{7}$$

where $\mathbf{z}_{t-l:t}$ and $\mathbf{e}_{t-l:t}$ are l-frame historical latent and end-effector states, $\hat{\mathbf{z}}_{t+1:t+k+1}$ is the imagined k-frame future states, and $\mathcal{T}$ is the task instruction.

We employ a diffusion policy with the transformer architecture as our policy model. The diffusion policy has advantages in learning multimodal action distributions and addressing high-dimensional action spaces, which is beneficial for solving diverse manipulation tasks. We build the diffusion process following Huang et al. [23] to improve the efficiency of action prediction. We utilized the different modalities of historical observations as input, including task instruction, diffusion time step, robotic end-effector states, and historical and imagined latent states.

Concretely, given the task instruction $\mathcal{T}$, the diffusion time step n, the historical l frames of latent representation $\mathbf{z}_{t-l:t}$ and end-effector poses $\mathbf{e}_{t-l:t}$, we first utilize the policy model to predict an initial action sequence $\hat{a}_{t:t+k-1}$. The inital action sequence is fed into the world model with $\mathbf{z}_{t-l:t}$, producing the imagined future states $\hat{\mathbf{z}}_{t+1:t+k}$. Then, we utilize $\hat{\mathbf{z}}_{t+1:t+k}$ as additional inputs of the policy model, guiding the prediction of the denoising process. More specifically, we tokenize different modalities of inputs and concatenate them together, and utilize the transformer encoder to fuse the different modality information. The fused features are sent to the transformer decoder, accompanied by the noisy action, obtaining the denoised action. The detailed architecture of the policy model can be found in Appendix A.

Iterative refinement in inference time. During inference, our policy model allows for an iterative refinement process through multi-time exploration within the world model. When an initial action is generated by the policy model, the world model produces corresponding future state predictions. However, since the initial action may be suboptimal, the resulting imagined future states may lack sufficient precision to provide effective guidance. To address this limitation, we implement an iterative correction mechanism in which the policy model continuously refines its actions based on progressively more accurate imagined states generated by the world model. This closed-loop interaction between policy refinement and world model prediction enables increasingly precise action optimization, ultimately leading to more robust and effective policy execution. The detailed inference procedure is provided in Appendix B. We also analyze the convergence of the iterative refinement, which can be found in Appendix E.

4 Experiments

We conduct simulated experiments on two public benchmarks, LIBERO-LONG [25] and CALVIN D-D [26]. The detailed information about the two benchmarks and evaluation metrics can be found in Appendix C. The quantitative comparisons to baselines, ablation studies, and real-world experiments are provided in Sec. 4.1, Sec. 4.2, and Sec. 4.3, respectively.

4.1 Quantitative Results

LIBERO-LONG benchmark results. The results on the Libero benchmark are shown in Tab. 8, where we report detailed performance for each task along with the average success rate, with all

Table 1: **Performance on LIBERO-LONG benchmark.**

Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR↑
DreamerV3 [20]	38.3	21.7	33.3	26.7	30.0	28.3	41.7	31.7	33.3	50.0	33.5
TDMPC2 [5]	45.0	36.7	35.0	31.7	31.7	23.3	36.7	40.0	38.3	51.7	37.0
ATM [1]	46.7	58.3	60.0	31.7	33.3	20.0	43.3	51.7	41.7	53.3	44.0
Seer [27]	71.7	50.0	48.3	**51.7**	66.7	53.3	51.7	45.0	48.3	50.0	53.6
Ours	**88.3**	**68.3**	**63.3**	45.0	**83.3**	**65.0**	**78.3**	**63.3**	**60.0**	**71.7**	**68.7**

Table 2: **Performance on CALVIN D-D benchmark.**

Method	Task completed in a row					Avg.Len.↑
	1	2	3	4	5	
Vanilla BC	81.4	60.8	45.9	32.1	24.2	2.44
DreamerV3 [20]	82.0	63.1	46.6	34.1	25.1	2.51
ATM [1]	83.3	70.9	58.5	46.3	39.1	2.98
Seer [27]	92.2	82.6	71.5	60.6	53.5	3.60
Ours	**92.7**	**83.1**	**72.1**	61.2	54.1	**3.63**

models trained using only 10 demonstrations per task. Notably, our proposed world model achieves significant improvements in policy learning, despite being trained on dynamics unseen during policy training. Compared to the previous SOTA [27], our approach improves the average success rate by 15.1%. This advancement demonstrates the effectiveness of our framework. We provide more results of LIBERO-LONG benchmark in Appendix D.

CALVIN D-D benchmark results. Tab. 2 reports the performance comparison on the CALVIN D-D benchmark. Firstly, compared to vanilla behavior cloning (BC) without the world model, our method significantly improves the Avg.Len. by 1.19. Moreover, our method surpasses other methods in terms of Avg.Len. and success rate, demonstrating the superiority of the predictive policy learning.

Scalability. Figure 3 demonstrates our method's scalability across three dimensions. For world model training, we evaluate performance using 500 to 4,500 demonstrations. Our results show that increasing training data progressively reduces test MSE while simultaneously improving policy learning performance. In policy learning, expanding the training set from 10 demonstrations to complete data yields corresponding improvements in success rate. Notably, our approach consistently outperforms the previous SOTA [27] across all data scales, ultimately achieving a 90.7% average success rate (Avg.SR), i.e., a 3% improvement over Seer's best performance. Furthermore, scaling up the policy model architecture produces measurable gains on both the LIBERO-LONG and CALVIN benchmarks, further validating our method's scalability.
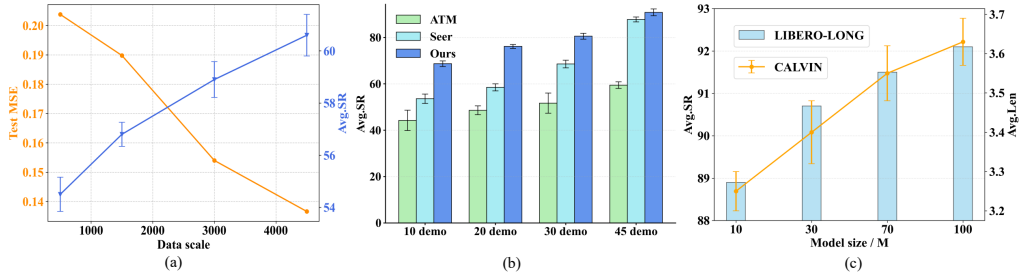

Figure 3: (a) scaling up the training data of the world model; (b) scaling up the training data of the policy model; (c) scaling up the model size of the policy model.

6

Table 3: **Ablation study of different world model architecture**, evaluated with 1-time refinement.

| Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR
without diffusion	78.3	75.0	45.0	40.0	51.7	50.0	45.0	66.7	35.0	35.0	52.1
without siglip	85.0	81.7	**51.7**	45.0	60.0	50.0	51.7	68.3	40.0	40.0	57.3
without interaction	**90.0**	80.0	46.7	48.3	61.7	51.7	**55.0**	70.0	40.0	**46.7**	58.9
without clean interaction	83.3	80.0	43.3	45.0	60.0	53.3	50.0	70.0	41.7	40.0	56.7
Ours	**90.0**	**83.3**	50.0	45.0	**66.7**	**55.0**	**55.0**	**71.7**	**50.0**	40.0	**60.7**
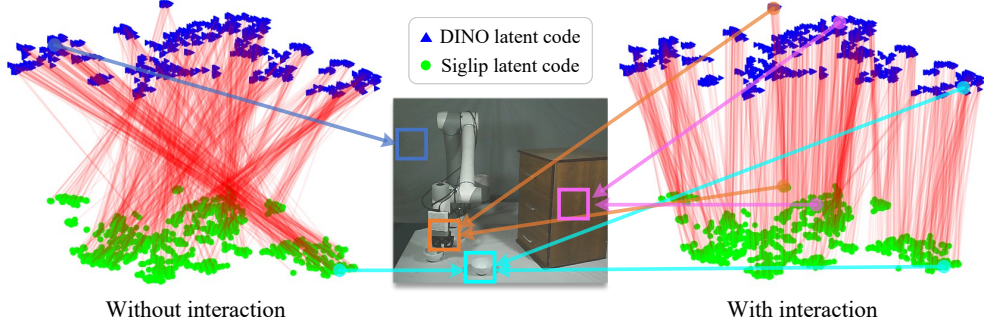
Figure 4: Visual analysis of interactive diffusion. Without the interactive diffusion, the latent codes exhibit connections to distinct regions (left). In contrast, interactive diffusion consistently aligns connections to identical objects (right).

4.2 Ablation studies

We conducted a series of ablation experiments in simulated environments to evaluate the impact of our design choices. Except for necessary explanations, the ablations are conducted on the LIBERO-LONG with 10 training demonstrations for each task.

Ablation study of world model architecture. As shown in the first row of Tab. 3, our method outperforms the transformer-only architecture by 7.8%. As shown in the second row of Tab. 3, incorporating semantic information leads to a 3.4% improvement in average success rate, particularly for Task 1, Task 5, and Task 9, which involve objects with distinct semantic features. As shown in the third row, the interactive diffusion process outperforms the concatenation method by 1.8%, demonstrating its ability to learn more accurate dynamic predictions.

We plot the learned relationship between the two types of dynamics in Fig. 4. According to the attention weight of the transformer layer, the most related features are connected by red lines. The result indicates that our interactive diffusion facilitates the alignment of two latent distributions.

Iterative refinement. We conduct an ablation study on the LIBERO-LONG benchmark to investigate the impact of interaction frequency during iterative refinement. The results are presented in Tab. 4, revealing that increasing the number of iterations (from 0 to 2) yields significant performance gains (27.9% in Avg. SR). However, performance plateaus beyond 2 iterations (3 to 4), suggesting that two rounds of imagination suffice for our model.

Other experiments. We also ablate the effects of latent diffusion and the influence of different imagined future frames. More details can be found in Appendix F.

4.3 Real-world Experiments

We conduct the manipulation experiment in the real world with a 7-DOF CR10 robot arm, as shown in Fig. 5. Same as in the simulated environment, we adopt two cameras to collect images from two views. The action space is the delta of the OSC pose. We define 7 tasks (details can be found in Appendix G) and use the same metric as LIBERO-LONG to evaluate our method. We finetune the

Table 4: **Ablation study of iteration in refinement.**

Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR↑
without imagination	66.7	63.3	31.7	40.0	35.0	41.7	35.0	43.3	25.0	26.7	40.8
1-iter imagination	**90.0**	**83.3**	50.0	45.0	66.7	55.0	55.0	**71.7**	50.0	40.0	60.7 (+19.9)
2-iter imagination	88.3	68.3	**63.3**	45.0	**83.3**	65.0	78.3	63.3	**60.0**	**71.7**	**68.7 (+27.9)**
3-iter imagination	**90.0**	76.7	58.3	**48.3**	78.3	63.3	71.7	66.7	58.3	68.3	67.9 (+27.1)
4-iter imagination	**90.0**	81.7	60.0	46.7	75.0	61.7	71.7	70.0	58.3	66.7	68.2 (+27.4)

Table 5: **Real-world performance of variants.**

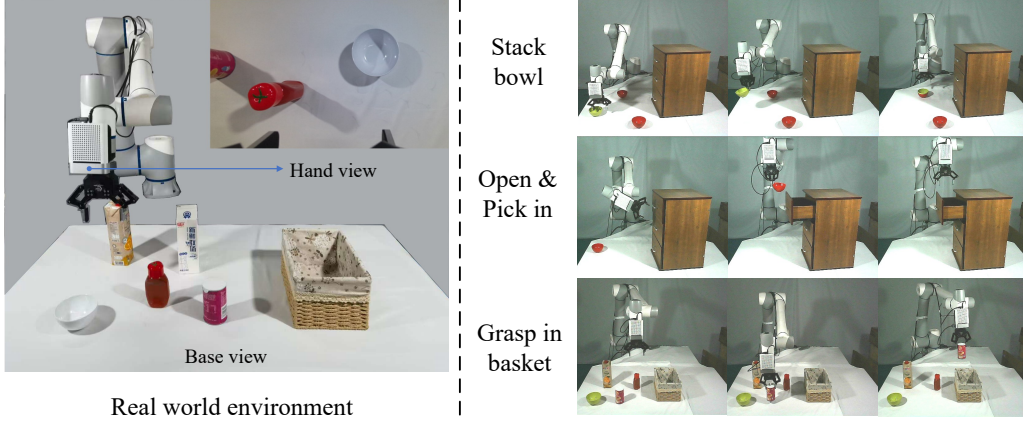| Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Avg.SR
Vanilla BC	35.0	25.0	15.0	45.0	55.0	50.0	55.0	40.0
Ours	**55.0**	**45.0**	**35.0**	**65.0**	**75.0**	**70.0**	**75.0**	**60.0**
Ours (pixel diffusion)	40.0	30.0	25.0	**65.0**	65.0	65.0	70.0	51.4
Ours (without diffusion)	35.0	25.0	25.0	60.0	65.0	65.0	70.0	49.3

Figure 5: Real-world experiments: left shows the setup and right shows several execution examples.

policy model that is trained on the simulated environment using 10 realistic demonstrations per task.

Tab. 5 reports the detailed performance comparison. Compared to the vanilla behavior cloning method, our policy model obtains efficient guidance via imagination of the world model, achieving an improvement of 20% in average success rate. Moreover, we also ablate the latent and the diffusion designs in real-world experiments. The latent design and the diffusion design enhance the Avg.SR by 8.6% and 10.7% respectively.

5   Conclusion

We introduce LaDi-WM, a novel Latent Diffusion World Model that harnesses pre-trained visual foundation models to predict robotic future states in latent space. Our framework establishes distinct latent representations through DINO (geometric features) and SigLip (semantic features), combined with an iterative diffusion process that effectively models complex dynamics to generate high-fidelity future state predictions. The tight integration between LaDi-WM and diffusion policies enables enhanced policy learning through latent-space future state conditioning. Comprehensive evaluations across both synthetic and real-world benchmarks demonstrate LaDi-WM's superior performance in augmenting diffusion-based policy effectiveness. These results demonstrate the significant potential of leveraging established latent representations for developing robust world models, advancing toward robotic systems capable of reliable long-horizon prediction.

6 Limitations

We propose a latent diffusion world model to learn general environmental dynamics, which can provide future imaginations as efficient guidance for solving various manipulation tasks. One limitation of our method lies in the limited scale of training samples. We plan to incorporate more training data [28, 29] from diverse environments and tasks. On the other hand, longer-term future predictions can intuitively provide superior guidance for policy learning of long-horizon tasks, yet current world models face fundamental limitations in generating accurate extended predictions due to compounding error accumulation. To address this challenge, we identify the integration of memory mechanisms for maintaining long-horizon temporal consistency as a promising direction for future research.

7 Acknowledgement

References

[1] C. Wen, X. Lin, J. I. R. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems*, 2024.

[2] P. Ko, J. Mao, Y. Du, S. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations, ICLR*, 2024.

[3] S. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations, ICLR*, 2024.

[4] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[5] N. Hansen, H. Su, and X. Wang. TD-MPC2: scalable, robust world models for continuous control. In *International Conference on Learning Representations, ICLR*, 2024.

[6] V. Micheli, E. Alonso, and F. Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

[7] D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations, ICLR 2020*, 2020.

[8] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations, ICLR*, 2021.

[9] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[10] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

[11] S. Goel, H. Bansal, S. Bhatia, R. Rossi, V. Vinay, and A. Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

[12] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[13] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[14] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[15] J. Schmidhuber. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In *IJCNN international joint conference on neural networks*, pages 253–258. IEEE, 1990.

[16] J. Schmidhuber. Reinforcement learning in markovian and non-markovian environments. *Advances in neural information processing systems*, 3, 1990.

[17] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in neural information processing systems*, 2017.

[18] Z. Ding, A. Zhang, Y. Tian, and Q. Zheng. Diffusion world model: Future modeling beyond step-by-step rollout for offline reinforcement learning. *arXiv preprint arXiv:2402.03570*, 2024.

[19] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. J. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

[20] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[23] Y. Huang, Z. Qin, X. Liu, and K. Xu. Simultaneous image-to-zero and zero-to-noise: Diffusion models with analytical image attenuation. *arXiv preprint arXiv:2306.13720*, 2023.

[24] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[25] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

[27] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, 2025.

[28] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[29] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[30] B. Tang, I. Akinola, J. Xu, B. Wen, A. Handa, K. V. Wyk, D. Fox, G. S. Sukhatme, F. Ramos, and Y. S. Narang. Automate: Specialist and generalist assembly policies over diverse geometries. In D. Kulic, G. Venture, K. E. Bekris, and E. Coronado, editors, *Robotics: Science and Systems*, 2024.

[31] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713, 2024.

Appendix

A Architecture Details

World model. As shown on the left of Fig. 6, there are a noising process and a denoising process. We separately add the Gaussian noise to the DINO and Siglip latent codes in the noising process. For the denoising process, we adopt an 8-layer transformer with a hidden dimension of 384 as the decomposition network. For the denoising network, we utilize a cross-attention layer to let the two types of latent representations interact with each other.

We use the mean square error (MSE) loss to train the world model, and the training objective is formulated by:

$$
\min_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\boldsymbol{\theta}_3,\boldsymbol{\theta}_4} \mathbb{E}_{q(\mathbf{z}_{t+1:,0})}\mathbb{E}_{q(\boldsymbol{\eta})}[\|\mathbf{z}^D_{t+1:,\boldsymbol{\theta}} - \mathbf{z}^D_{t+1:,0}\|^2 + \|\boldsymbol{\eta}^D_{\boldsymbol{\theta}} - \boldsymbol{\eta}^D\|^2
$$
$$
\mathbb{E}_{q(\boldsymbol{\eta})}[\|\mathbf{z}^S_{t+1:,\boldsymbol{\theta}} - \mathbf{z}^S_{t+1:,0}\|^2 + \|\boldsymbol{\eta}^S_{\boldsymbol{\theta}} - \boldsymbol{\eta}^S\|^2 \tag{8}
$$
$$
+ \|\boldsymbol{C}^D_{\boldsymbol{\theta}} - \boldsymbol{C}^D_n\|^2 + \|\boldsymbol{C}^S_{\boldsymbol{\theta}} - \boldsymbol{C}^S_n\|^2]
$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4]$ are the parameters of the diffusion world model.

Policy network. The policy model is depicted in the right of Fig. 6. There are two processes in the diffusion policy: the noising and denoising processes. In the noising stage, we gradually add the Gaussian noise to the ground truth action until it is pure noise; while we leverage the transformer layers to learn the denoising process for recovering the action. We first tokenize the different modalities of inputs, which are fed into the transformer encoder and decoder to obtain the final output. The transformer encoder and decoder contain 6 layers and 4 layers, and the hidden dimension is 256. We utilize the MSE loss between the denoised action and the ground truth action to train the policy model.
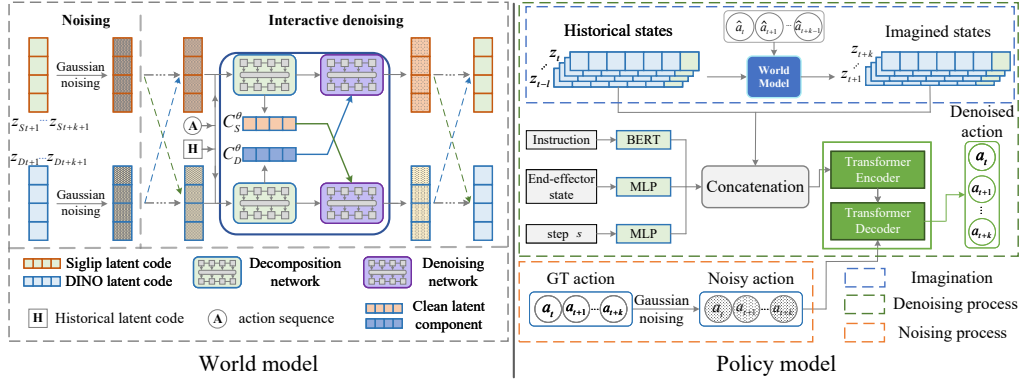

Figure 6: Detailed architecture of the world model and policy model.

Implementation details. The length l of historical videos is fixed at 4 frames, while the length of the imagination is ablated in Sec. 4.2. The world model is trained with a batch size of 4 on an NVIDIA 4090 GPU, which requires approximately three days. We utilize the same device to train the policy model with a batch size of 24, which takes approximately six hours. The raw image resolution is 128×128, and the patch size is 14×14. The action space is seven-dimensional, representing the delta of translation, rotation, and aperture of the end-effector.

B Inference procedure

In the inference stage, we first let the policy model generate an initial action, which is then fed into the world model to obtain the imagination guidance. Since the initial action may be suboptimal, the

---
Algorithm 1 Pseudocode of inference procedure.

1: **Initialize** $t = 0$, environment env, policy network $\pi_{\boldsymbol{\theta}}$, world model $p_{\boldsymbol{\theta}}$, $done = $ False, $max\_step = 600$, instruction $\mathcal{T}$, refine iterations m.
2: **Reset env**: $\mathbf{z}_0, \mathbf{e}_0 = env.reset(), a_{0:k} = \mathbf{0}$;
3: **while** not $done \mid t < max\_step$ **do**

4:      # iterative refinement loop:
5:      **for** $i \leftarrow 1$ to m **do**
6:         # get imagination:
7:         $\hat{\mathbf{z}}_{t+1:t+k+1} = p_{\boldsymbol{\theta}}(\mathbf{z}_{t-l:t}, a_{t:t+k})$;
8:         # get action:
9:         $a_{t:t+k} = \pi_{\boldsymbol{\theta}}(\mathbf{z}_{t-l:t}, \mathbf{e}_{t-l:t},$
10:           $\hat{\mathbf{z}}_{t+1:t+k+1}, \mathcal{T})$;
11:      **end for**
12:      # environment step:
13:      $\mathbf{z}_t, \mathbf{e}_t, done = env.step(a_t)$;
14:      $t = t + 1$;

15: **end while**

---

corresponding imagination guidance may not be accurate enough to provide effective guidance. We adopt an iterative correction mechanism to refine the action based on progressively more accurate imagined states generated by the world model. This closed-loop interaction between policy refinement and world model prediction enables increasingly precise action optimization, leading to more robust and effective policy execution. The pseudocode is shown in Alg. 1.

C Experiment Setup

C.1 Benchmarks and Evaluation Metrics

LIBERO-LONG is designed for language-conditioned long-horizon manipulation tasks and provides various suites with expert human demonstrations. We train the world model on the LIBERO-90 dataset, which includes demonstrations for 90 short-horizon tasks. Note that the world model does not access the tasks of policy learning. Following Tian et al. [27], we report the average success rate (Avg.SR) of the top-3 checkpoints over 20 rollouts for the evaluation. Here is the list of tasks:
• Task1: Turn on the stove and put the moka pot on it.
• Task2: Put the black bowl in the bottom drawer of the cabinet and close it.
• Task3: Put the yellow and white mug in the microwave and close it.
• Task4: Put both moka pots on the stove.
• Task5: Put both the alphabet soup and the cream cheese box in the basket.
• Task6: Put both the alphabet soup and the tomato sauce in the basket.
• Task7: Put both the cream cheese box and the butter in the basket.
• Task8: Put the white mug on the left plate and put the yellow and white mug on the right plate.
• Task9: Put the white mug on the plate and put the chocolate pudding to the right of the plate.
• Task10: Pick up the book and place it in the back compartment of the caddy.

CALVIN D-D is a benchmark focusing on language-conditioned visual robot manipulation, which contains 34 tasks varying in object and scene visual appearance. To make the training data of the world model different from that of the policy model, we selected half of the tasks, including 'push', 'move', 'open', and 'place', to train the world model. The remaining tasks, such as 'lift', 'rotate', and 'stack', are used to evaluate policy learning. Following Tian et al. [27], we report the average success rate and the average length (Avg.Len) of completed sequences.

C.2 Baseline methods

- **DreamerV3** [20] and **TDMPC2** [5] employ learned world models for reinforcement learning and planning. While these methods also operate in latent space, their representations are optimized for image reconstruction or reward prediction, neglecting critical geometric and semantic features essential for manipulation tasks. In contrast, our approach harnesses

Table 6: Performance comparisons on LIBERO-LONG benchmark with all training data.

Method	Avg.SR↑	Put bowl in drawer and close it	Put mug in microwave and close it	Put both pots on stove	Put soup and box in basket
DreamerV3	39.3	58.3	25.0	30.0	35.0
TDMPC2	46.8	61.7	33.3	40.0	36.7
OpenVLA [31]	54.0	45.0	55.0	20.0	35.0
ATM	59.6	83.3	45.0	43.3	60.0
Seer	87.7	**100**	71.7	61.7	91.7
Ours	**90.7**	93.3	**95.0**	**91.7**	**95.0**

Put soup and sauce in basket	Put box and buffer in basket	Put mugs on left and right plates	Put mug on plate and put pudding to right	Pick book and place it in back	Turn on stove and put pot
43.3	33.3	45.0	36.7	30.0	56.7
45.0	46.7	58.3	45.0	40.0	61.7
45.0	**95.0**	40.0	60.0	80.0	65.0
53.3	55.0	63.3	60.0	45.0	88.3
88.3	90.0	**91.7**	85.0	**93.3**	98.3
88.3	90.0	85.0	**85.0**	88.3	95.0

visual foundation models to derive latent representations that explicitly capture both geometric and semantic dynamics, enabling more effective learning for manipulation. In particular, since DreamerV3 and TDMPC2 require dense rewards for world model training, we follow Tang et al. [30] to design appropriate dense rewards for manipulation tasks. Additionally, given that these reinforcement learning-based methods struggle with manipulation tasks, we initialize their policies using behavior cloning.

- **ATM** [1]. ATM employs a language-conditioned trajectory prediction model that provides point-based trajectories as policy inputs. However, this approach faces two key limitations: (1) the language grounding inherently restricts generalization to unseen tasks, and (2) the sparse point representation fails to capture critical observation details. Differently, our world model offers two distinct advantages: (1) it learns from task-agnostic video clips, enabling broader task generalization; (2) it operates in a rich latent space preserving geometric and semantic information.

- **Seer** [27]. Seer represents the previous state-of-the-art method on both the LIBERO-LONG and CALVIN benchmarks. The approach introduces an inverse dynamics model that jointly predicts actions and subsequent frame images. However, this implicit dynamics model cannot directly provide future state guidance for policy learning. In contrast, our world model allows the policy model to obtain future states through imagination, enabling iterative refinement of the action prediction.

D More Quantitative Comparisons

Comparison on LIBERO-LONG with full data. While the main paper demonstrates our method's superiority under limited action-labeled data conditions, we additionally report full-data performance to establish the upper bound of our approach. As shown in Tab. 6, our method achieves the best average success rate of 90.7%, representing a 3% absolute improvement over the previous state-of-the-art.

Cross-scene generalization with world model. To assess the generalization capacity of our world model, we directly transfer the model trained on the LIBERO benchmark to policy learning on the CALVIN benchmark, despite their distinct scene configurations. As demonstrated in the fifth row of Tab. 7, our cross-scene implementation achieves an average sequence length (Avg.Len.) of 3.05,

Table 7: Comparisons against state-of-the-art methods on CALVIN D-D benchmark.

Method	Task completed in a row					
	1	2	3	4	5	Avg.Len.↑
Vanilla BC (LIBERO-LONG)	81.4	60.8	45.9	32.1	24.2	2.44
Vanilla BC (CALVIN)	81.4	60.8	45.9	32.1	24.2	2.44
DreamerV3 [20]	82.0	63.1	46.6	34.1	25.1	2.51
ATM [1]	83.3	70.9	58.5	46.3	39.1	2.98
Seer [27]	92.2	82.6	71.5	60.6	53.5	3.60
Ours (cross-scene)	84.1	72.2	60.3	48.5	40.3	3.05
Ours	**92.7**	**83.1**	**72.1**	**61.2**	**54.1**	**3.63**

surpassing both vanilla behavior cloning (BC) by 0.61 and ATM by 0.07. These results substantiate our world model's ability to generalize across different environmental scenarios.

E Analysis of Iterative Refinement

The proposed latent diffusion-based world model allows the policy model to refine itself iteratively, and the results in the main paper show it can significantly surpass the vanilla BC by 27.9%. Here we conduct a series of experiments to analyze the effect of iterative refinement.

Why does the iterative refinement improve the success rate? Generally speaking, the output of the diffusion policy is not a deterministic value but a stochastic distribution. We claim that the iterative refinement can gradually reduce the entropy of the predicted distribution, thereby aligning predictions more closely with ground truth at each execution step and consequently improving task success rates. To validate this claim, we perform 10 repeated policy evaluations under different iteration settings. We leverage PCA to reduce the action dimension and Kernel Density Estimation to plot 4 distributions. As shown in Fig. 7, as the number of iterations increases, the entropy of the predicted distribution gradually reduces.

Table 8: Performances of iterative refinement trained with different data scales. The results are obtained under 10 repeated evaluations.

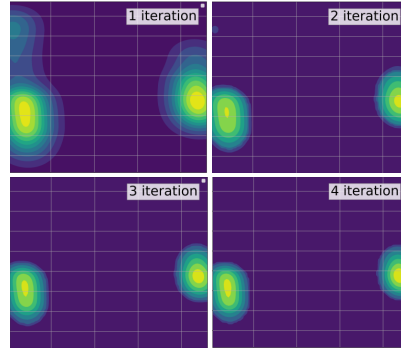| Data scale	10 demo	20 demo	30 demo
1-iter	59.9±7.23	75.1±6.65	80.4±4.68
2-iter	68.1±4.38	77.3±4.63	81.5±3.82
3-iter	68.3±3.25	77.5±4.01	81.5±2.63
4-iter	68.6±2.51	77.5±3.12	81.7±1.93

Figure 7: Predicted action distribution.

Convergence of iterative refinement. We conduct additional experiments with varying numbers of training demonstrations to further verify convergence properties. As shown in Tab. 8, our method consistently achieves convergence within 2 refinement iterations across all experimental settings. Furthermore, the observed variance continuously decreases with increasing iterations, demonstrating that iterative refinement enhances performance through increasingly precise (low-entropy) predictions.

Table 9: Ablation on the effect of the world model.

Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR
BC	66.7	63.3	31.7	40.0	35.0	41.7	35.0	43.3	25.0	26.7	40.8
Copy model	68.3	40.0	51.7	36.7	41.7	26.7	50.0	20.0	30.0	36.7	40.2
Ours	**88.3**	**68.3**	**63.3**	**45.0**	**83.3**	**65.0**	**78.3**	**63.3**	**60.0**	**71.7**	**68.7**

Table 10: Ablation on different diffusion spaces.

Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR
Pixel diffusion	83.3	**78.3**	50.0	35.0	61.7	50.0	45.0	**65.0**	36.7	35.0	54.0
Ours	**88.3**	68.3	**63.3**	**45.0**	**83.3**	**65.0**	**78.3**	63.3	**60.0**	**71.7**	**68.7**

F Ablation Studies

Effect of the world model. We employ an imagination-guided policy based on behavior cloning (BC). To intuitively demonstrate the effect of world model guidance, we compare our method with a vanilla behavior cloning policy that solely relies on historical observations as inputs. Besides, we implement a comparison model that utilizes a copy of the current observation as the future imagination, and we call it the copy model. Our world model, which serves as a simulated environment, can provide future imagined states as efficient guidance. As reported in Tab. 9, the copy model can not bring improvement and is even slightly worse than vanilla behavior cloning. In contrast, our approach improves the vanilla behavior cloning method by 27.9%, highlighting the significant potential of imagination-guided policy learning.

Latent diffusion vs. image pixel diffusion. Unisim [3] proposes a pixel diffusion world model; however, it is not open-source. To ensure a fair comparison, we also employ our diffusion world model in the pixel space, where we retain the network architecture but remove the latent representation extraction stage. Tab. 10 presents a detailed performance comparison. Our method learns the diffusion process in a compact and more generalizable latent space, leveraging the features of pretrained foundation models, which results in robust dynamic predictions for unseen tasks. In contrast, the pixel diffusion world model exhibits inferior performance, with a 14.7% drop in the average success rate.

Table 11: Ablation on imagination frames and denoising steps. The results are obtained with 1-iter refinement.

Frames	1	2	4	6	8	Denoising step	1	2	5	10	30
Avg.SR	44.9	49.6	56.3	60.6	60.0	Avg.SR	60.0	60.6	60.5	60.3	60.6

Effect of the imagined future frames. The imagination-guided policy network takes the future latent states generated by the proposed world model as inputs. We conduct a series of experiments to investigate the impact of varying numbers of imagined latent state frames, and the results are presented in Tab. 11. As the number of frames increases, the average success rate gradually improves, peaking at 6 frames. We hypothesize that excessively long future imagination may become unreliable due to compounding errors.

Effect of different denoising steps. For the policy network, we utilize ADM [23] as our diffusion algorithm, which introduces an analytical attenuation process to speed up the denoising process. We report the results of different denoising steps in Tab. 11. Benefiting from the fast diffusion algorithm, our policy network can achieve a convergent performance with only 2 denoising steps.

Effect of different robotic states. There are two common options for the robotic states as inputs to the policy network: joint states and end-effector states (EE states). As shown in Tab. 12, the use of

Table 12: Ablation on different robotic states as the inputs of the policy network. The results are obtained with 1-iter refinement.

Method	Task1	Task2	Task3	Task4	Task5	Task6	Task7	Task8	Task9	Task10	Avg.SR
EE states	75.0	61.7	50.0	43.3	55.0	51.7	50.0	65.0	35.0	30.0	51.7
joint states	90.0	83.3	50.0	45.0	66.7	55.0	55.0	71.7	50.0	40.0	60.7

joint states outperforms EE states by 9.0%, suggesting that joint states are more suitable for solving manipulation tasks.

G   More Details of Real-world experiments

Task definition. We define 7 manipulation tasks in the real scene, in order: 'stack the middle bowl on the back bowl', 'put the bowl inside the middle drawer of the cabinet and close it', 'open the top drawer and put the bowl inside', 'pick up the milk and place it in the basket', 'pick up the cans and place it in the basket', 'pick up the orange juice and place it in the basket', 'pick up the ketchup and place it in the basket'.

More visual results of manipulation trajectory. We depict more visual trajectories of the real-world manipulation results in Fig. 8 and Fig. 9. Benefiting from the powerful dynamic modeling ability of the world model, our method can deal with complex lighting changes.
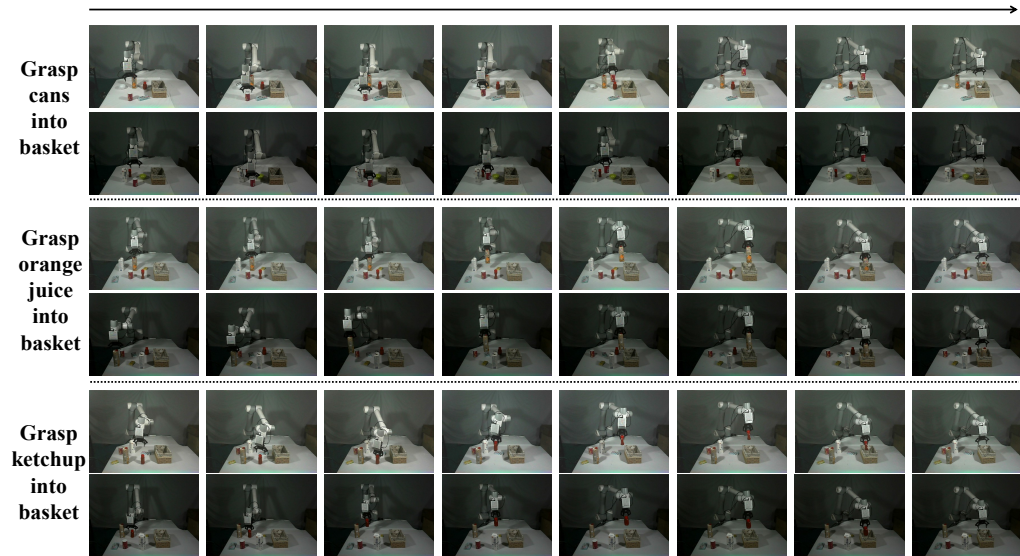
Figure 8: More visual results of real-world manipulation.

Figure 9: More visual results of real-world manipulation.