ODEWorld: A Continuous Predictive Architecture via Physical-Time Flow

Dongxiu Liu[1*†], **Haoyi Niu**[2*✉], **Peng Cheng**[1], **Yuan Gao**[1], **Xirui Kang**[1†], **Sangli Teng**[2], **Koushil Sreenath**[2], **Xianyuan Zhan**[1✉]

[1] Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China

[2] Berkeley Artificial Intelligence Research (BAIR) Lab, University of California, Berkeley, USA

Abstract

In the physical world we inhabit, space and time are fundamentally continuous. However, existing machine learning paradigms for world modeling are largely confined to discrete-time prediction and inference, thereby exhibiting significant inefficiency in capturing the underlying dynamics of the physical world. To bridge this gap, we introduce Physical-Time Flow (**PT-Flow**), a novel approach that learns a continuous latent velocity field operating in physical time. Crucially, the underlying dynamics of sequential data are parameterized by an ordinary differential equation (ODE) embedded in a well-structured dynamical representation space. Under this paradigm, the prediction of the future can be recast as temporal integration via an ODE solver in the compressed latent space. Building upon PT-Flow, we construct **ODEWorld**, a continuous-time latent world model that is both efficient and versatile. By extracting time-variant features and enforcing ODE properties on both the dynamical representation space and the latent velocity field, ODEWorld effectively addresses the long-standing representation collapse issue in the latent world model literature. This also enables high-quality image reconstruction with ODEWorld even after long-horizon prediction. Moreover, its continuous nature allows for arbitrary temporal resolution and even backward prediction, which is impossible for most discrete-time models. Lastly, benefiting from the learned dynamics-centric latent space, ODEWorld can provide rich planning-oriented information to facilitate downstream policy learning. Comprehensive experiments across multiple simulation and real-world datasets demonstrate that ODEWorld successfully reconciles planning-conducive dynamics abstraction with visual realism, excelling in both video generation and robotic control. Project Website.

1 Introduction

The physical world we inhabit is fundamentally continuous in both space and time. In our daily life, objects move, interact, and collide in continuous time, governed entirely by underlying physical laws; meanwhile, biological organisms, such as humans and animals, are naturally attuned to the perception, understanding, and decision-making of continuous transitions. Despite this, contemporary AI models almost exclusively rely on discrete-time modeling paradigms, such as modeling video as sequences of image frames [1, 2, 3, 4, 5] and solving decision-making tasks in discrete time steps [6, 7, 8, 9, 10]. This inherent discrepancy makes modern predictive architectures highly inflexible when handling irregularly sampled data, imposing strict requirements on data for temporal alignment during both training and generation [11, 12]. Furthermore, simply relying on learning discrete-time pixel/sensor measurement transitions ignores the fundamental continuity in physics-grounded systems, causing current models to function primarily as "video-mimicking", and become inefficient in internalizing the intrinsic, higher-order physics inherent in reality [13, 14, 15, 16].

[*]Equal contribution, order determined by coin flip. † Work done during internship at AIR.

[✉]Corresponding authors: niu@berkeley.edu, zhanxianyuan@air.tsinghua.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To bridge this gap, we introduce the concept of Physical-Time Flow (PT-Flow). At its core, PT-Flow conceptualizes the underlying dynamics in the data as a continuous velocity field in a well-structured latent space, where the latent state transition is smoothly governed by an ordinary differential equation (ODE) along actual physical time. To make such a continuous modeling framework possible, PT-Flow features two critical designs: 1) *dynamical representation decoupling* and 2) *direct first-order supervision*. The former decouples the time-varying, dynamics-related information from the data into a clean, ODE-compatible dynamical representation space, getting rid of the nuanced background and time-invariant details for efficient learning. The latter directly supervises the latent velocity field with the approximated ground truth latent velocities through an elegant Jacobian-vector product (JVP) projection. The above two designs lead to a new predictive architecture, in which the prediction of the future can be recast as temporal integration via an ODE solver in the compressed latent space, in some sense similar to flow matching [17], but operating in physical time.

Building upon PT-Flow, we construct a continuous-time latent world model, called **ODEWorld**, which embodies many desirable features. First, it disentangles dynamical representation by conditioning the encoder and decoder directly on static context, thereby offloading the burden of dynamical representation learning, allowing the latent space to focus exclusively on capturing essential dynamical changes. Second, ODEWorld circumvents the long-standing challenge of representation collapse in Joint Embedding Predictive Architecture (JEPA)-based frameworks [18, 19, 20], where the latent space can degenerate into trivial solutions due to the indirect supervision of the coupled encoder and latent predictor. While some recent efforts [21, 22] have added external regularization schemes to mitigate this issue, however, at the expense of sacrificing model expressiveness. By contrast, ODEWorld imposes direct supervision on the latent velocity field while harmoniously regularizing the ODE properties of the dynamical representations. These architectural choices empower ODEWorld to achieve high-fidelity image reconstruction with an extremely compact model, even across extended temporal horizons. Moreover, its continuous nature also enables arbitrary temporal resolution and even backward prediction–capabilities fundamentally absent in most discrete-time predictive models.

Empirical evaluations across diverse simulation and real-world benchmark datasets, including LIBERO [23] and Agibot-World [24], demonstrate that ODEWorld effectively reconciles planning-conducive dynamics abstraction with visual realism. We show that the learned latent velocity field produces interpretable and smooth transformation paths between the given initial and goal states. This well-behaved continuous latent flow enables temporally consistent, long-horizon predictions by leveraging off-the-shelf ODE solvers within a highly compact latent manifold. Moreover, prediction by solving the continuous-time ODE naturally supports temporal super-resolution and robust video generation from irregularly sampled or dropped-frame data, where discrete-time models inherently fail. Lastly, robotic control experiments on the LIBERO Benchmark also reveal that ODEWorld extracts rich, planning-centric representations that can greatly facilitate downstream policy learning. Ultimately, these capabilities establish ODEWorld as a highly versatile architecture, setting a new continuous recipe for both high-fidelity video synthesis and effective robotic policy learning.

2 Related Work

Predictive world models. Recent video generation models have achieved notable progress and shown great promise as scalable generative simulators of the physical world [3, 25, 26, 27, 28]. These models learn to predict future observations as a surrogate for physical dynamics, producing full trajectory videos [29, 30] or subgoal images [31, 32]. These models have also garnered substantial attention from the robotics community, which translates their video prediction capability for control. A first line of methods adopts a planning-and-control paradigm, where actions are inferred from predicted videos, via inverse dynamics models [29, 30], goal-conditioned policies [31, 33, 34, 35, 36], or direct pose extraction using off-the-shelf foundation models [37, 38]. A second line of work instead integrates action generation into the world model, learning the joint distributions over future observations and actions [9, 10, 6, 39, 40, 41, 42]. However, all of these models suffer from architectural heaviness due to costly pixel-level reconstruction to derive world understanding, which have long been critiqued for their fundamental computational inefficiency.

To address the efficiency issue, another branch of work focuses on learning latent world models, where predictions are performed in a compact representation space [20, 22, 43, 44, 45, 46, 47, 48, 49]. These approaches construct latent representations through various objectives, including image reconstruction [43, 50], contrastive learning [48, 51, 52], and explicit motion prediction [53, 54, 55], aiming to capture task-relevant structure while reducing the complexity of pixel-space modeling.

An outstanding line of work is the JEPA family [18, 21, 22, 20, 56], which jointly learns to encode observations into a compact latent space and models transition dynamics by predicting future latents. While JEPA provides a compelling paradigm for latent world modeling, these methods can suffer from representation collapse, where the latent space degenerates toward trivial solutions, thus requiring carefully designed training strategies and regularization at the expense of expressivity loss [21, 22]. Both the aforementioned video world models and latent world models are exclusively constructed in discrete-time, which potentially become inefficient when capturing the fundamental higher-order physics in reality [13, 14]. In contrast, our proposed ODEWorld is directly modeled in continuous time while avoiding the representation collapse issue in JEPA frameworks by design, thereby offering great efficiency to model the physical world.

ODE dynamics modeling. Modeling continuous dynamical systems through ordinary differential equations has found extensive applications across diverse scientific domains, including physics [57, 58, 59], climate science [60], and neuroscience [61]. One prominent methodology involves explicitly constructing physics-informed learnable ODE dynamics [59, 62, 63, 15, 16]. While these methods effectively embed known invariant properties into the network structure, they strictly require explicit prior knowledge of the true underlying governing equations or a given set of analytical terms to model the system dynamics, thereby severely limiting their expressiveness. Within this data-driven paradigm, classic approaches like Koopman operator theory [64] are frequently employed in the control community. However, it also suffers from computationally expensive coordinate transformations and remains fundamentally restricted to an approximation of the true system dynamics. To circumvent the reliance on prior knowledge, neural ODE networks that learn neural ODE dynamic models directly from observation data have received increasing interest [65, 66, 67, 68]. Despite being promising, training such neural ODE networks often relies on computationally expensive adjoint methods [65], which are hard to scale and often suffer from instability during training. By contrast, in our proposed PT-Flow paradigm, we directly supervise the latent ODE velocity field with the approximated ground truth latent velocities, which reconciles the preservation of ODE properties with efficient and scalable training.

3 Physical-Time Flow

To go beyond the discrete-time modeling to a new architecture capturing continuous-time physical reality, we introduce the concept of **Physical-Time Flow (PT-Flow)**. The core idea of PT-Flow is to capture the underlying dynamics of high-dimensional data as a continuous velocity field within a proper latent space. Let $s_t \in \mathbb{R}^{d_s}$ denote the state input at physical time t, and $z_t \in \mathbb{R}^{d_z}$ be a properly encoded latent representation of s_t. Here, t represents the physical time elapsed after starting at the initial state s_0. In our setting, s_0 can be flexibly selected from any frame in a demonstration rather than only the first frame. Accordingly, t is defined as the relative temporal offset between s_0 and s_t. To capture the time-evolving dynamics of the physical system, a straightforward approach is to learn a time-dependent velocity field defined via the ordinary differential equation:

$$\frac{dz_t}{dt} = v_\theta(z_t, t; z_0, c) \implies z_T = z_0 + \int_0^T v_\theta(z_t, t; z_0, c)dt \tag{1}$$

where z_0 is the latent representation of the initial state s_0 and c is the target conditioning (e.g., a goal state or a language instruction), which jointly specify the start and end state of the system. If we can learn such a latent velocity field v_θ, then future prediction can be redefined as a temporal integration task over the continuous physical time, thereby unlocking many interesting capabilities, such as arbitrary temporal resolution or backward prediction, as well as being amenable to many nice properties in control theory [13]. The remaining questions are, what is the "proper" latent space and how can we learn v_θ with discrete-time supervision data?

Dynamical representation decoupling. Obviously, not all dynamical systems can be modeled as an ODE function, especially for nuanced mixtures of visual features in a video. Ideally, for video data, we do not want to learn a velocity field to fit every static background detail and time-invariant object, but want to extract clean, essential dynamics-related representations z_t out of the state space. To achieve this, we learn a pair of initial state-conditioned encoder and decoder: $f_{\text{dyn}}(s_t; s_0) = z_t$ and $g_{\text{dyn}}(z_t; s_0) = s_t$. By explicitly isolating static context (time-invariant part between s_0 and s_t) in both encoder and decoder, we make the encoded latent representation z_t focus exclusively on dynamical changes. To prevent information loss, we introduce the following reconstruction loss:

$$\mathcal{L}_{\text{dyn-recon}} = \mathbb{E}_{s_0, s_t \sim \mathcal{D}} \| g_{\text{dyn}}(f_{\text{dyn}}(s_t; s_0); s_0) - s_t \|^2. \tag{2}$$

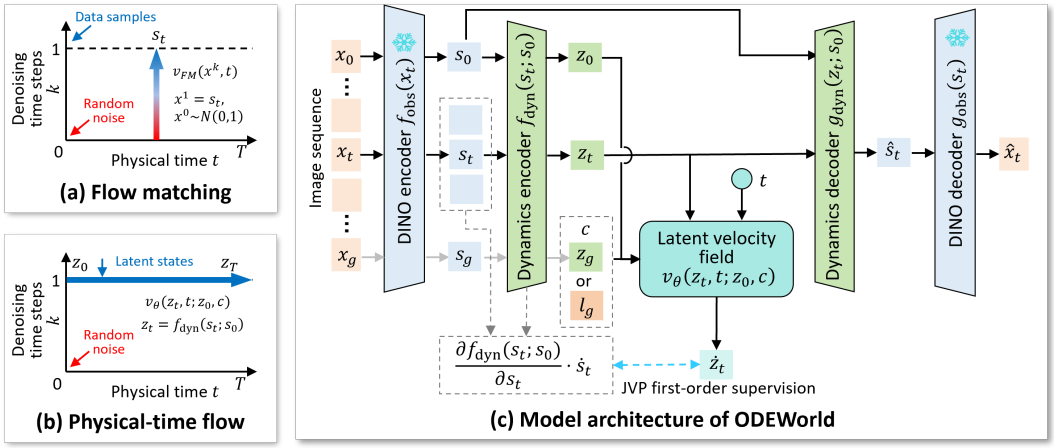


Figure 1: Illustration of PT-Flow and ODEWorld. (a), (b) Comparison between flow matching and our proposed PT-Flow. Although both methods learn time-dependent velocity fields and perform generation with ODE solvers, their notion of time and design philosophies are completely different. (c) Model architecture of ODEWorld.

To further encourage the encoder f_{dyn} to extract fundamental, ODE-compatible dynamical representations, we also enforce f_{dyn} to also satisfy the ODE property, i.e., $\dot{z}_t = \frac{df_{\text{dyn}}(s_t;s_0)}{dt}$. Next, we will show how this can facilitate the latent velocity field learning while achieving a direct, harmonious supervision on both f_{dyn} and v_θ.

Direct first-order supervision. Note that the time derivative of the latent state satisfies

$$\dot{z}_t = \frac{dz_t}{dt} = \frac{df_{\text{dyn}}(s_t;s_0)}{dt} = \frac{\partial f_{\text{dyn}}(s_t;s_0)}{\partial s_t}\cdot\frac{ds_t}{dt} + \cancel{\frac{\partial f_{\text{dyn}}(s_t;s_0)}{\partial s_0}\cdot\frac{ds_0}{dt}} = \frac{\partial f_{\text{dyn}}(s_t;s_0)}{\partial s_t}\cdot\dot{s}_t \tag{3}$$

As s_0 is time-invariant, hence $\frac{ds_0}{dt} = 0$ and the related term vanishes. The above equation suggests that $\dot{z}_t$ can be directly obtained utilizing a Jacobian-vector product (JVP) projection, i.e., $\dot{z}_t = \text{JVP}(f_{\text{dyn}}(s_t;s_0), \dot{s}_t)$. As $\dot{s}_t$ can be directly approximated from training samples, i.e., simply calculating as $\dot{s}_t = (s_{t+1} - s_t)/\Delta t$ (Δt is the time interval between two consecutive states) or adopting more advanced higher-order estimation tricks as we will describe in the next section, the evaluation of $\dot{z}_t$ is practically feasible. This enables us to introduce the following learning objective for the latent velocity field v_θ, which directly supervises it with the approximated ground-truth latent velocities $\dot{z}_t$:

$$\mathcal{L}_v = \mathbb{E}_{s_0,s_t\sim\mathcal{D},t}||v_\theta(z_t,t;z_0,c) - \text{sg}(\text{JVP}(f_{\text{dyn}}(s_t;s_0),\dot{s}_t))||^2, \tag{4}$$

We use the stop-gradient operation $\text{sg}(\cdot)$ to stabilize the optimization of the latent velocity field, following common practices in representation learning [69]. Interestingly, we find that PT-Flow remains effective even without stop-gradient, as shown in App. B.6. This suggests that the proposed objective itself can provide effective constraints for learning meaningful representations. Unlike existing JEPA-style latent world models [18, 19, 20] that supervise the latent predictor via the consistency loss between current and predicted embeddings, PT-Flow employs a direct first-order supervision strategy to specifically optimize the latent velocity field, effectively decoupling the dynamics learning from embedding reconstruction. Such a design greatly mitigates the risk of representation collapse [21, 22] in JEPA frameworks, and ensures the latent manifold remains expressive and dynamically meaningful.

By jointly optimizing the loss objectives $\mathcal{L}_{\text{dyn-recon}}$ and $\mathcal{L}_v$, we can learn a well-structured latent space and velocity field, in which the future prediction at an arbitrary time can be achieved by integrating with an off-the-shelf ODE solver in the compact latent space.

4 ODEWorld: Latent World Modeling via PT-Flow

In this section, we instantiate the PT-Flow paradigm as **ODEWorld** (illustrated in Fig. 1(c)), a highly versatile and efficient continuous-time predictive architecture for both high-fidelity video generation and effective robotic policy learning.

Architecture overview. Apart from building PT-Flow, we first employ a frozen, pre-trained DINOv2 encoder [70] as our vision backbone f_{obs} to project high-dimensional raw image observations x_t into a compressed feature space s_t. This transformed state space offers superior numerical stability compared to raw pixels when estimating the velocity target $\dot{s}_t$ in Eq. (3). Correspondingly, we train a dedicated image decoder g_{obs} to reconstruct the DINO feature back to image space $\hat{x}_t$ for video generation. We follow the same recipe with RAE [71] to train g_{obs} with a combination of L1, LPIPS [72] and adversarial losses [73]:

$$\begin{aligned} s_t = f_{obs}(x_t), \quad \hat{x}_t = g_{obs}(s_t), \quad x_t \in \mathcal{D} \\ \mathcal{L}_{\text{DINO-recon}}(x_t) = \mathcal{L}_1(\hat{x}_t, x_t) + \omega_L \mathcal{L}_{\text{LPIPS}}(\hat{x}_t, x_t) + \omega_G \lambda \mathcal{L}_{\text{GAN}}(\hat{x}_t, x_t). \end{aligned} \tag{5}$$

We further embed PT-Flow within the DINO feature space, where DINO image encoder and decoder f_{obs} and g_{obs} are frozen during the training of PT-Flow. We find that PT-Flow is sufficiently expressive to maintain consistency in latent planning and reconstruction within the DINO feature space, ultimately enabling high-fidelity video generation through g_{obs}.

Following the design of PT-Flow, we introduce an initial state-conditioned dynamics encoder f_{dyn} and a decoder g_{dyn}, both of which are implemented using cross-attention blocks. The dynamics encoder $f_{\text{dyn}}(s_t; s_0) = z_t$ extracts the dynamical representation z_t from the current state s_t conditioned on the initial state s_0, where a few learnable query tokens attend to both s_t and s_0 as keys and values, producing a compact latent representation that summarizes temporal changes between the initial and current states. The decoder g_{dyn} shares a similar structure, again with the initial state s_0 injected but serves as the query, while the latent representation z_{t} acts as keys and values. This design forces the model to reconstruct s_t by combining static context from s_0 with dynamical information encoded in z_t, thereby preventing z_t from carrying static content.

In our experiments, we find that a single token is sufficient to model z_t while achieving strong performance. Specifically, $z_t \in \mathbb{R}^{1\times 768}$ is highly compact compared to $s_t \in \mathbb{R}^{16\times 16\times 768}$ in original DINO space. More importantly, this result further supports the key insight of latent world modeling [18, 22]: raw pixels contain substantial redundancy, while the underlying dynamics can be effectively captured by a much smaller latent representation. Based on the compact z_t, we parameterize the velocity network $v_\theta(z_t, t; z_0, c)$ by a lightweight 3-layer MLP. The time t is incorporated by Feature-wise Linear Modulation (FiLM) layers [74], while other parts z_t, z_0, c are concatenated and fed into the model as input. Such a compact latent dynamics model design makes the latent planning and prediction with the ODE solver extremely efficient, while maximally preserving information for image reconstruction.

Additional enhancements on PT-Flow. To ensure numerical stability, instead of modeling the latent velocity $v_\theta(z_t, t; z_0, c)$ with the actual physical time t as in Eq. (1), we implement $v_\theta(z_t, \tau; z_0, c)$ based on a rescaled physical time τ, where $\tau = \frac{t}{L}$ and L is a normalizing horizon length which is approximately the desired planning length. This will make sure τ mostly resides within the range $[0, 1]$ during training. Under this reparameterization, $\dot{s}_\tau$ corresponds to $\dot{s}_t$ scaled by L in time dimension. We assume that the system reaches a stationary state once the target condition c is reached (e.g., the goal state z_g or language instruction l_g). Therefore, when τ exceeds the rescaled time index τ_c of the last frame in a training sequence ending with condition c, we fix the future state $s_{\tau>\tau_c}$ to the last frame and drive the velocity field v_θ to zero in the end.

To enhance the estimation quality of the target velocity $\dot{s}_t$ for JVP projection in Eq. (4), we employ a Savitzky–Golay derivative filter [75], which performs local polynomial regression over a sliding window. Specifically, given a sequence of states $\{s_{t-k}, \ldots, s_{t+k}\}$ within a window of size $2k+1$, the target velocity is calculated as a linear convolution: $\dot{s}_t = \sum_{i=-k}^{k} w_i s_{t+i}$, where w_i are fixed coefficients obtained by fitting a low-order polynomial in a least-squares sense. In our implementation, we instantiate this filter with a window size of $2k+1 = 5$, yielding a filter kernel $w = \frac{1}{10}[-2, -1, 0, 1, 2]$. Unlike conventional smoothing kernels that blindly average signals (e.g., Gaussian blurring) or standard finite difference kernels, the chosen Savitzky–Golay kernel provides a decent compromise between numerical accuracy and temporal smoothness, due to its adoption of local least-squares polynomial fit. This ensures that the projected target $\dot{z}_t$ (via Eq. (3)) is both temporally smooth and physically accurate, which substantially enhances the learning performance of the latent velocity field v_θ.

Training and inference pipeline. With the pre-trained frozen DINO encoder f_{obs} and decoder g_{obs}, the training of ODEWorld strictly follows the recipe of PT-Flow described in Section 3, which jointly

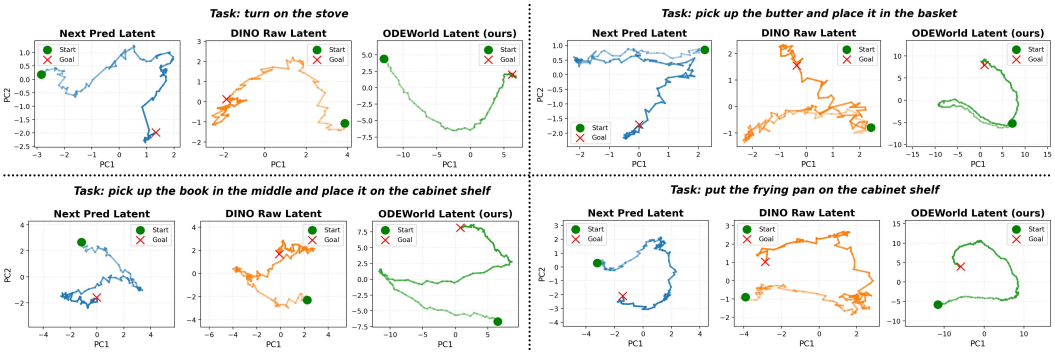


Figure 2: **Data trajectories in different latent spaces.** ODEWorld produces significantly smoother trajectories in its dynamical latent space across various LIBERO tasks compared to the latent spaces of DINO and the next-step prediction baseline.

minimizes the dynamics reconstruction loss $\mathcal{L}_{\text{dyn-recon}}$ (Eq. 2) and the velocity supervision loss $\mathcal{L}_v$ (Eq. 4), i.e.,

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{\text{dyn-recon}} + \lambda_v\mathcal{L}_v \tag{6}$$

In our experiments, we find that simply setting the weighting parameters $\lambda_{rec} = \lambda_v = 1$ without any tuning generally leads to reasonably good results. This further demonstrates the learning robustness of our proposed PT-Flow framework.

During inference, ODEWorld serves as a continuous-time ODE engine for both high-fidelity video synthesis and efficient latent planning. The inference pipeline works as follows: given an initial raw observation x_0 and a target condition c, the model first projects the observation into the DINO space $s_0 = f_{obs}(x_0)$ and then encodes this state into the dynamical latent space $z_0 = f_{dyn}(s_0; s_0)$. We can employ any off-the-shelf ODE solver, e.g. fourth-order Runge-Kutta method (RK4), to integrate the learned velocity field v_θ by $z_\tau = z_0 + \int_0^\tau v_\theta(z_t, t; z_0, c)dt$ to predict the latent state z_τ at arbitrary time $\tau \in [0, 1]$. The evolved latent state z_τ can then be used for: 1) future video frame generation: the latent state is sequentially mapped back to the pixel space with $\hat{s}_\tau = g_{dyn}(z_\tau; s_0)$ and $\hat{x}_\tau = g_{obs}(\hat{s}_\tau)$; and 2) downstream policy learning: z_τ serves as a compact, dynamically meaningful subgoal. Its low-dimensional dynamics-centric nature provides a stable target for policy learning, bypassing the complexity of high-dimensional visual planning. A notable advantage of ODEWorld lies in its remarkable inference speed. By operating within the compact latent manifold z and utilizing a shallow MLP to parameterize the velocity field v_θ, the numerical integration becomes computationally lightweight. This efficiency allows for real-time video generation and fast iterative latent planning, facilitating responsive closed-loop control in complex environments.

5 Experiments

5.1 Experimental Setups

Datasets. We train ODEWorld on LIBERO [23] simulation dataset and AgiBot-World [24] real-world robot datasets. LIBERO consists of 130 tasks with 6,500 high-quality human teleoperation demonstrations, covering diverse motions and atomic skills. We use the full dataset to train ODEWorld and all baseline models for image generation. For the AgiBot-World, we train ODEWorld on the AgiBotWorldChallenge-2025 subset, a large-scale embodied manipulation benchmark built upon the AgiBot World Colosseo platform [24]. The dataset contains approximately 30K manipulation trajectories with an average length of around 260 frames per trajectory, providing diverse robot-object interaction behaviors with visual observations across various real-world manipulation tasks. More details about datasets are provided in App. A.1.

Baselines. To evaluate the video generation quality of ODEWorld, we compare it against latent world models that also support reconstruction from semantic representations. Specifically, we consider Latent Diffusion Planning (LDP) [50] and V-JEPA 2 [20]. LDP employs a reconstruction-based VAE latent space and performs planning and policy learning within this latent space, with generated latents decoded back into images via a VAE decoder. V-JEPA 2 jointly learns a representation encoder and a latent predictor, and similarly decodes predicted latents into raw images through a learned decoder.

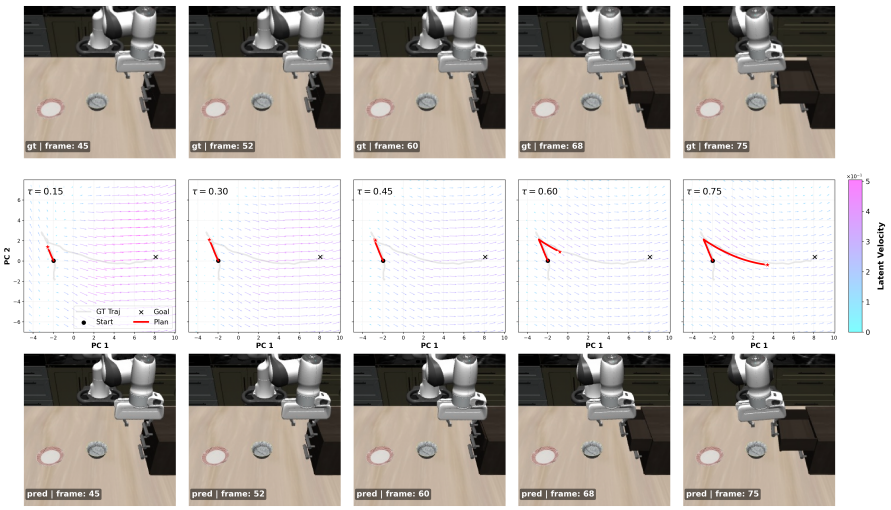


Figure 3: **Velocity field and latent planning.** The middle row shows visualization of globally consistent goal-directed convergence of velocity field and smooth latent plans from ODEWorld that adhere well to the ground truth. The top row shows ground truth frames, and the bottom row shows synthesized frames.

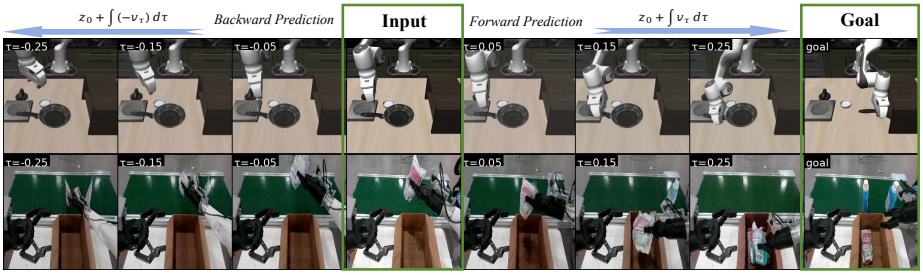


Figure 4: The continuous formulation enables ODEWorld to perform both forward and backward prediction, with motion consistent with real-world dynamics. We present the results on LIBERO benchmark and Agibot World. More results are provided in the Appendix.

These models serve as suitable baselines for assessing generative quality in latent world model settings. As for policy learning, we compare ODEWorld against planning-based baselines, including the editing-based subgoal planner SuSIE [31] and the end-to-end predictive inverse dynamics model Seer [76], the video prediction policy [43], and a vanilla goal-and-language-conditioned behavior cloning (GLCBC) policy. More details about these baselines are provided in App. A.3 and App. A.4.

5.2 Visualization

Latent space analysis. In Fig. 2, we compare the latent space of ODEWorld against the raw DINO feature space and the latent of a discrete next-step predictive baseline. For the latter, we replace v_θ with a discrete transition model $z_{t+1} = p_\theta(z_t; c)$, remove dynamical representation decoupling designs in f_{dyn} and g_{dyn}, and then optimize p_θ with JEPA-style objectives [18]. PCA visualizations for latent trajectories from multiple LIBERO tasks reveal that ODEWorld yields the smoothest latent trajectories. This is attributed to our direct first-order supervision, which regularizes the manifold by aligning it with an analytically filtered velocity target that effectively suppresses high-frequency jitters. Crucially, while ODEWorld enhances dynamical smoothness, it maintains a manifold topology remarkably similar to the original DINO space. Unlike over-regularized models that tend to straighten the latent space to achieve smoothness at the expense of expressivity, ODEWorld's latent space preserves the principal components and the relative geometric structure in the original space. This structural fidelity ensures that the learned latent space retains rich semantic information, providing a high-fidelity foundation for downstream tasks such as video generation and subgoal planning for policy learning.

Table 1: Comparison on video prediction quality and latency.

Methods	Short Fragment @16frames			Long Fragment @64frames		
	PSNR↑	LPIPS↓	Latency(s)↓	PSNR↑	LPIPS↓	Latency(s)↓
LDP	16.33	0.489	1.104	16.27	0.461	3.953
V-JEPA 2	17.60	0.157	0.123	16.47	0.166	0.619
ODEWorld (Ours)	**20.53**	**0.109**	**0.030**	**19.46**	**0.134**	**0.072**

Table 2: On the LIBERO-LONG benchmark, ODEWorld achieves consistent improvements in policy performance. Averaged on 3 random seeds. Tasks are specified in App. A.1.

Method \ Task ID	1	2	3	4	5	6	7	8	9	10	Avg. Suc ↑
GLCBC	83.3	93.3	83.3	93.3	76.6	63.3	73.3	83.3	50.0	80.0	78.0
SuSIE	83.3	63.3	**96.6**	**100.0**	83.3	83.3	83.3	39.9	53.3	76.6	76.3
Seer	88.3	90.0	91.6	81.6	85.0	65.0	**86.6**	80.0	51.6	66.6	78.6
VPP	**96.7**	**100.0**	73.3	93.3	80.0	**96.7**	50.0	76.7	**70.0**	73.3	81.0
ODEWorld (Velocity)	80.0	93.3	76.6	**100.0**	73.3	80.0	83.3	86.6	66.6	83.3	82.3
ODEWorld (Single subgoal)	86.6	86.6	86.6	96.6	**86.6**	86.6	80.0	83.3	46.6	**86.6**	82.6
ODEWorld (Sequential subgoals)	90.0	96.6	90.0	**100.0**	73.3	90.0	76.6	**90.0**	60.0	80.0	**83.6**

Table 3: Real-world robot experiments with ODEWorld-guided policy learning.

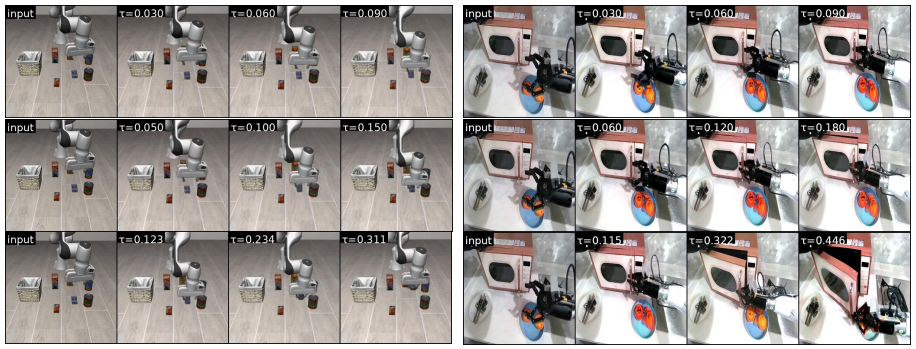
Method	Shrimp	Mouse	Pen	Rearrangement	Average
X-VLA	80%	60%	30%	50%	55%
ODEWorld (Sequential subgoals)	**90%**	**80%**	**70%**	**80%**	**80%**

Velocity field and open-loop latent planning. In the middle row of Fig. 3, we sample a uniform grid in the projected PCA space and utilize the learned velocity field v_θ to compute the instantaneous latent velocities at each point. These are visualized as a vector field with arrows, the color of which represents the magnitude of dynamical transition. Starting from a latent state at a random time of the sampled trajectory, we perform a long-horizon rollout using an ODE solver (RK4) to integrate the velocity v_θ along normalized time τ without any further environmental observations. To verify semantic fidelity, the integrated latent trajectory z_τ is passed through the decoders to synthesize image-space predictions shown in the bottom row, which are then compared against the ground-truth video sequence shown in the top row. The visualization reveals three key properties: (1) The velocity field captures the time-varying magnitude of the dynamics: in this example, the robot first approaches the handle and then pulls the drawer out, and the velocity correctly decays toward zero as the robot reaches the handle before accelerating again into the pulling phase. This shows that the learned velocity field encodes not only the direction toward the goal, but also the correct instantaneous speed at each stage of the task. (2) The model demonstrates high physical consistency even during open-loop planning. The planned trajectory (red line) adheres closely to the ground-truth latent path (gray line), while effectively filtering out the high-frequency noise and jitter present in the latter. (3) Finally, the synthesized predictions in the bottom row remain spatially coherent and semantically accurate. This combination of planning accuracy and smoothness provides a stable, jerk-free foundation for downstream high-fidelity video generation and precise subgoal planning for policy learning.

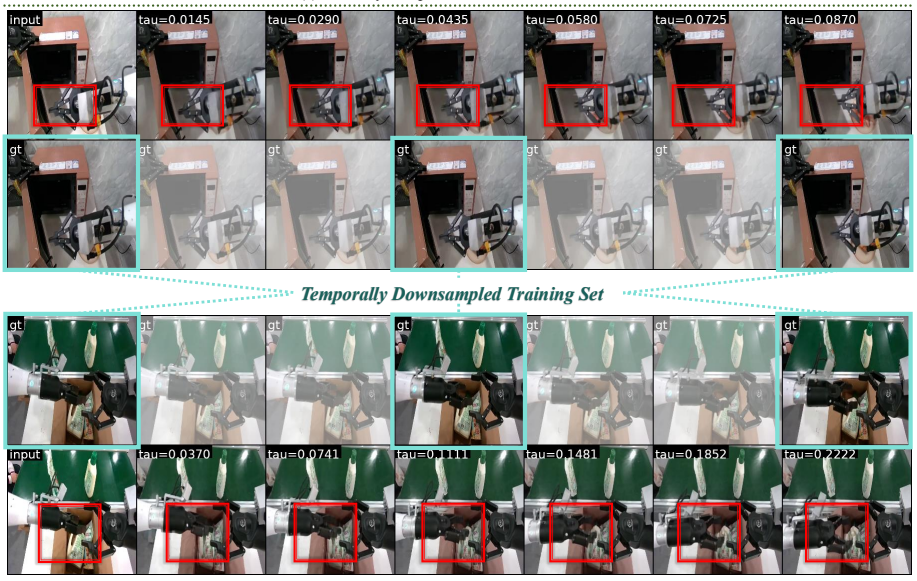
5.3 ODEWorld for Video Generation

Bidirectional video generation. As demonstrated in Fig. 4, ODEWorld exhibits unique bidirectional flexibility inherent to its ODE-governed architecture. By simply reversing the integration sign ($v_\tau \to -v_\tau$), the model can synthesize physically plausible backward video sequences from the same learned latent velocity field. This symmetry validates that ODEWorld internalizes the underlying continuous physics , rather than merely memorizing forward predictive patterns typical of those discrete predictive architectures.

Any-resolution video generation. As illustrated in Fig. 5, ODEWorld leverages its continuous, ODE-governed nature to achieve arbitrary temporal resolution in video generation. Unlike discrete models constrained by fixed step sizes, our model can generate motion at any speed (Fig. 5(a)) and effectively interpolate missing frames (Fig. 5(b)). As shown in Fig. 5(b), even when trained on a



(a) Arbitrary Temporal Resolution Generation



(b) Temporal Super-Resolution

Figure 5: ODEWorld allows any resolution generation with flexible temporal speed. Based on this, ODEWorld can generalize beyond the sampling rate of the training data. In particular, when trained on temporally downsampled sequences, the model is able to generate smooth and coherent intermediate frames, performing temporal super-resolution and filling in missing observations.

down-sampled dataset (e.g., $1/3$ of the original frame rate), ODEWorld successfully recovers the intermediate temporal dynamics, a capability inherently absent in discrete-time baselines.

Quantitative comparisons. Tab. 1 provides a comprehensive evaluation of video prediction quality and efficiency. ODEWorld consistently outperforms baselines in both pixel-level metrics PSNR [77] and perception-level metrics LPIPS [78] across short and long horizons. ODEWorld maintains high structural fidelity and perceptual realism even in long-horizon predictions (@64 frames). Notably, our model achieves substantially lower prediction latency (0.072s for @64 frames) on a single A100 GPU. These results demonstrate that ODEWorld provides accurate long-horizon predictions while maintaining an efficient architecture.

5.4 ODEWorld for Policy Learning

We assess how ODEWorld can empower downstream policy learning by inferring latent subgoals as guidance. We instantiate three guidance paradigms: (1) velocity-conditioned $\pi(a|z, c, v)$, (2) single-subgoal $\pi(a|z, c, \hat{z}_\tau), \tau = 0.25$, and (3) sequential-subgoal $\pi(a|z, c, \{\hat{z}_{\tau_i}\}_{i=1}^n), n = 5, \tau_i = 0.05i$, where subgoals are generated via ODE integration. c denotes image goal and language instruction.

Results on LIBERO-LONG. As shown in Tab. 2, all ODEWorld variants outperform the baselines on LIBERO-LONG. Among them, the sequential-subgoal paradigm achieves the best performance with an average success rate of 83.6%. This performance gain demonstrates the high-fidelity temporal

consistency of ODEWorld rollouts; by maintaining precise trajectory alignment over horizon, the model provides dense and reliable guidance that does not drift from the task objective. Such consistent sequential subgoal guidance effectively benefits downstream policy learning.

Results on real-world robot experiments. To further validate the effectiveness of ODEWorld beyond simulation benchmarks, we conduct real-world control experiments on a bi-manual AgileX robot. We use X-VLA [79] as the policy backbone and augment it with ODEWorld-generated latent subgoals for guidance. We evaluate four manipulation tasks, including: *Shrimp-to-pot*, *Mouse packing*, *Pen insertion*, *Bi-manual rearrangement*. Detailed task settings are provided in App. A.2. Using the same policy backbone and training protocol, ODEWorld improves the average success rate from 55% to 80% across these four tasks, demonstrating the effectiveness of ODEWorld-generated dynamics guidance in real-world manipulation scenarios.

6 Conclusions

We introduce ODEWorld, a novel continuous latent world model grounded in physical time. The core of ODEWorld is a new predictive paradigm called PT-Flow, which learns a latent ODE velocity field defined on physical time. It recasts future prediction as temporal integration with the latent velocity via an ODE solver, which shares some similarities to flow matching, but operates in physical time rather than the noise space. Through comprehensive experiments on both simulation and real-world benchmark datasets, we show that ODEWorld enjoys a number of desirable properties, such as interpretable latent velocity fields, prediction with arbitrary directions or temporal resolutions, high-quality reconstruction after long-term prediction, and planning-centric representations that facilitate downstream policy learning. Many of the above properties are fundamentally absent in most discrete-time predictive models. We anticipate that PT-Flow and ODEWorld will inspire a new generation of predictive models, paving the way for a more faithful understanding and modeling of the continuous physical world.

Acknowledgment

This work was supported by Wuxi Research Institute of Applied Technologies at Tsinghua University under the Grant 20242001120, and Xiongan AI Institute.

References

[1] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022. 1

[2] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024. 1

[3] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 2

[4] Muyang He, Hanzhong Guo, Junxiong Lin, and Yizhou Yu. Video generation models as world models: Efficient paradigms, architectures and algorithms. *arXiv preprint arXiv:2603.28489*, 2026. 1

[5] Zhihao Wang, Jianxiong Li, Yu Cui, Yuan Gao, Xianyuan Zhan, Junzhi Yu, and Xiao Ma. World value models for robotic manipulation. *arXiv preprint arXiv:2606.24742*, 2026. 1

[6] Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. 1, 2

[7] Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998. 1

[8] N Hansen, X Wang, and H Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning, PMLR*, 2022. 1

[9] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025. 1, 2

[10] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 1, 2

[11] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018. 1

[12] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 1

[13] François E Cellier and Jurgen Greifeneder. *Continuous system modeling*. Springer Science & Business Media, 2013. 1, 3

[14] Aditi S Krishnapriyan, Alejandro F Queiruga, N Benjamin Erichson, and Michael W Mahoney. Learning continuous models for continuous physics. *Communications Physics*, 6(1):319, 2023. 1, 3

[15] Peng Cheng, Xianyuan Zhan, Zhihao Wu, Wenjia Zhang, Youfang Lin, Han Wang, Li Jiang, et al. Look beneath the surface: Exploiting fundamental symmetry for sample-efficient offline rl. *Advances in Neural Information Processing Systems*, 36, 2023. 1, 3

[16] Peng Cheng, Zhihao Wu, Jianxiong Li, Ziteng He, Haoran Xu, Wei Sun, Youfang Lin, Yunxin Liu, and Xianyuan Zhan. Sample efficient offline RL via t-symmetry enforced latent state-stitching. In *The Fourteenth International Conference on Learning Representations*, 2026. 1, 3

[17] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[18] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023. 2, 3, 4, 5, 7

[19] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning, 2024. 2, 4

[20] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 2, 3, 4, 6, 23

[21] Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026. 2, 3, 4

[22] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. 2, 3, 4, 5

[23] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 2, 6, 23

[24] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 2, 6, 23

[25] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[26] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*. 2

[27] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 2

[28] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loïc Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In *9th Annual Conference on Robot Learning*, 2025. 2

[29] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023. 2

[30] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. In *International Conference on Machine Learning*, pages 61885–61896. PMLR, 2024. 2

[31] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 7, 23, 24

[32] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025. 2

[33] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. 2

[34] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025. 2

[35] Jinliang Zheng, Jianxiong Li, Sijie Cheng, Yinan Zheng, Jiaming Li, Jihao Liu, Yu Liu, Jingjing Liu, and Xianyuan Zhan. Instruction-guided visual masking. *Advances in neural information processing systems*, 37:126004–126031, 2024. 2

[36] Jianxiong Li, Zhihao Wang, Jinliang Zheng, Xiaoai Zhou, Guanming Wang, Guanglu Song, Yu Liu, Jingjing Liu, Ya-Qin Zhang, Junzhi Yu, et al. Robo-mutual: Robotic multimodal task specification via unimodal learning. *arXiv preprint arXiv:2410.01529*, 2024. 2

[37] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *Conference on Robot Learning*, pages 3943–3960. PMLR, 2025. 2

[38] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025. 2

[39] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 2

[40] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. 2

[41] Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024. 2

[42] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025. 2

[43] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, pages 24328–24346. PMLR, 2025. 2, 7, 24

[44] Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Learning an actionable discrete diffusion policy via large-scale actionless video pre-training. *Advances in Neural Information Processing Systems*, 37:31124–31153, 2024. 2

[45] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22508–22519, 2025. 2, 23

[46] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. 2

[47] Guan Wang, Haoyi Niu, Jianxiong Li, Li Jiang, Jianming Hu, and Xianyuan Zhan. Are expressive models truly necessary for offline rl? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21062–21070, 2025. 2

[48] Dongxiu Liu, Haoyi Niu, Zhihao Wang, Jinliang Zheng, Yinan Zheng, Zhonghong Ou, Jianming Hu, Jianxiong Li, and Xianyuan Zhan. Efficient robotic policy learning via latent space backward planning. In *International Conference on Machine Learning*, pages 39147–39161. PMLR, 2025. 2

[49] Xirui Kang, Yanpei Shi, Lucy Liang, Roy Gan, Dongxiu Liu, Pushi Zhang, Danpeng Chen, Xiaoyi Qin, Yinan Zheng, Jinliang Zheng, et al. X-tokenizer: A multimodal action tokenizer for vision-language-action pretraining. *arXiv preprint arXiv:2606.14752*, 2026. 2

[50] Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning. In *International Conference on Machine Learning*, pages 68710–68724. PMLR, 2025. 2, 6, 24

[51] Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in robotic manipulation through image edits. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7490–7499. IEEE, 2025. 2

[52] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023. 2

[53] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems*, 2024. 2

[54] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *The Thirteenth International Conference on Learning Representations*. 2

[55] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Cai Zhehao, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. In *International Conference on Learning Representations*, volume 2025, pages 21927–21948, 2025. 2

[56] Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. 3

[57] John David Anderson and J Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995. 3

[58] Ilia Grigorenko and Elena Grigorenko. Chaotic dynamics of the fractional lorenz system. *Physical review letters*, 91(3):034101, 2003. 3

[59] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016. 3

[60] Kevin E Trenberth and Kevin E Trenberth. *Climate system modeling*. Cambridge University Press, 1992. 3

[61] Eugene M Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007. 3

[62] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. 3

[63] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019. 3

[64] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005. 3

[65] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 3, 25

[66] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[67] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019. 3

[68] In Huh, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Time-reversal symmetric ode network. *Advances in Neural Information Processing Systems*, 33:19016–19027, 2020. 3

[69] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 4, 27

[70] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024. 5, 29

[71] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations*, 2026. 5

[72] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[73] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 5

[74] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 5, 24

[75] Tapio Saramäki, SK Mitra, and JF Kaiser. Finite impulse response filter design. *Handbook for digital signal processing*, 4:155–277, 1993. 5, 29

[76] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pages 92033–92052, 2025. 7, 24

[77] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 9

[78] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 9, 24

[79] Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026. 10, 23

[80] Yuchun Feng, Jinliang Zheng, Zhihao Wang, Dongxiu Liu, Jianxiong Li, Jiangmiao Pang, Tai Wang, and Xianyuan Zhan. Demystifying action space design for robotic manipulation policies. In *Forty-third International Conference on Machine Learning*. 23

[81] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 24

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 24

[83] Jianxiong Li, Jinliang Zheng, Yinan Zheng, Liyuan Mao, Xiao Hu, Sijie Cheng, Haoyi Niu, Jihao Liu, Yu Liu, Jingjing Liu, et al. Decisionnce: embodied multimodal representations via implicit preference learning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 29461–29488, 2024. 24, 29

[84] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019. 25

[85] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 26

[86] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pages 23301–23320. PMLR, 2023. 26

[87] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pages 10929–10974. PMLR, 2023. 27

[88] Jianxiong Li, Zhihao Wang, Jinliang Zheng, Xiaoai Zhou, Guanming Wang, Guanglu Song, Yu Liu, Jingjing Liu, Ya-Qin Zhang, Junzhi Yu, et al. Robo-mutual: Robotic multimodal task specification via unimodal learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4182–4189. IEEE, 2025. 29

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We introduce PT-Flow and ODEWorld that are our main contributions.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix E

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: No proofs and assumptions, but we mathematically and rigorously formulate our method in Section 3.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In Section 5 and Appendix A.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: LIBERO and Agibot World datasets can be open accessed. Our code will be released after acceptance.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In Section 5 and Appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report the number of random seeds in experiments.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We train on 8 A800 GPUs and evaluate inference time on 1 A100 GPU. Details are revealed in Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: Yes, we keep this submission a totally anonymous one.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: We talk about it in Section 6.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer [NA] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite them and properly describe them.

    Guidelines:

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

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets

    Guidelines:

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

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

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

    Answer: [NA]

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

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [NA] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix

A Experiment Details

A.1 Datasets

LIBERO [23]. LIBERO is a large-scale language-conditioned robotic manipulation benchmark comprising 130 tasks and 6,500 high-quality human teleoperation demonstrations. These tasks span a diverse set of atomic and compositional skills, including grasping, lifting, placing, opening, and multi-step object rearrangement. The benchmark is organized into five suites, namely LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, LIBERO-90, and LIBERO-Long, each targeting distinct generalization capabilities such as spatial reasoning, object-level variation, goal specification, and long-horizon compositional manipulation. Following the standard evaluation protocol, we use the official LIBERO dataset and re-render all observations at a resolution of 256×256 to ensure a controlled comparison across methods. We train ODEWorld and all image-generation baselines on the full dataset.

AgiBot-World [24]. For AgiBot-World, we train ODEWorld on the AgiBotWorldChallenge-2025 subset, a large-scale embodied manipulation benchmark built on the AgiBot World Colosseo platform [24]. The subset used in our experiments contains approximately 30K manipulation trajectories, with an average length of around 260 frames per trajectory. It captures diverse robot-object interaction patterns and provides visual observations across a wide range of real-world manipulation tasks. Compared with existing short-horizon robot datasets, its relatively long trajectories and rich interaction dynamics make it well suited for continuous-time world modeling and future visual prediction.

LIBERO-LONG. As a subset of LIBERO, LIBERO-LONG focuses on long-horizon compositional manipulation, where each episode requires completing multiple sequential subgoals involving interactions with several objects. The benchmark consists of ten tasks: (1) Soup and sauce to basket, (2) Box and butter to basket, (3) Turn on the stove and place the pot, (4) Bowl to drawer, (5) Mugs on plates, (6) Book to rack, (7) Mug and pudding, (8) Soup and box to basket, (9) Both pots on the stove, and (10) Mug to microwave.

A.2 Real-World Settings

We follow recent SOTA practice [31, 45, 80] to conduct real-world robotics experiments on a bi-manual AgileX robot equipped with two 6-DoF robotic arms, parallel-jaw grippers, and RGB cameras. We use X-VLA [79] as the policy backbone and incorporate ODEWorld-generated latent subgoals for sequential guidance. All experiments are conducted under the same training and evaluation protocol for a fair comparison with the original X-VLA policy. We evaluate ODEWorld on four manipulation tasks: (1) *Shrimp-to-pot*: Place a shrimp into a pot; (2) *Mouse packing*: Pack a mouse into a box; (3) *Pen insertion*: Grasp a thin pen and insert it into a narrow container, requiring fine-grained perception and precise control; (4) *Bi-manual rearrangement*: Simultaneously move objects using both arms in a complex scene, evaluating coordinated bi-manual control and multi-object dynamics modeling. Fig. 6 presents the initial states of these four real-world manipulation tasks.

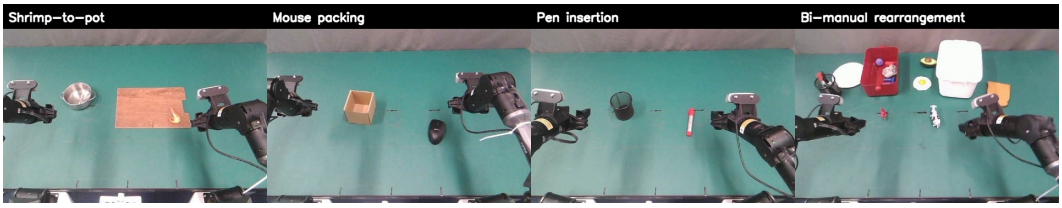


Figure 6: Initial states of the four real-world manipulation tasks. From left to right: *Shrimp-to-pot*, *Mouse packing*, *Pen insertion*, and *Bi-manual rearrangement*.

A.3 Video Generation Baselines

V-JEPA 2 [20]. We adopt the pre-trained V-JEPA 2 ViT-L/16 checkpoint as the visual representation backbone and fine-tune it on LIBERO demonstration videos from the agentview camera. The model follows the standard online-target encoder design of V-JEPA 2, where the target encoder is updated by EMA. During training, we use short video clips cropped to 256×256 and apply a first-tubelet-visible

masking strategy, forcing the predictor to infer future latent tokens from the initial visible spatio-temporal context. To incorporate task information, we extract language features with a frozen CLIP ViT-B/32 text encoder [81] and inject them into the predictor through zero-initialized cross-attention layers. Since V-JEPA 2 predicts latent representations rather than pixels, we additionally train a lightweight convolutional decoder for RGB reconstruction. The encoder-predictor stack is frozen when training this decoder. To improve robustness to errors in predicted latents, we perturb the training latents with Gaussian noise. The decoder is optimized with an L1 reconstruction loss over video frames and an LPIPS [78] loss on temporally subsampled frames.

LDP [50]. We implement LDP based on the official codebase and make several modifications for fair comparison under high-resolution LIBERO video modeling. Specifically, we increase the visual input resolution to 256×256 and enlarge the visual token budget in the planner/IDM pipeline accordingly. To handle the high-dimensional SD-VAE latent representation, we introduce a spatial convolutional projection before policy learning, which compresses the latent feature map and stabilizes training. We also add explicit goal-image conditioning by encoding the goal frame with the same VAE and injecting the goal latent into the planning module. During inference, LDP performs open-loop rollout, where future latent trajectories are autoregressively predicted without intermediate environment feedback. These settings are used consistently for all reported LDP results.

A.4 Policy Learning Baselines

GLCBC. GLCBC is a goal-conditioned language behavioral cloning baseline implemented within the LBP policy architecture but without the high-level planner. The current image observation is encoded by a ResNet-34 [82], while the language instruction is encoded by CLIP [81]. In addition, we use a predefined final goal image as goal conditioning and encode it with the DecisionNCE image encoder [83]. The language and goal-image embeddings are concatenated and injected into the policy through a FiLM [74] module to modulate the visual features. Finally, the low-level diffusion policy takes the conditioned visual semantics and current proprioception as inputs, and predicts the diffusion noise for action generation.

SuSIE [31]. We utilize the official codebase with minimal modifications, altering only the datasets. For simulation experiments, we fine-tune the released SuSIE checkpoint on the LIBERO dataset.

Seer [76]. We use Seer as a vision-language-action baseline following its original implementation. The model predicts future visual representations from the current observation and language instruction, and then uses an inverse-dynamics head to predict actions conditioned on both the current observation and the predicted future features.

VPP [43]. VPP learns an implicit inverse dynamics model conditioned on predicted future representations inside VDMs. To predict a more precise future, VPP fine-tunes the pre-trained video foundation model on robot datasets along with internet human manipulation data. We follow the official codebase to finetune it on the LIBERO dataset.

A.5 Computation Resource

We train models on 8 A100 GPUs and evaluate infernece time in Tab. 1 on 1 A100 GPU.

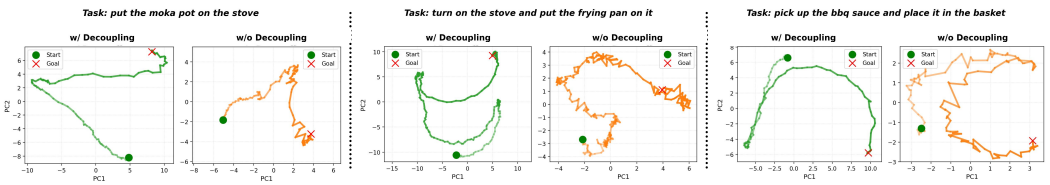


Figure 7: Latent trajectory visualization of ablation on dynamical representation decoupling.

Table 5: Comparisons on the choices of vision feature encoder.

Encoder	Short Fragment @16frames		Long Fragment @64frames	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
LIV	19.90	0.158	18.25	0.237
SigLIP 2	20.38	0.133	19.21	0.168
DINOv2 (Ours)	**20.53**	**0.109**	**19.46**	**0.134**

Table 6: Ablation on the number of latent tokens.

# Tokens	Short Fragment @16frames		Long Fragment @64frames	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
4	**20.99**	0.114	**19.68**	0.148
1 (Ours)	20.53	**0.109**	19.46	**0.134**

Table 4: Ablations on dynamical representation decoupling and first-order supervision

Method	Short Fragment @16frames			Long Fragment @64frames		
	PSNR↑	LPIPS↓	FPS↑	PSNR↑	LPIPS↓	FPS↑
w/o dynamical representation decoupling	**20.65**	0.127	7.301	**19.53**	0.212	2.290
w/o first-order supervision	12.71	0.751	**33.67**	-	-	-
ODEWorld (Ours)	20.53	**0.109**	**33.67**	19.46	**0.134**	**13.83**

B Ablation Studies

B.1 Ablation on Dynamical Representation Decoupling

To validate the effectiveness of our architectural design, we evaluate the impact of the dynamical representation decoupling module in ODEWorld. We observe that without explicitly isolating the static context s_0 from the state space, the learned latent trajectories become significantly non-smooth and jittery as shown in Fig. 7. The primary reason is that not all nuanced visual features in a raw video observation can be seamlessly modeled as an ODE function. Raw feature spaces inherently contain a substantial amount of noisy, time-invariant, and non-ODE-compatible information, such as static backgrounds and object textures. Forcing a continuous velocity field to fit these static details fundamentally disrupts the dynamics learning process. As shown in Tab. 4, the proposed decoupling strategy not only significantly improves inference efficiency (significantly higher FPS) but also enhances perceptual consistency (much lower LPIPS scores, especially for long-horizon prediction). We note that the non-decoupled variant attains marginally higher PSNR, which is expected rather than contradictory: as an MSE-based pixel-level metric, PSNR is dominated by large static regions (e.g., backgrounds and textures) that this variant retains and reconstructs faithfully, whereas its worse LPIPS reflects the temporal and semantic inconsistency that such per-frame fidelity does not capture. Consequently, without decoupling, the latent structure fails to effectively scaffold the direct first-order supervision to optimize the velocity field. By introducing the initial state-conditioned encoder and decoder, our decoupling mechanism ensures that the latent representation focuses exclusively on essential dynamical changes, which is crucial for learning a well-behaved, temporally smooth, and physically meaningful latent flow.

B.2 Ablation on First-Order Supervision

To verify the effectiveness of the proposed first-order supervision, we implement consistency loss below to replace the reconstruction loss $\mathcal{L}_{\text{dyn-recon}}$ and velocity loss $\mathcal{L}_v$:

$$\mathcal{L}_{\text{consistency}} = \mathbb{E}_{s_0, s_t \sim \mathcal{D}, k \in \{1, \cdots, K\}, t} \left\| g_{\text{dyn}} \left(z_t + \int_t^{t+k} v_\theta(z_t, t; z_0, c) dt; s_0 \right) - s_{t+k} \right\|^2 \tag{7}$$

where $z_t = f_{\text{dyn}}(s_t; s_0)$, $z_0 = f_{\text{dyn}}(s_0; s_0)$ and $K = 4$. Specifically, this consistency-loss variant minimizes the error of reconstructing the subsequent K frames obtained by planning through the ODE latent space, which has been adopted in many works on neural ODE [65, 84]. In Tab. 4, we

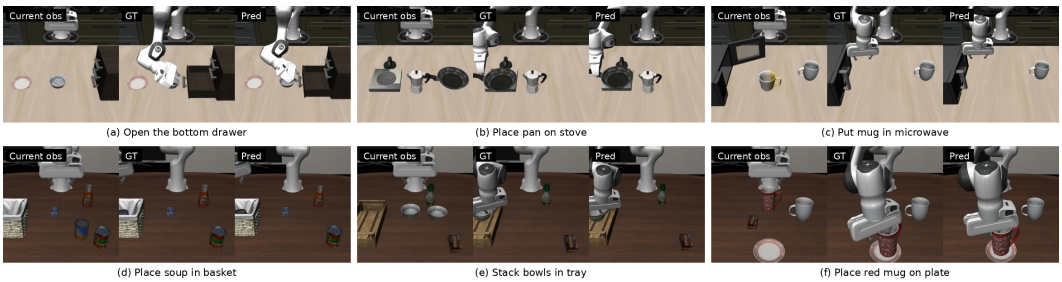


Figure 8: Qualitative results of the goal image predictor. Given the current observation and a language instruction, the goal image predictor generates a semantically consistent goal image for ODEWorld.

Table 7: Comparison between ground-truth image goals and predicted language goals.

Goal Conditioning	Short Fragment @16frames		Long Fragment @64frames	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Predicted Goal Image	21.06	0.110	19.91	0.142
GT Goal Image	20.53	0.109	19.46	0.134

observe that it suffers from optimization difficulty, yielding poor short-horizon prediction and visually implausible long-horizon rollouts (so we fill those results in Tab. 4 with "-"). We attribute this to two aspects. First, the consistency loss couples two distinct objectives — prediction and reconstruction — forcing the model to learn a latent space that is simultaneously reconstructable and ODE-plannable. Second, this coupling is especially harmful in our setting: unlike standard dynamical systems where the goal is to fit one specific governing ODE, ODEWorld only needs to fit some admissible ODE in the latent space, so the target is underdetermined and the entangled objective adds too strict constraints to settle on a consistent solution. In contrast, ODEWorld decouples these objectives via a reconstruction loss and a direct first-order supervision loss, optimizing each independently, which leads to more stable optimization and, in turn, better video generation quality.

B.3 Ablations on Vision Encoders

We examine how ODEWorld is compatible with different choices of vision feature extractors by replacing DINOv2 with SigLIP 2 [85] and LIV [86]. As shown in Tab. 5, ODEWorld maintains robust video generation quality across all three encoders. While the gaps are small, DINOv2 achieves the best overall performance on both short- and long-horizon prediction, consistent with its widely recognized effectiveness as a visual encoder.

B.4 Ablation on Token Numbers

We increase the number of tokens from 1 to 4 and keep other configurations unchanged. Results in Tab. 6 show comparable performance between 1- and 4-token models, indicating that a *single* latent token is already sufficient to achieve strong performance across the evaluated scenarios. This also indicates that our key designs make ODEWorld capable of efficiently capturing the underlying dynamics from pixels with very compact latent representation.

B.5 Ablation on Goal Descriptions

ODEWorld naturally supports image goals, where the target image provides explicit visual guidance for future dynamics prediction. However, in practical scenarios, goals are often specified by language instructions. To extend ODEWorld to language-based goal descriptions, we introduce a lightweight goal image predictor to bridge the language and visual modalities.

Specifically, since ODEWorld adopts DINOv2 as the visual backbone rather than a large-scale vision-language model, we avoid introducing additional multimodal alignment complexity into the dynamics model. Instead, we train a lightweight goal image predictor consisting of four cross-attention layers, which takes the current observation and language instruction as inputs and predicts the corresponding goal image. As shown in Fig. 8, the goal image predictor generates semantically meaningful goal images from the current observation and language instruction, providing effective visual guidance for ODEWorld.

Table 8: Ablation on the detach in velocity learning.

Methods	Short Fragment @16frames		Long Fragment @64frames	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
w/o detach	21.18	0.107	20.00	0.123
w/ detach	20.53	0.109	19.46	0.134

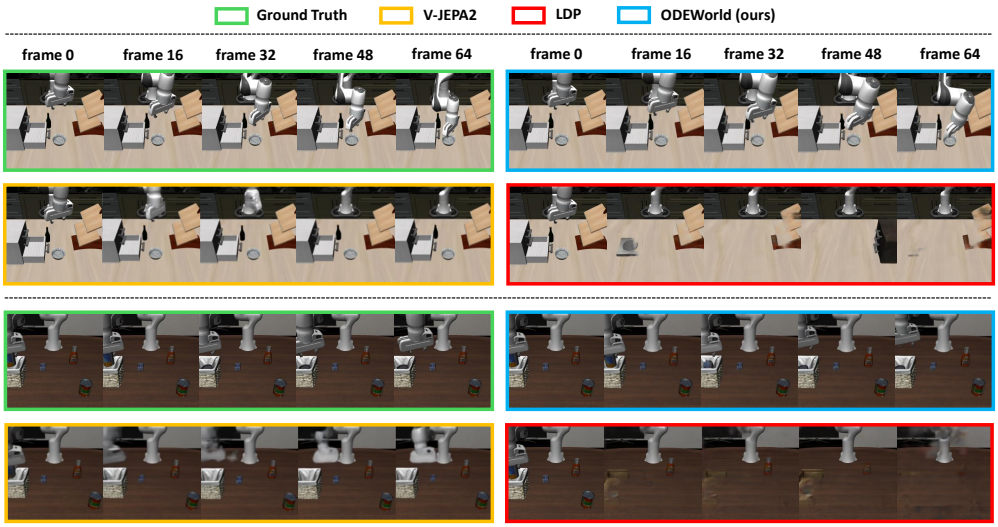


Figure 9: ODEWorld enables long-horizon video generation in the LIBERO simulation environment, producing temporally coherent and physically plausible videos that closely match the ground-truth observations. In contrast, our baseline methods LDP and V-JEPA 2 often suffer from blurry visual predictions and dataset-induced hallucinations in long-horizon generation, especially at later frames, such as frame 64.

We further evaluate ODEWorld using the predicted goal images for goal conditioning and compare it with the standard setting using ground-truth goal images. As reported in Tab. 7, the two settings achieve nearly identical video prediction performance on both short- and long-horizon rollouts. These results demonstrate that the proposed goal image predictor effectively bridges language and visual goal representations, enabling ODEWorld to naturally support language-specified goals without sacrificing prediction quality.

B.6 ODEWorld without Detach in Velocity Learning

In PT-Flow, we use the stop-gradient operation $\mathtt{sg}(\cdot)$ on the target latent velocity to stabilize the optimization, following common practices in representation learning [69]. Interestingly, we find that PT-Flow remains effective even without stop-gradient, as shown in Tab. 8. This suggests that the proposed objective itself can provide effective constraints for learning meaningful representations, as the first-order velocity constraint directly regularizes the temporal evolution in latent space.

C Analysis of Representation Collapse

Following RankMe [87], we use the effective rank of mean-centered representations to assess potential representation collapse. We randomly sample 5,000 representations across planning horizons $h \in \{1, 4, 8, 16, 32\}$. As shown in Tab. 9, ODEWorld's 768-dimensional dynamics latent z consistently maintains an effective rank above 400 across all horizons. In Tab. 10, we further compare ODEWorld with V-JEPA 2 representations and DINOv2 CLS features under the same sampling protocol. ODEWorld achieves a substantially higher effective rank than V-JEPA 2 and DINOv2, indicating that its learned latent dynamics representations remain diverse and expressive with a lower risk of representation collapse.

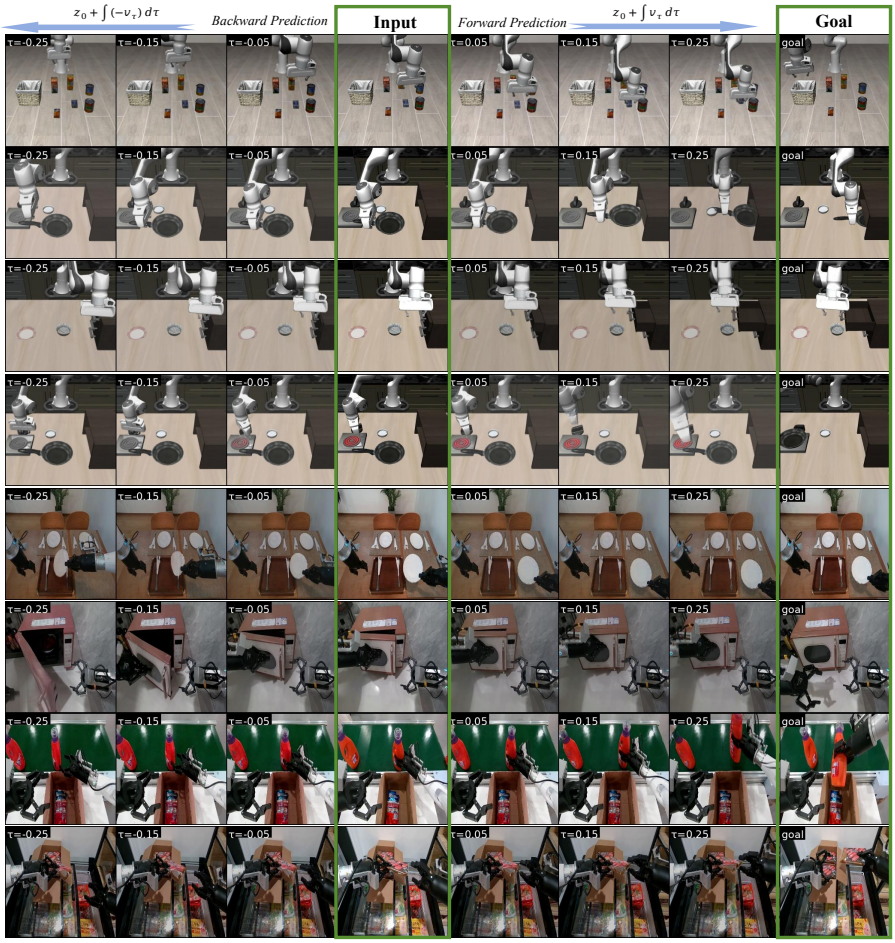


Figure 10: Additional visualization results of bidirectional prediction.

Table 9: Centered RankMe effective rank analysis of ODEWorld dynamics latents across different planning horizons.

Horizon h	1	4	8	16	32
Effective Rank / Dim.	417.2/768	413.9/768	422.1/768	433.1/768	439.5/768

Table 10: Centered RankMe effective rank comparison with V-JEPA 2 and DINOv2 representations.

Representation	Dimension	Effective Rank (Mean)
V-JEPA 2	1024	203.7
DINOv2 CLS	768	376.1
ODEWorld z	768	425.2

D Additional Results

D.1 Additional Video Generation Quality Comparisons

In Fig. 9, we provide additional comparisons for long-horizon video generation in the LIBERO simulation environment. These visualizations complement the main paper and further illustrate that ODEWorld can produce temporally coherent and physically plausible rollouts, while baseline methods often suffer from visual degradation and temporal inconsistency over extended horizons.

D.2 Additional Bidirectional Video Generation Results

In Fig. 10, we show additional bidirectional video generation results on both LIBERO and AgiBot World. We perform closed-loop planning that updates the initial state every integration step $\tau = 0.05$. When we plan backward, we integrate velocity $-v_\theta(z_0, 0; z_0, c)$ for every integration step and reset the planned latent state as z_0 for the next integration step.

E Limitations

Although our proposed framework provides a promising architecture for continuous-time predictive modeling, the current version of ODEWorld still has several limitations and can be further enhanced in future research. First, while the proposed ODEWorld demonstrates some impressive capabilities and efficiency, its scale and amount of training data (two robotics datasets) remain small as compared to many large video-based world models. This limitation stems primarily from computational resource constraints. Future studies can be conducted to further scale the model size and train on larger and more diverse datasets. Second, our current version of ODEWorld uses a frozen DINOv2 [70] image encoder to encode raw observations. As DINO is designed to encode single-frame images without much consideration for temporal consistency, it could produce noisy and temporally inconsistent embeddings for sequential modeling. In our study, we introduce the Savitzky–Golay derivative filter [75] to enhance the smoothness and approximation quality of the ground truth velocity target to mitigate the deficiencies of the DINO encoder. Further investigations can consider replacing DINO with an image encoder with temporal consistency design [83, 88] or jointly tuning the DINO encoder during model training. Lastly, the current version of ODEWorld does not incorporate action conditioning and primarily focuses on video generation. Future studies can consider injecting action conditioning into a pre-trained ODEWorld model, which can further unlock its capability for control tasks.