

Predicting Consequences and Reinforcing Navigation Policies with Latent World Models

Zengmao Wang^{1,2} , Wei Gao^{2,3*} , and Shuhan Shen^{2,3} 

¹ School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences, Beijing 101408, China

² Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

³ School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. World models enable agents to reason about future outcomes and learn policies from their knowledge of state transition, but existing approaches primarily focus on reconstructing future observations or features, which introduces unnecessary complexity and limits their effectiveness for decision making. In this work, we propose a compatibility prediction Latent World Model (LWM) for robot navigation that predicts action-conditioned latent feature compatibility rather than reconstructing observations. Our key insight is that spatial proximity correlates with latent feature similarity, enabling action consequences to be evaluated directly in latent space. To support counterfactual training, our model leverages action sequences sampled across trajectories and learns to predict which sequences lead closer to the goal. Furthermore, we demonstrate how the learned world model can supervise policy learning from unlabeled video data and further improve policies through reinforcement learning entirely within the world model. This imagination-driven framework eliminates the need for action annotations and additional environment interaction. Extensive experiments on multiple real-world robot navigation datasets show that our approach significantly outperforms prior world model and imitation learning methods in prediction accuracy, policy learning, and real-world navigation performance. The code, pretrained models, and additional materials are available at <https://wzm206.github.io/latent-world-model-nav/>.

Keywords: World model · Robot learning · Visual navigation

1 Introduction

Autonomous navigation requires robots to reason about the consequences of their actions before executing them in the real world. World models have emerged as a promising paradigm for enabling such predictive reasoning by learning environment dynamics from offline data [13]. By simulating future outcomes internally, world models allow agents to plan [2], evaluate [25], and improve policies [41]

* Corresponding author.

without costly real-world interaction. This paradigm is particularly attractive for robotics, where data collection is expensive and safety constraints limit online exploration.

However, existing world models suffer from a fundamental limitation: they are typically trained to reconstruct future observations conditioned strictly on the factual actions executed in the dataset [2, 15]. When we use the world model for planning or decision-making, the model needs to infer situations that have not actually occurred, also known as counterfactual reasoning. This creates a critical misalignment: while planning requires counterfactual reasoning to evaluate alternative actions, training relies solely on ground-truth trajectories. Furthermore, optimizing for pixel or latent reconstruction introduces unnecessary complexity, compounding errors, and representations that are misaligned with decision-making. We argue that world models should focus on predicting action consequences rather than generating exact observations.

In this work, we propose a fundamentally different perspective: instead of predicting future observations, we train a world model to predict the compatibility between action consequences and future latent state which leads counterfactual reasoning into the training phase. Our approach is motivated by a simple yet powerful observation: in navigation tasks, spatial proximity strongly correlates with visual feature similarity. Therefore, rather than reconstructing exact future observations, it is sufficient to predict the compatibility.

To achieve this counterfactual training, we introduce an imagination-based world model that leverages action sequences from different trajectories to simulate counterfactual futures in latent space. Given a current observation, our model predicts latent state features resulting from multiple candidate action sequences and learns to distinguish which sequences lead to states closer to the ground-truth future. Crucially, simulating these counterfactual scenarios during training significantly maximizes data utilization efficiency because collecting real-world interaction data is notoriously expensive and challenging in robotics field. This formulation enables the model to learn robust, decision-centric representations without requiring pixel-level prediction. Therefore, we achieve training world models in counterfactual imagination.

Beyond world model learning, we demonstrate how the learned model can be used to train and improve navigation policies from unlabeled video data. We first use the world model to label candidate action sequences according to their predicted similarity to goal states, enabling supervised policy learning without ground-truth actions. We then further improve the policy using reinforcement learning entirely within the learned world model. A well-trained world model can naturally serve as a reward model. This imagination-driven reinforcement learning enables policy improvement without additional environment interaction.

We evaluate our approach in real environments and three robot navigation datasets. Our method demonstrates significant improvements over prior world model and navigation methods. Our results show that compatibility prediction provides a powerful and efficient alternative to generative world modeling, en-

abling effective planning and policy learning from offline data. We summarize our contributions as follows.

- A new world model formulation based on latent compatibility prediction. We propose to predict action-conditioned latent feature compatibility rather than reconstructing future observations, enabling efficient and decision-oriented world modeling.
- A counterfactual imagination framework using cross-trajectory action sequences. Our method leverages action sequences from different trajectories to simulate alternative futures, enabling counterfactual reasoning from purely offline data.
- A unified framework for policy learning and reinforcement entirely within the world model. We show how unlabeled video data can be used to supervise policy learning and further improve policies via imagination-based reinforcement learning.
- State-of-the-art performance in real-world environment and robot navigation benchmarks.

2 Related Work

Driven by advancements in large language models [35], video generation [18], and embodied intelligence [7], world models have garnered significant attention [5]. While the concept continues to evolve, it primarily serves two functions: learning internal representations to understand the environment, and utilizing predictive modeling to guide decision-making for external actions.

2.1 Building a World Model

Numerous efforts focus on constructing world models. Video generation-based approaches [1, 9, 23, 36] are intuitive and demonstrate remarkable physical consistency—such as understanding gravity and collisions—highlighting their potential as implicit world models [8, 19, 20, 28]. These methods typically condition future frame prediction on current observations and actions. For instance, LingBot-World [34] employs a virtual rendering pipeline for interactive data generation, NWM [2] leverages large-scale real-world datasets to generate diverse action-conditioned videos, and 3D-VLA [43] extends this paradigm to 3D point clouds.

However, pixel-level video generation suffers from slow inference speeds and inconsistent physics. Dino-WM [44] addresses this by shifting to feature generation using a Dino-v2 encoder [27], though its primary objective remains reconstruction. In contrast, our method avoids pixel-space generation entirely, preventing the model from fixating on irrelevant details. By optimizing to predict the consequences of actions in the latent space rather than reconstructing features, our approach eliminates the need for iterative diffusion, enabling rapid prediction and decision-making.

2.2 Learning in the World Model

Ultimately, world models aim to facilitate decision-making and learning for agents. Learning paradigms within these models generally fall into two categories.

Joint Training of World and Policy Models Akin to model-based reinforcement learning [13, 14, 16, 32], these methods jointly optimize both models. PlaNet [16] introduces the RSSM architecture to enable dynamic learning in a compact latent space. Dreamer [15] and DreamerV2 [17] learn latent dynamics from images and optimize policies via imagined trajectories, excelling in long-horizon tasks. PWM [10] facilitates first-order policy optimization within regularized world models. However, these methods often require continuous agent-environment interaction, confining them to simulated environments and limiting real-world practicality. While our method also leverages latent state representations, it can be deployed directly in the real world without requiring online interaction.

Utilizing Frozen World Models as Simulators Alternatively, frozen world models can serve as realistic simulators. DreamZero [41] learns physical dynamics via joint video-action prediction for real-time control. World-env [39] replaces physical interactions with a virtual simulator for RL training. RWM [24] uses a dual-autoregressive model for robust sim-to-real transfer, and WorldVLA [3] unifies VLA and world models. These approaches typically train policies on synthesized pixel-level images, which is significantly more complex than predicting feature-space similarities. Furthermore, the generalization of video generation remains problematic [20]. For instance, autonomous driving models struggle to imagine out-of-distribution scenarios like driving on sidewalks due to a lack of training data. Conversely, our cross-trajectory imagination approach can effectively evaluate such deviations, keeping the agent safely aligned with the goal.

3 Method

We propose a compatibility prediction Latent World Model (LWM) that learns action-conditioned latent dynamics by predicting feature compatibility rather than reconstructing future observations. As shown in Fig. 1, our framework enables counterfactual reasoning from offline trajectories to alleviate the challenges of data annotation and collection. Our framework supports policy learning and reinforcement entirely within LMW. We will elaborate on our method in detail below.

3.1 General Problem Formulation

We consider a dataset of offline trajectories:

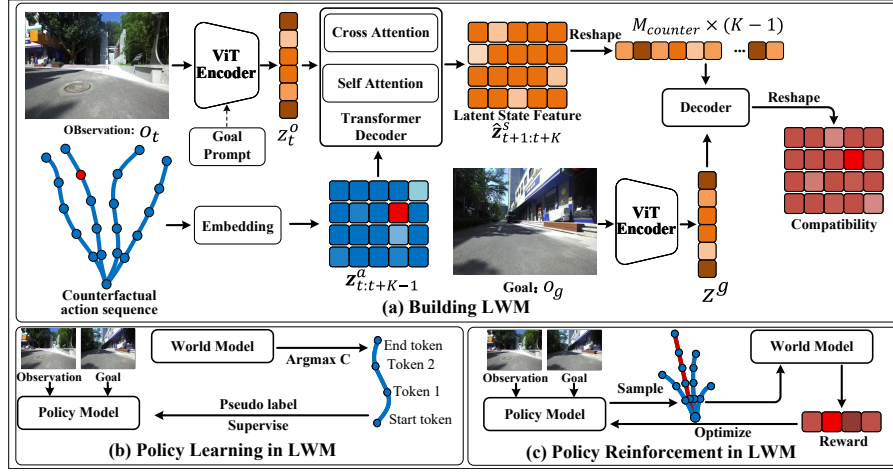


Fig. 1: The overview of LWM. (a) When training the world model, we need to predict the consequences of different action sequences and obtain compatibility with the goal. (b) LWM can provide pseudo labels for the policy model. (c) The policy model can be reinforced in LWM.

$$\mathcal{D} = \{\tau_i\}_{i=1}^N, \quad \tau_i = \{(o_1, a_1), (o_2, a_2), \dots, (o_T, a_T)\} \quad (1)$$

where $o_t \in \mathbb{R}^{H \times W \times 3}$ is the visual observation and $a_t \in \mathbb{R}^{d_a}$ is the robot action at timestep t . In visual navigation tasks, $d_a = 3$ represents the coordinates of motion (x, y, θ) .

Our goal is to learn a world model WM that can evaluate the consequences of arbitrary action sequences starting from a given observation. Within the world model, it can be divided into two modules: the prediction module and the discrimination module. In the prediction module, the world model predicts the state features after executing the action. In the discrimination module, the world model determines the compatibility between the predicted features and the goal.

$$\hat{\mathbf{z}}_{t+1:t+K} = \text{WM}_{\text{predict}}(o_t, A_{t:t+K-1}), \quad C_{t+1:t+K} = \text{WM}_{\text{discriminative}}(\hat{\mathbf{z}}_{t+1:t+K}, o_g) \quad (2)$$

where o_t is the observation at time t , $A_{t:t+K-1} = \{a_t, a_{t+1}, \dots, a_{t+K-1}\}$ is a future action sequence, $\hat{\mathbf{z}}_{t+1:t+K}$ is the predicted features and C is the compatibility.

In video generation world models, $\hat{\mathbf{z}}_{t+1:t+K}$ in Eq.(2) could be considered as images generated based on observation and action sequences. C in Eq.(2) is obtained through pre-trained feature loss, such as comparing the LPIPS [42] loss between the generated image and the goal image. Unlike these works, we do not attempt to reconstruct future observations. Instead, we bring the entire process into the latent space for better decision-making and planning.

3.2 Latent World Model

Latent Observation and Action Encoding We first map observations into a latent feature space using a visual encoder:

$$z_t^o = E_o(o_t), \quad z_t \in \mathbb{R}^{N_p \times D} \quad (3)$$

where E_o is a Vision Transformer [6] encoder. N_p represents the number of patches in the images. Similarly, we encode each action into a latent token:

$$\mathbf{z}_{t:t+K-1}^a = E_a(A_{t:t+K-1}), \quad \mathbf{z}_{t:t+K-1}^a \in \mathbb{R}^{(K-1) \times D} \quad (4)$$

Our world model predicts latent state features conditioned on the current state and an action sequence. We implement the world model prediction using a standard causal Transformer decoder:

$$\hat{\mathbf{z}}_{t+1:t+K}^s = \text{Decoder}(z_t^o, \mathbf{z}_{t:t+K-1}^a) \quad (5)$$

Each prediction follows autoregressive causal masking. For each action token, it can only see its previous actions. At this point, we can predict $(K-1)$ future features. For previous methods, they often predicted pixels or features and optimized neural networks using reconstruction loss: $Loss = MSE(\hat{\mathbf{z}}, \mathbf{z})$. However, they did not take into account the assistance of other action sequences in the dataset. Although we don't know what the observations of performing other actions look like, we can estimate their compatibility with the goal images.

Compatibility Learning Objective A key limitation of offline data is that each observation is paired with only one action sequence. To enable counterfactual reasoning, we construct alternative action sequences by sampling from other trajectories. Given a state z_t^o , we sample M_{counter} candidate action sequences:

$$\mathcal{A} = \{A^{(1)}, A^{(2)}, \dots, A^{(M_{\text{counter}})}\} \quad (6)$$

The world model predicts corresponding future latent state features:

$$\hat{\mathbf{Z}} = \{\hat{\mathbf{z}}^{s(1)}, \hat{\mathbf{z}}^{s(2)}, \dots, \hat{\mathbf{z}}^{s(M_{\text{counter}})}\}, \quad \hat{\mathbf{z}}_{t+1:t+K}^{s(i)} = \text{Decoder}(z_t^o, E_a(A^{(i)})) \quad (7)$$

This allows the model to learn action-conditioned compatibility across diverse trajectories, enabling counterfactual reasoning without environment interaction. Our method relies on a simple and reasonable assumption: features $\mathbf{z}$ with closer spatial distances often more similar in the latent space. For simplicity, we consider training with only one random goal image $z^g = E_o(o_g)$ in ground truth action sequence $\{a_t, a_{t+1}, \dots, a_{t+K_{gt}}\}$. It is worth noting that the selection of goal image during training is random and K_{gt} is smaller than K . Therefore, for each $\hat{\mathbf{z}}_j^{s(i)}$, we can calculate a compatibility with the ground truth goal image. For M_{counter} action sequences, we can obtain $M_{\text{counter}} \times (K-1)$ compatibility score:

$$C = \left\{ S(\hat{z}_j^{s(i)}, z^g) \right\}_{j=1,2,\dots,K-1}^{i=1,2,\dots,M_{\text{counter}}}, \quad C \in \mathbb{R}^{M_{\text{counter}} \times (K-1)} \quad (8)$$

where S is a neural network used for compatibility prediction. Because the state features are in the latent space, we still choose standard Transformer Decoder with a linear output layer instead of LPIPS [42]. Among C_{ij} , the score corresponding to ground truth action sequence $\{a_t, a_{t+1}, \dots, a_{t+K_{gt}}\}$ should be the highest. This naturally reminds us of classification objectives similar to contrastive learning. Specifically, we can consider one result C_{gt} obtained from ground truth action sequences as positive samples and $M_{\text{counter}} \times (K-1) - 1$ results obtained from counterfactual action sequences as negative samples. In this way, we can use Info-NCE loss [12] to optimize the world model.

$$\mathcal{L} = -\log \frac{\exp C_{gt}}{\sum_{i=1}^{M_{\text{counter}}} \sum_{j=1}^{K-1} \exp C_{ij}} \quad (9)$$

However, we found that although this approach can obtain a world model, it ignores the spatial relationships between action sequences. To better capture the underlying environment dynamics, we define the compatibility label in Eq. (10) using a logarithmic mapping of spatial distances. Physically, this transformation converts the geometric Euclidean metric into a high-fidelity compatibility density. The $-\log$ operator ensures that the supervision signal is highly sensitive to small spatial deviations, thereby forcing the world model to distinguish subtle differences between the ground-truth trajectory and its spatially-close neighbors (i.e., hard negatives).

$$C_{\text{label}} = \{-\log[d(a_j^i, a_{gt}) + \epsilon]\}_{j=1,2,\dots,K-1}^{i=1,2,\dots,M_{\text{counter}}} \quad (10)$$

where d represents the metric distance of space, and we choose the simplest Euclidean distance. ϵ is a small constant. In the spatial distance, the smallest element value is 0, which corresponds to the ground truth action. It is worth noting that this method can be directly used to train multiple goal images together with almost no increase in computational. We only need to replace z_g and a_{gt} with multiple ground truth on the same action sequence during training. Afterwards, we use simple MSE loss function: $\mathcal{L} = \text{MSE}(C, C_{\text{label}})$.

Best practices When we need to use world models for planning or decision-making, we first give the current observation o_t , different action sequences $\mathcal{A}$, and predict the executed features $\hat{\mathcal{Z}}$. Then we can get the compatibility score C with the goal image we want to reach, and select the action sequence with the highest compatibility score to execute. In fact, this is still the paradigm of predicting first and then discriminating. This reminds us of a question: Can we use the information of the goal when encoding the image from the beginning? If the goal features are given first during prediction in Eq. (3), we can get better results. That is to say, if the world model is for planning, we can provide goal information in advance when encoding observations and then determine

the compatibility score. Then Eq. (3) becomes $z_t^o = E_o(o_t, o_g)$. This approach is equivalent to giving a goal in advance, and then the world model imagines the compatibility score to the goal after performing different actions. This goal can not only be an image target, but can also be extended to semantic targets [26] and language instructions [40]. However, previous world model mainly focused on dynamic models of the environment without providing goal information in advance. Therefore, for the best performance and fair comparison, we test two options separately: providing goal prompts in advance and not providing them. The specific details can be further discussed in the experimental section.

3.3 Policy Learning from World Model

Once the world model is trained, it can be leveraged to utilize large-scale unlabeled video data for policy learning. We define a goal-conditioned policy $\pi(a_t | o_t, o_g)$ that takes the current observation o_t and the goal observation o_g as inputs, and outputs an action sequence that drives the robot toward the goal state.

Following prior autoregressive policy formulations such as OpenVLA [22] and ArMoNa [37], our policy generates actions as a sequence of discrete tokens. Specifically, we first compute action increments Δa and apply k-means clustering to obtain $K_{\text{center}} = 64$ cluster centers, which serve as discrete action tokens. Any continuous action sequence can thus be represented as a sequence of these tokens. Our policy model consists of a visual encoder and a causal transformer decoder. The visual encoder extracts features from the current and the goal observation, while the causal decoder autoregressively predicts the action tokens. Similar to next-token prediction in natural language processing, the model generates tokens sequentially, starting from a special start token and continuing until an end token is produced.

In the absence of ground-truth action labels, we exploit the predictive capability of the world model to generate pseudo action labels from unlabeled video sequences $\{o_t, o_{t+1}, \dots\}$. We first construct a candidate action set by clustering existing action sequences into $M_c = 64$ cluster centers, denoted as $\mathcal{A}_c = \{A_c^{(1)}, A_c^{(2)}, \dots, A_c^{(M_c)}\}$. We then randomly sample observation-goal pairs from video trajectories. For each pair (o_t, o_g) , we evaluate all candidate action sequences using the compatibility score from world model. The action sequence with the highest compatibility score is selected as the pseudo label. The discretized action sequence can be represented as $\{a_{start}, a_t, a_{t+1}, \dots, a_{t+K_{\text{pre}}}, a_{end}\}$.

Finally, the policy model is trained using a standard autoregressive cross-entropy loss:

$$\mathcal{L} = - \sum_{i=t}^{t+K_{\text{pre}}} \log \pi_{\theta}(a_i | o_t, o_g, a_{<i}), \quad (11)$$

which encourages the model to generate action sequences that lead to the desired goal observation.

3.4 Policy Reinforcement in Latent World Model

After the initial imitation stage, the policy model acquires the basic ability to generate action sequences conditioned on the current and goal observations. However, this stage is limited by the fixed candidate action set constructed from clustering, which restricts policy expressiveness. To further improve the policy, we perform reinforcement learning entirely within the latent world model. This process is analogous to the cold-start reinforcement paradigm used in DeepSeek-R1 [11], where the model is progressively improved beyond the initial imitation distribution.

Unlike the imitation stage, where the policy is trained on a fixed set of clustered action sequences, the autoregressive policy can now freely generate novel action sequences by composing discrete tokens. Given a current observation o_t and a goal observation o_g , we sample multiple candidate action sequences from the pretrained policy:

$$\mathcal{A} = \{A^{(1)}, A^{(2)}, \dots, A^{(M_{\text{gen}})}\}, \quad A^{(i)} \sim \pi_{\theta}(\cdot \mid o_t, o_g). \quad (12)$$

After predicting the consequences of different actions, the discrimination module of LWM can naturally be seen as a reward model. Specifically, for each sampled action sequence $A^{(i)}$, the world model evaluates its compatibility with the goal observation, producing a scalar reward:

$$\hat{z}^i = \text{WM}_{\text{predict}}(o_t, A^{(i)}), \quad r_i = \text{WM}_{\text{discriminative}}(\hat{z}^i, o_g) \quad (13)$$

where higher reward indicates that the predicted outcome is closer to the goal state. To stabilize training, we normalize the rewards across sampled sequences to obtain advantages:

$$A_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}. \quad (14)$$

We optimize the policy using a GRPO-style [11] objective, which encourages high-reward actions while constraining the policy to remain close to a reference policy:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{M_{\text{gen}}} \sum_{i=1}^{M_{\text{gen}}} (r_{\text{ratio}} A_i - \beta \mathbb{D}_{KL}(\pi_{\theta} \parallel \pi_{\text{ref}})), \quad (15)$$

where the probability ratio is defined as:

$$r_{\text{ratio}} = \min \left(\frac{\pi_{\theta}(\mathbf{a}_i)}{\pi_{\theta_{\text{ref}}}(\mathbf{a}_i)}, \text{clip} \left(\frac{\pi_{\theta}(\mathbf{a}_i)}{\pi_{\theta_{\text{ref}}}(\mathbf{a}_i)}, 1 - \epsilon, 1 + \epsilon \right) \right). \quad (16)$$

This objective encourages the policy to increase the likelihood of action sequences that lead to goal-consistent future states. Importantly, this reinforcement process is performed entirely within the learned latent world model, without requiring additional real-world interaction or action annotations. By iteratively sampling, evaluating, and optimizing in the latent space, the policy can continuously improve and discover more effective action sequence beyond the initial imitation distribution.

4 Experiment

4.1 Experimental Setting

Datasets. We conduct experiments using a total of three datasets. For the public dataset, we use the datasets RECON [29] and SCAND [21] used by NWM [2] respectively. The RECON dataset is a fisheye camera dataset collected in outdoor environments, which includes a large number of lawn houses, etc. The SCAND dataset is collected in the campus environment, containing a large number of dynamic objects and human interaction scenarios. We selected data from all wheeled robots for approximately three hours. In addition, to validate the practicality of our framework, we collect our own dataset LWM (navigation dataset in Latent World Model) using the robot shown in Fig. 3. The dataset includes campus, residential area, and park scenes, covering approximately 6 hours and $600,000 m^2$. We use wheel encoders and IMUs to obtain the position of the robot, sample one keyframe over 0.2m, and set the maximum predicted frame $K = 64$. We use 5% of the dataset as the test set and the remaining 95% as the training set. Within the training set, half the data is used to train the world model and imitation learning, while the other half is used to reinforce the world model.

Baselines.

- NWM [2] is a world model based on image generation for robot navigation tasks. When planning and making decisions, the world model generates different image sequences based on different actions. We follow the approach in the paper to select the action sequence with the minimum LPIPS [42] loss between the goal image and the generated image. The model has already been trained on RECON and SCAND. To align with our evaluation method, we fine tuned NWM-CDiT/XL-1B model for 10 epochs on these two datasets and 30 epochs on LWM.
- Dino-WM [44] is a world model based on feature space. All observed images are encoded into features by a Dino-V2 encoder [44]. The features of observation image and actions are fed into the world model to predict the feature of the goal image. During deployment, Dino-WM also used different action sequences for feature prediction, and ultimately selects the action sequence with the smallest error compared to the feature of goal image for execution.
- NoMaD [33] is a robot navigation method based on diffusion policy [4]. The current image and the goal image are encoded as feature conditions, and then the noise is guided to transform into an action sequence.
- BC is a baseline robot navigation method we designed based on ViNT [31] and GNM [30]. It is based on imitation learning. The current and the goal image are input to an image encoder and then decoded into an action sequence.

For fair comparison with baseline methods, we deploy three LWM: LWM-S uses Dino-v2-small [27] as encoder which has the same 49M parameters as Dino-WM [44]. LWM-B and LWM-B-P uses Croco-v2-base [38] as encoder which has

217M parameters. LWM-B-P is given the goal prompt in advance as described in 3.2. We compare policy model with the same number of parameters. Specific implementation details can be found in the supplementary materials.

Evaluation Metrics. For the evaluation of the world model, we measure prediction accuracy. We use the 64 action sequences in Section 3.3 obtained from clustering as candidate trajectories. Following NWM [2], for each sample in the test dataset, we roll out all candidate action sequences in the world model and select the trajectory whose final state best matches the goal.

- **PE and OE.** To quantify prediction error, we compute the Euclidean distance between the predicted goal coordinates and the corresponding ground-truth goal coordinates. Specifically, we report two metrics: Position Error (translation error) and Orientation Error (rotation error), corresponding to the discrepancies in spatial position and attitude, respectively.
- **ACC.** In addition, we evaluate the consistency of the world model by measuring its ability to distinguish the correct action sequence from counterfactual alternatives. Specifically, for each sample, we construct a candidate set consisting of the ground-truth action sequence and several counterfactual action sequences from the dataset. Given the current and goal observation, the world model can select an action sequence from candidate set. We report the classification accuracy under different candidate set sizes, including 3 and 5 candidates, denoted by ACC3 and ACC5.
- **SR and LPIPS.** To evaluate whether the policy model can enable robots to reach their goals, we deploy them in real-world environments. If the robot reaches within 0.5m of the goal, it is considered successful. To evaluate the accuracy of reaching the goal, when the robot succeeds in reaching the goal, we compare the minimum LPIPS between the robot observation and the goal image.

Table 1: World Model Prediction and Planning Evaluation. Comparison of prediction errors (PE, OE) and action selection accuracy (ACC3, ACC5). Our method achieve the lowest errors and highest accuracy compared to baseline methods.

Method	RECON				SCAND				LWM			
	PE ↓	OE ↓	ACC3 ↑	ACC5 ↑	PE ↓	OE ↓	ACC3 ↑	ACC5 ↑	PE ↓	OE ↓	ACC3 ↑	ACC5 ↑
NWM	7.222	1.279	40.97	27.23	5.534	0.623	41.66	26.04	5.710	0.884	42.43	26.26
DINO-WM	5.937	1.254	84.05	73.92	3.993	0.212	41.96	24.35	4.687	0.705	55.40	40.40
LWM-S (ours)	1.896	0.377	85.34	79.20	2.862	0.207	67.87	47.66	3.200	0.404	66.40	51.60
LWM-B (ours)	1.283	0.394	91.45	87.50	2.481	0.215	58.03	39.37	3.360	0.388	61.60	46.40
LWM-B-P (ours)	1.264	0.425	93.10	88.14	0.651	0.122	83.93	70.98	1.122	0.125	90.60	83.80

4.2 Latent World Model

Compared to other world models, can our framework better predict and plan? We compare our method with several baseline approaches in Table 1. Our method achieves the lowest prediction error and the highest action selection accuracy, demonstrating the effectiveness of compatibility modeling. Although Dino-WM also performs prediction in the latent space, it does not use counterfactual action sequences for enhancement. Moreover, the Dino-V2 encoder remains frozen to avoid mode collapse during training, which constrains the adaptability of the representation to the navigation task. As a result, its performance is bounded by the pre-trained visual features and cannot fully exploit task-specific dynamics. Although NWM achieves strong performance in pixel-space prediction, it performs worst when used for planning with predicted images, yielding the highest error and lowest accuracy. This suggests that visually plausible pixel-level generations do not necessarily preserve accurate spatial or geometric consistency, which is critical for robot navigation.

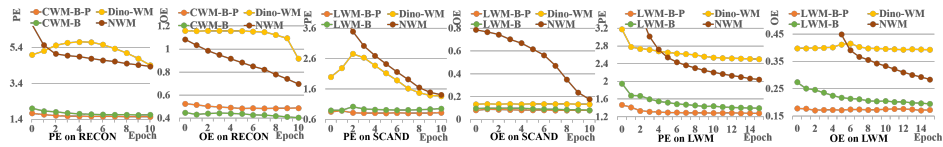


Fig. 2: Policy Reinforcement within the World Model. Policies starting from different initializations achieve performance improvements when further optimized via GRPO within LWM.

Table 2: Policy Learning Performance. Evaluation of policies trained under the supervision of different world models demonstrate that LWM provides superior pseudo-labels for policy imitation.

Method	RECON		SCAND		LWM	
	PE ↓	OE ↓	PE ↓	OE ↓	PE ↓	OE ↓
NWM	4.983	0.976	4.442	0.787	4.983	0.976
DINO-WM	4.994	1.159	1.995	0.133	3.177	0.397
LWM-S (ours)	2.013	0.448	0.899	0.096	1.678	0.209
LWM-B-P (ours)	1.729	0.523	0.853	0.081	1.468	0.178

4.3 Learning and reinforcing policy in LWM

After training the world model, our ultimate objective is to leverage it for policy learning. We evaluate the effectiveness of different world models for policy training in Table 2. For NWM, we use the generated image as the pseudo goal image

and the ground truth action sequence as the label. Because the policy model is an autoregressive action generation model, we still report PE and OE as metrics. Our method achieves the lowest error on both metrics, demonstrating that the learned world model provides reliable supervisory signals for training policies directly from unlabeled video data. This capability is particularly valuable in robotics, where action annotations are expensive and scarce.

During the initial policy training stage, we rely on a fixed set of 64 clustered action sequences, which limits the coverage of the action space. To overcome this limitation, we further improve the policy through reinforcement learning within the world model. In practice, policy initialization may come from different sources, so we use the LWM-B-P world model as a unified reward function to evaluate and optimize different initial policies. The reinforcement results are shown in Fig. 2. We observe performance improvements across all policy initializations, demonstrating that our world model can effectively refine policies regardless of their origin. This highlights the generality and practical applicability of our method for scalable policy learning from video data.

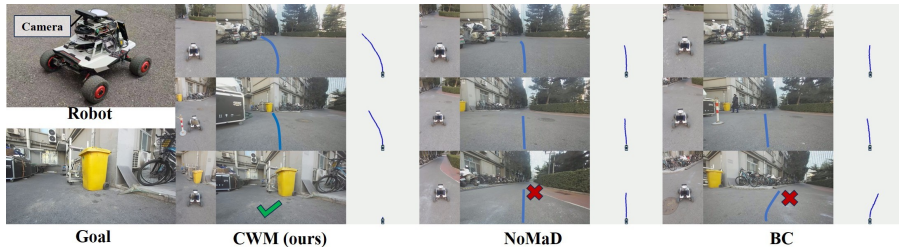


Fig. 3: Navigation in real-world environment. LWM can learn policy models within the world model.

Table 3: Real-World Navigation Performance.

Method	SR $\uparrow$	LPIPS $\downarrow$
NoMaD	33.3	0.448
BC	33.3	0.566
LWM-B-P (ours)	66.7	0.396



Fig. 4: LWM demonstrates zero-shot generalization to indoor environments.

4.4 Navigation in the real world

To validate the effectiveness of our policy trained and reinforced within the world model, we deploy it in real-world robotic environments. We evaluate our

method and baseline approaches across six distinct navigation scenarios. The average success rate and the LPIPS are reported in Table 3. Our method achieves the highest success rate and the lowest LPIPS, demonstrating that our world model has rich imagination and spatial knowledge, even in previously unseen environments. Importantly, our approach does not rely on any action annotations during training. The policy is learned entirely from video data using the world model. As illustrated in Fig. 3, our method remains robust even when the goal is located far from the initial position. The policy consistently maintains alignment with the goal state throughout execution. In contrast, imitation learning and diffusion-based policies can perform well in simple scenarios but often fail in more challenging settings, where they tend to lose track of the goal or produce inconsistent trajectories.

Furthermore, we evaluate our method in indoor environments that are completely absent from the training data. As shown in Fig. 4, our policy successfully reaches the same goal location from different starting positions. This behavior indicates that the policy has learned meaningful spatial and geometric relationships, rather than merely memorizing action sequences. More real-world robot navigation deployment videos are provided in the supplementary materials.

Table 4: Ablation on Counterfactual Action Sequences. Different action sequences can introduce richer comparisons for supervision and improve performance.

Method	PE ↓	OE ↓	ACC3 ↑	ACC5 ↑
$M = 1$	2.239	0.668	34.19	23.83
$M = 2$	1.234	0.327	61.86	44.55
$M = 4$	1.128	0.293	64.76	52.33
$M = 8$	1.008	0.263	65.28	46.11
$M = 16$	0.890	0.147	70.46	58.54

Table 5: Ablation on Loss Functions. Our proposed method (MSE with log) effectively captures spatial relationships and subtle action consequences, achieving the best performance.

Method	PE ↓	OE ↓	ACC3 ↑	ACC5 ↑
Contrastive loss	1.298	0.124	36.00	25.60
Contrastive and ranking loss	1.186	0.212	85.40	80.60
MSE	1.129	0.209	86.40	79.00
MSE with log (ours)	1.122	0.125	90.60	83.80

4.5 Ablation

We first ablate the effect of modeling interactions among multiple candidate action sequences during training on SCAND dataset using LWM-B-P. Specifically, we vary the number of imagined action sequences evaluated by the world model. As shown in Table 4, $M > 1$ indicates that the model uses counterfactual reasoning during training which has better performance. Different action sequences can introduce richer comparisons for supervision and allows the world model to better capture their consequences. For completeness, more extensive results are presented in the supplementary material.

We also study different loss functions on LWM dataset using LWM-B-P. As shown in Table 5, using a contrastive loss leads to higher prediction error compared to MSE. This is because contrastive loss only enforces discrete separability between representations, without capturing the magnitude of differences between

predicted and target features. We further introduce a ranking loss to encourage better relative ordering among action sequences. While this reduces the prediction error compared to pure contrastive loss, it still underperforms MSE loss. In contrast, our proposed loss achieves the best performance. This indicates that our loss formulation provides more informative supervision, enabling the model to better distinguish subtle differences between action outcomes and focus on harder action sequences.

5 Conclusion

We present the compatibility prediction Latent World Model (LWM) for visual navigation. Instead of reconstructing observations, LWM predicts action-conditioned latent feature compatibility via cross-trajectory counterfactual reasoning. This efficient latent space supports policy supervision from unlabeled videos and imagination-driven reinforcement learning, significantly outperforming existing world models in real-world robotic navigation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. U22B2055, 62273345) and in part by the Beijing Natural Science Foundation (No. L223003).

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025)
3. Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., et al.: Worldvla: Towards autoregressive action world model. arXiv preprint arXiv:2506.21539 (2025)
4. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research* **44**(10-11), 1684–1704 (2025)
5. Ding, J., Zhang, Y., Shang, Y., Zhang, Y., Zong, Z., Feng, J., Yuan, Y., Su, H., Li, N., Sukiennik, N., et al.: Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys* **58**(3), 1–38 (2025)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)

8. Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., Abbeel, P.: Learning universal policies via text-guided video generation. *Advances in neural information processing systems* **36**, 9156–9172 (2023)
9. Gao, Y., Guo, H., Hoang, T., Huang, W., Jiang, L., Kong, F., Li, H., Li, J., Li, L., Li, X., et al.: Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113* (2025)
10. Georgiev, I., Giridhar, V., Hansen, N., Garg, A.: Pwm: Policy learning with multi-task world models. *arXiv preprint arXiv:2407.02466* (2024)
11. Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025)
12. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. pp. 297–304. *JMLR Workshop and Conference Proceedings* (2010)
13. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. *Advances in neural information processing systems* **31** (2018)
14. Ha, D., Schmidhuber, J.: World models. *arXiv preprint arXiv:1803.10122* **2**(3), 440 (2018)
15. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603* (2019)
16. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: *International conference on machine learning*. pp. 2555–2565. *PMLR* (2019)
17. Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193* (2020)
18. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. *Advances in neural information processing systems* **35**, 8633–8646 (2022)
19. Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080* (2023)
20. Kang, B., Yue, Y., Lu, R., Lin, Z., Zhao, Y., Wang, K., Huang, G., Feng, J.: How far is video generation from world model: A physical law perspective. *arXiv preprint arXiv:2411.02385* (2024)
21. Karnan, H., Nair, A., Xiao, X., Warnell, G., Pirk, S., Toshev, A., Hart, J., Biswas, J., Stone, P.: Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters* **7**(4), 11807–11814 (2022)
22. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246* (2024)
23. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603* (2024)
24. Li, C., Krause, A., Hutter, M.: Robotic world model: A neural network simulator for robust policy optimization in robotics. *arXiv preprint arXiv:2501.10100* (2025)
25. Li, Y., Zhu, Y., Wen, J., Shen, C., Xu, Y.: Worldeval: World model as real-world robot policies evaluator. *arXiv preprint arXiv:2505.19017* (2025)

26. Nie, D., Guo, X., Duan, Y., Zhang, R., Chen, L.: Wmnav: Integrating vision-language models into world models for object goal navigation. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2392–2399. IEEE (2025)
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
28. Ren, Z., Wei, Y., Guo, X., Zhao, Y., Kang, B., Feng, J., Jin, X.: Videoworld: Exploring knowledge learning from unlabeled videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29029–29039 (2025)
29. Shah, D., Eysenbach, B., Kahn, G., Rhinehart, N., Levine, S.: Rapid exploration for open-world navigation with latent goal models. arXiv preprint arXiv:2104.05859 (2021)
30. Shah, D., Sridhar, A., Bhorkar, A., Hirose, N., Levine, S.: Gnm: A general navigation model to drive any robot. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 7226–7233. IEEE (2023)
31. Shah, D., Sridhar, A., Dashora, N., Stachowicz, K., Black, K., Hirose, N., Levine, S.: Vint: A foundation model for visual navigation. arXiv preprint arXiv:2306.14846 (2023)
32. Silver, D., Hasselt, H., Hessel, M., Schaul, T., Guez, A., Harley, T., Dulac-Arnold, G., Reichert, D., Rabinowitz, N., Barreto, A., et al.: The predictron: End-to-end learning and planning. In: International Conference on Machine Learning. pp. 3191–3199. PMLR (2017)
33. Sridhar, A., Shah, D., Glossop, C., Levine, S.: Nomad: Goal masked diffusion policies for navigation and exploration. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 63–70. IEEE (2024)
34. Team, R., Gao, Z., Wang, Q., Zeng, Y., Zhu, J., Cheng, K.L., Li, Y., Wang, H., Xu, Y., Ma, S., et al.: Advancing open-source world models. arXiv preprint arXiv:2601.20540 (2026)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
36. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
37. Wang, Z., Hu, H., Gao, W., Shen, S.: Causal enhanced autoregressive model for monocular image-goal navigation in unknown map environment. *IEEE Robotics and Automation Letters* (2025)
38. Weinzaepfel, P., Lucas, T., Leroy, V., Cabon, Y., Arora, V., Brégier, R., Csurka, G., Antsfeld, L., Chidlovskii, B., Revaud, J.: CroCo v2: Improved Cross-view Completion Pre-training for Stereo Matching and Optical Flow. In: ICCV (2023)
39. Xiao, J., Yang, Y., Chang, X., Chen, R., Xiong, F., Xu, M., Zheng, W.S., Zhang, Q.: World-env: Leveraging world model as a virtual environment for vla post-training. arXiv preprint arXiv:2509.24948 (2025)
40. Yao, X., Gao, J., Xu, C.: Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5536–5546 (2025)
41. Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., et al.: World action models are zero-shot policies. arXiv preprint arXiv:2602.15922 (2026)

42. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
43. Zhen, H., Qiu, X., Chen, P., Yang, J., Yan, X., Du, Y., Hong, Y., Gan, C.: 3d-vla: A 3d vision-language-action generative world model. arXiv preprint arXiv:2403.09631 (2024)
44. Zhou, G., Pan, H., LeCun, Y., Pinto, L.: Dino-wm: World models on pre-trained visual features enable zero-shot planning. arXiv preprint arXiv:2411.04983 (2024)