PhysMani: Physics-principled 3D World Model for Dynamic Object Manipulation

Peng Yun[1], Shouwang Huang[1], Hao Li[1], Jinxi Li[1], Jianan Wang[2], and Bo Yang[1]✉

[1] vLAR Group, The Hong Kong Polytechnic University, Hong Kong SAR, China
[2] Astribot, China ✉ corresponding author
{peng-aae.yun,bo.yang}@polyu.edu.hk

Abstract. Manipulating fast and dynamically moving targets in unstructured 3D environments remains challenging for embodied AI. Existing visual-language-action models and world models struggle with accurate 3D geometry and physically meaningful forecasting. We propose **PhysMani**, a framework that couples a physics-principled 3D Gaussian world model with a future-aware action policy model. The world model learns a divergence-free Gaussian velocity field via online optimization for fast and physically grounded future dynamics prediction. The policy model integrates the predicted 3D scene future dynamics through a learnable token based cross-attention module. We introduce PhysMani-Bench, a dynamic manipulation benchmark with 16 tasks, and demonstrate a superior success rate over strong baselines in both simulation and real-world robot experiments. Our code and data are available at https://github.com/vLAR-group/PhysMani

Keywords: Physical World Model · Dynamic Manipulation

1 Introduction

With the advancement of visual-language-action models (VLAs) [29, 48] and world models [17, 39, 68], embodied AI has seen tremendous progress in recent years, enabling robots to learn complex behaviors across diverse tasks. While achieving excellent performance, these models predominantly focus on static or quasi-static tasks, leaving the manipulation of dynamic objects underexplored.

To date, only a few works have begun to address the challenge of manipulating dynamic objects, such as playing table tennis [16], playing soccer [33], or grasping moving objects [49, 52] on conveyor belts [3, 65, 67] or table surfaces [61], *etc*. Despite achieving encouraging results, they are often limited to either specific or oversimplified scenarios. As a result, they fail to provide a general framework for dynamic object manipulation in open and unstructured environments, where objects or targets may exhibit complex motion patterns.

In general dynamic interaction scenarios, such as catching a thrown ball, loading an item into a moving container, or placing a plate onto a rotating table, a robot must quickly and accurately anticipate future dynamics, which are

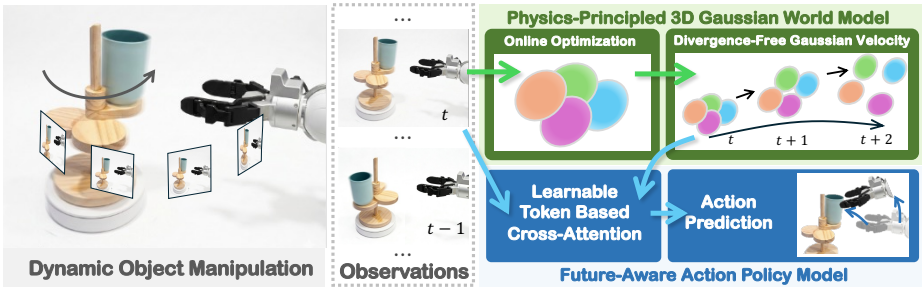


Fig. 1: The overall framework of our method. It features a physics-principled 3D Gaussian world model and a future-aware action policy model.

governed by physical laws, and use those predictions to guide precise actions, all in 3D space. To accomplish such tasks, a potential strategy is to leverage existing powerful VLAs [55] or video-based world models [17] to jointly predict future states and action policies. However, these models currently fall short in three critical aspects: 1) their learned world representations often lack explicit awareness of 3D scene geometry; 2) their generated future frames, while visually realistic, frequently fail to capture physically meaningful dynamics [51]; 3) their chained large visual and language models often incur significant inference latency, which is prohibitive for time-sensitive dynamic manipulation.

To tackle these issues, we propose **PhysMani**, a new framework for manipulating dynamic targets in general environments. As illustrated in Figure 1, our framework features two key designs: 1) a **physics-principled 3D Gaussian world model** that continuously optimizes and forecasts future 3D scene dynamics governed by fundamental physical laws; 2) a **future-aware action policy model** that seamlessly integrates predicted future dynamics into decision-making. These two components are realized in a lightweight pipeline that enables rapid future prediction and low-latency action execution.

For our physics-principled 3D world model, we build upon recent advances in 3D Gaussian-based physics learning, notably FreeGave [37]. Particularly, we introduce an online optimization mechanism that efficiently predicts a divergence-free per-Gaussian velocity field, thereby guaranteeing that the anticipated future scene dynamics adhere to fundamental physical laws.

For our future-aware action policy model, we adopt a simple yet effective cross-attention mechanism with learnable tokens to seamlessly integrate the predicted 3D scene future dynamics into decision-making. Specifically, for each 3D Gaussian in the dynamic environment, we attentively incorporate the estimated velocity information from its local spatial neighborhood into the policy network, so that the robot's predicted future actions are explicitly conditioned on the anticipated future motion of dynamic targets.

Since existing robotic benchmarks overwhelmingly focus on static or quasi-static manipulation and research on dynamic target manipulation is still in its infancy, we introduce **PhysMani-Bench**, a moderate-sized dynamic manipulation benchmark with 16 tasks, inspired by the widely-used static benchmark RLBench [27], to evaluate our method. Overall, our contributions are:

- We introduce a new framework and the first benchmark with 16 tasks for manipulating targets in general dynamic scenarios.
- Our framework features a physics-principled 3D Gaussian world model for forecasting future dynamics and a future-aware action policy model that integrates predicted future dynamics.
- We demonstrate superior manipulation success rates on both our newly introduced simulation benchmark and real-world experiments, clearly outperforming all baselines in terms of effectiveness and efficiency.

2 Related Work

VLAs and World Models: By leveraging advances in large language models (LLMs) [1, 22, 41, 62] and vision-language models (VLMs) [44, 54], vision-language-action models (VLAs) [10, 29, 32, 53, 64, 69] have recently attracted significant attention in robotics. By jointly learning visual, language, and action modalities in an end-to-end framework, VLAs aim to enable robots to perform a wide range of tasks across diverse scenarios. Powered by Transformer architectures [58] and/or diffusion models [25], VLAs have achieved strong success rates on many everyday static or quasi-static tasks. However, they often lack a comprehensive understanding of action and world dynamics, which limits their ability to perform foresight planning necessary for dynamic robotic tasks.

By learning to forecast future states conditioned on current observations and actions, world models [23, 24] aim to capture environment dynamics and thereby inform decision-making. Recently, rapid progress in large-scale video generation [2, 6, 11, 34, 60] has revealed their potential as world models [4, 5, 8, 12, 13, 57, 63], largely due to powerful diffusion Transformer architectures. Although these models can produce visually compelling future frames, the predicted dynamics are limited to 2D image space and often violate basic physical laws, making them unreliable for precisely guiding robotic actions in 3D space. Moreover, the use of chained large models typically results in slow inference, which hinders rapid forecasting and action execution in dynamic interaction scenarios.

Manipulation of Dynamic Targets: The recent VLAs and world models predominantly focus on static and quasi-static robotic tasks, leaving the manipulation of dynamic targets largely underexplored. Early efforts on dynamic robotic manipulation are mainly narrowed on specific tasks, including playing soccer [33], playing table tennis [16], object handovers [14, 45, 49, 52], grasping moving objects on conveyor belts [3, 65, 67] or table surfaces [61], *etc.* While achieving promising results, these methods do not provide a general framework for tackling dynamic manipulation in broader settings, where objects often exhibit complex and diverse physical dynamics. Moreover, most existing benchmarks and datasets [27, 42, 50] focus on static or quasi-static tasks, inherently hampering the study of dynamic manipulation.

Visual Physics Learning for Robotics: Obtaining an accurate physical model of complex, dynamic 3D environments is crucial for embodied manipulation, especially in highly interactive settings. One line of work [7, 38, 40] relies on

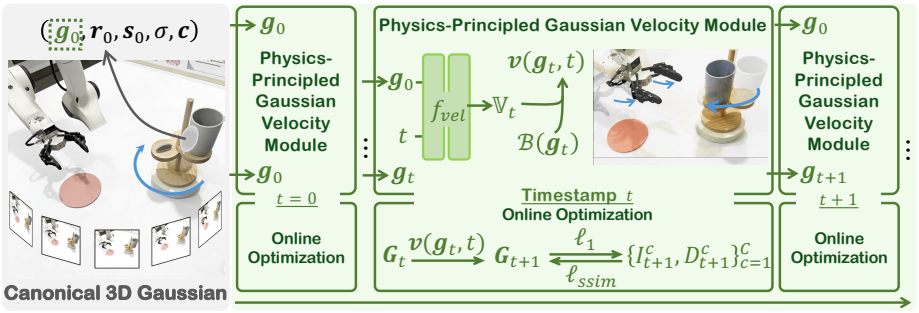


Fig. 2: The left panel shows the canonical 3D Gaussian module. The top-right panel shows the physics-principled Gaussian velocity module, and the bottom-right panel illustrates the online optimization process.

physics simulators [15] to compute object-centric dynamics from known physical properties, but such approaches are typically restricted to specific object categories and lack generality across diverse objects. More recently, another line of research [35–37] has focused on learning 3D dynamics, particularly 3D velocity fields, directly from videos without requiring additional priors, offering a promising avenue for robots to predict future dynamics that adhere to fundamental physical laws.

3 PhysMani

3.1 Overview

Our framework consists of two parallel models. The physics-principled 3D Gaussian world model is designed to continuously learn the 3D geometry, appearance, and, most importantly, physical dynamics of the target dynamic environment. For generality, we treat the embodiment as an intrinsic part of the dynamic environment rather than modeling it separately. While the world model predicts 3D scene future dynamics, our future-aware action policy model takes the current visual observations together with the predicted future dynamics as input and infers robot's future actions for execution.

3.2 Physics-principled 3D Gaussian World Model

Given a dynamic environment for robot manipulation, our world model aims to learn both the 3D scene structure and dynamics. We draw inspiration from the recent FreeGave [37] whose key innovation is a divergence-free Gaussian velocity field for future frame prediction while respecting basic physical laws. However, its complex pipeline, relying on a separate deformation network and offline optimization, makes it impractical for real-time dynamic manipulation. To this end, we substantially refine FreeGave's architecture and reformulate it as an online optimization model. For clarity and fluency, we present necessary details of the three main modules as follows.

Algorithm 1 Online Optimization of Future Dynamics.

Preliminary: Canonical 3D Gaussians: $\{\cdot\cdot(\boldsymbol{g}_0, \boldsymbol{r}_0, \boldsymbol{s}_0, \sigma, \boldsymbol{c}, \boldsymbol{z})\cdot\cdot\}$;
while true **do**

• receiving new RGB/Ds, denoted by $\{I_t^c, D_t^c\}_{c=1}^C$, at time t, where $t \leftarrow t' + 1$;

- Note that, here t' denotes the previous time step, while t denotes the current time step, which may correspond to the immediately subsequent frame or to a frame after several intervals following t'.

• getting $\boldsymbol{v}(\boldsymbol{g}_t, t)$ from the Gaussian velocity module;

• applying $\boldsymbol{v}(\boldsymbol{g}_t, t)$ to drive the corresponding Gaussian from t' to t using the same interleaved mid-point method proposed in FreeGave [37];

• optimizing the Gaussian's position $\boldsymbol{g}_t$, orientation $\boldsymbol{r}_t$, and f_{vel}, based on ℓ_1 and ℓ_{ssim} losses between rendered RGB/Ds and $\{I_t^c, D_t^c\}_{c=1}^C$. This gradient-based optimization lasts T iterations considering the trade-off between accuracy and efficiency;

- Note that the above optimization stage is primarily conducted for learning physically meaningful basic components $\mathbb{V}_t$ and velocity field $\boldsymbol{v}(\boldsymbol{g}_t, t)$ for each Gaussian.

• refining the Gaussian properties for a few additional T' iterations using the same ℓ_1 and ℓ_{ssim} losses, while freezing the velocity network f_{vel}. This compensates for potential minor changes in scene geometry and appearance over time.

• **Output**: the six basic velocity components $\mathbb{V}_t$ and $\boldsymbol{v}(\boldsymbol{g}_t, t)$ for every Gaussian
end while

- In our experiments, we set $T = 50$ and $T' = 7$ iterations. Each round of optimization takes $\sim 200ms$ on a single RTX4090 GPU.

Canonical 3D Gaussian Module: As shown in the left block of Figure 2, at time $t = 0$, given RGB/Ds from C cameras, denoted by $\{(I_0^c, D_0^c)\}_{c=1}^C$, typically mounted at fixed and known poses in standard robot manipulation setups such as RLBench [27] and CALVIN [50], we first initialize a set of 3D Gaussian kernels $\boldsymbol{G}_0$ to represent the canonical scene geometry and appearance at time $t = 0$ based on the sparse point cloud back-projected from C depth views. Each Gaussian is parameterized by a 3D position $\boldsymbol{g}_0$, scaling $\boldsymbol{s}_0$, covariance matrix converted from quaternion $\boldsymbol{r}_0$, opacity σ, and color $\boldsymbol{c}$. Following 3DGS [31], we then optimize these canonical kernels using the C RGB images at time $t = 0$ via ℓ_1 and ℓ_{ssim} losses as used in 3DGS:

$$\underbrace{\{\cdot\cdot(\boldsymbol{g}_0, \boldsymbol{r}_0, \boldsymbol{s}_0, \sigma, \boldsymbol{c})\cdot\cdot\}}_{\boldsymbol{G}_0} \underset{\ell_1+\ell_{\text{ssim}}}{\overset{\text{render}}{\rightleftarrows}} \{I_0^1 \cdot\cdot I_0^c \cdot\cdot I_0^C\} \tag{1}$$

Physics-principled Gaussian Velocity Module: As shown in the top-right block of Figure 2, for each Gaussian at current time t, we follow FreeGave to use MLPs, denoted by f_{vel}, to learn six basic velocity components: $\mathbb{V}_t \leftarrow [v_t^x, v_t^y, v_t^z, w_t^z, w_t^y, w_t^x]$, where $v_t^x/v_t^y/v_t^z$ represent linear velocities and $w_t^z/w_t^y/w_t^x$ angular velocities. The velocity field, denoted by $\boldsymbol{v}(\boldsymbol{g}_t, t)$, is composed as follows:

$$\boldsymbol{v}(\boldsymbol{g}_t, t) = \mathbb{V}_t \cdot \mathcal{B}(\boldsymbol{g}_t), \quad \mathbb{V}_t = f_{vel}(\boldsymbol{g}_0, t), \quad \mathbb{V}_t \in \mathcal{R}^{1\times 6} \quad \mathcal{B}(\boldsymbol{g}_t) \in \mathcal{R}^{6\times 3} \tag{2}$$

where $\mathcal{B}(\boldsymbol{g}_t)$ is a set of basis vectors and $g_t^x/g_t^y/g_t^z$ are coordinates of $\boldsymbol{g}_t$:

$$\mathcal{B}(\boldsymbol{g}_t) = \begin{bmatrix} 1\,0\,0 & -g_t^y & g_t^z & 0 \\ 0\,1\,0 & g_t^x & 0 & -g_t^z \\ 0\,0\,1 & 0 & -g_t^x & g_t^y \end{bmatrix}^T \tag{3}$$

A key advantage of this Gaussian velocity module is that it enforces a divergence-free property, as proved in the FreeGave paper.

Online Optimization of Future Dynamics: In the context of dynamic manipulation, a stream of RGB/Ds is continuously available, and our world model must be optimized rapidly and online so that it can capture the ongoing dynamics of the target and environment, and accurately forecast future evolution. To achieve this, we remove the auxiliary deformation field and the deformation-aided optimization strategy in FreeGave [37]. Instead, we propose the following online optimization Algorithm 1 to learn precise velocity components $\mathbb{V}_t$ and field $\boldsymbol{v}(\boldsymbol{g}_t, t)$ for every Gaussian at time t. More details about the network and optimization of our world model are provided in Appendix A.1.

3.3 Future-aware Action Policy Model

Our action policy model aims to leverage both the current visual observations and the predicted future dynamics to infer future actions. For generality, we adopt the existing 3D FlowMatch Actor (3DFA) [20] as our policy network backbone, though others could also be used. In fact, 3DFA is built on 3D Diffuser Actor (3DDA) [30] by replacing DDPM-based diffusion with Rectified Flow [46] for fast training and inference, without touching the network architecture. Our policy model consists of the following components.

Encoding 3D Visual Geometry, Appearance and Language: As illustrated in the left block of Figure 3, we follow 3DFA [20] to process the posed RGB/Ds at current time t, lifting 2D visual features to a sparse 3D scene point cloud $\boldsymbol{P}_t \in \mathcal{R}^{4096\times 3}$ converted from depth views, getting a feature cloud or a cloud of visual tokens, denoted by $\boldsymbol{F}_t^{vis} \in \mathcal{R}^{4096\times 120}$. We also encode the language task instruction into a set of tokens, denoted by $\boldsymbol{F}_t^{lan} \in \mathcal{R}^{53\times 120}$. Eventually, these two sets of tokens are fused via cross-attention, obtaining a cloud of new tokens, denoted by $\tilde{\boldsymbol{F}}_t^{vis} \in \mathcal{R}^{4096\times 120}$. More details can be found in 3DFA.

Incorporating 3D Scene Future Dynamics: At current time t, our world model predicts six basic velocity components $\mathbb{V}_t$ for each Gaussian, yielding the future dynamics of the entire 3D scene, denoted by $\mathbf{D}_t \in \mathcal{R}^{H\times 6}$, where H is the total number of 3D scene Gaussians $\boldsymbol{G}_t$. We set $H = 30000$ in all experiments.

As shown in the right block of Figure 3, we incorporate the scene dynamics $\mathbf{D}_t$ into visual tokens $\tilde{\mathbf{F}}_t^{vis}$ of the sparse 3D scene point cloud $\boldsymbol{P}_t$ as follows:

- Step #1: For each point $\boldsymbol{p}_t$ in the 3D scene point cloud $\boldsymbol{P}_t$, we retrieve its K nearest neighboring 3D Gaussians from our world model via KNN, based on Euclidean distance between point and Gaussian coordinates.
- Step #2: For each of the K nearest Gaussians $\{\boldsymbol{g}_t^1 \cdots \boldsymbol{g}_t^k \cdots \boldsymbol{g}_t^K\}$ associated with the center point $\boldsymbol{p}_t$, we compute the relative offset: $\Delta\boldsymbol{p} = \boldsymbol{p}_t - \boldsymbol{g}_t^k$.

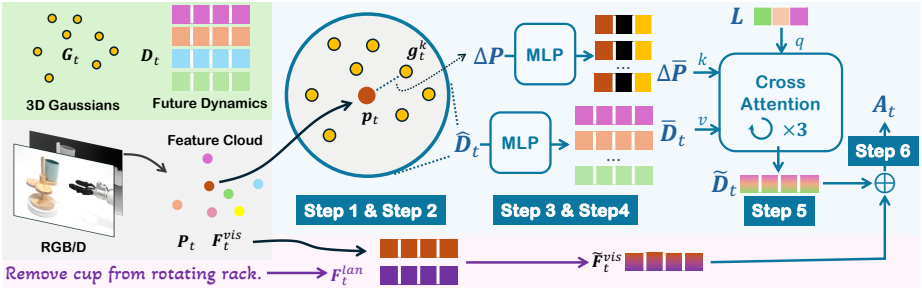


Fig. 3: The left block shows the encoding of visual observations and language. The right block shows the incorporation of 3D scene future dynamics for action prediction.

- Step #3: For the entire 3D scene point cloud $\boldsymbol{P}_t$, we similarly compute relative offsets to all K neighboring Gaussians, yielding $\Delta\boldsymbol{P} \in \mathcal{R}^{4096\times K\times 3}$, and retrieve their basic velocity components, denoted by $\hat{\boldsymbol{D}}_t \in \mathcal{R}^{4096\times K\times 6}$.
- Step #4: Both $\Delta\boldsymbol{P}$ and $\hat{\boldsymbol{D}}_t$ are fed into two separate MLPs to obtain two sets of tokens, denoted by $\Delta\bar{\boldsymbol{P}} \in \mathcal{R}^{4096\times K\times 120}$ and $\bar{\boldsymbol{D}}_t \in \mathcal{R}^{4096\times K\times 120}$ respectively.
- Step #5: We introduce a learnable vector $\boldsymbol{L} \in \mathcal{R}^{1\times 120}$ to act as the query token, and regard $\Delta\bar{\boldsymbol{P}}$ and $\bar{\boldsymbol{D}}_t$ as key and value tokens respectively. These key, query, value tokens are fed into three attention blocks, obtaining the final value tokens $\tilde{\boldsymbol{D}}_t \in \mathcal{R}^{4096\times 120}$. Basically, $\tilde{\boldsymbol{D}}_t$ serves as the future dynamics tokens corresponding to the 3D scene point cloud $\boldsymbol{P}_t$.
- Step #6: Lastly, we add dynamics tokens $\tilde{\boldsymbol{D}}_t$ to visual tokens $\tilde{\mathbf{F}}_t^{vis}$, obtaining future-aware tokens $\boldsymbol{A}_t \in \mathcal{R}^{4096\times 120}$, which will be used to infer robot actions.

More details of these incorporation steps are provided in Appendix A.2.

Denoising End-effector's Keyposes with Rectified Flow: Following the neural architecture of 3DFA [20], our action model also aims to predict the end-effector's keyposes, which is commonly adopted in prior works [21,28,43,56]. Particularly, each action keypose $\boldsymbol{a}$ consists of 3D position, 3D orientation, and a binary open/closed state of the end-effector: $\boldsymbol{a} = \{\boldsymbol{a}^{pos} \in \mathcal{R}^3, \boldsymbol{a}^{rot} \in \mathcal{R}^6, \boldsymbol{a}^{open} \in \{0,1\}\}$. At current timestep t, our goal is to predict the corresponding end-effector trajectory $\boldsymbol{\tau}_t = (\boldsymbol{a}^{pos}_{t:t+J}, \boldsymbol{a}^{rot}_{t:t+J})$ of temporal horizon J.

Given a clean training trajectory, denoted by $\boldsymbol{\tau}^0$ where time step t is ignored for simplicity, at a denoising step i, we create the noisy trajectory by a linear interpolation: $\boldsymbol{\tau}^i = (1-i)\boldsymbol{\epsilon} + i\boldsymbol{\tau}^0$, where $\boldsymbol{\epsilon}$ is the sampled noise.

Our model takes as input the noisy trajectory $\boldsymbol{\tau}^i$, proprioception information $\boldsymbol{o}$, language task instruction $\boldsymbol{l}$, and most importantly, our future-aware 3D scene tokens $\boldsymbol{A}$ (subscript time t is also ignored for simplicity), and outputs the velocity field V_{flow} for flow matching and gripper openness f_{open}, respectively.

During training, we follow 3DFA [20] to sample the action time step t uniformly in training trajectories, denoising time step $i \sim \sigma(\mathcal{N}(0,1))$, and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{1})$. We use the ℓ_2 loss to supervise the velocity field for flow matching and binary-cross-entropy (BCE) to supervise the openness:

$$\ell = \ell_2\Big(V_{flow}(\boldsymbol{A},\boldsymbol{o},\boldsymbol{l},\boldsymbol{\tau}^i,i),(\boldsymbol{\epsilon}-\boldsymbol{\tau}^0)\Big) + BCE\Big(f_{open}(\boldsymbol{A},\boldsymbol{o},\boldsymbol{l},\boldsymbol{\tau}^i,i),\boldsymbol{a}^{open}_{t:t+J}\Big) \quad (4)$$

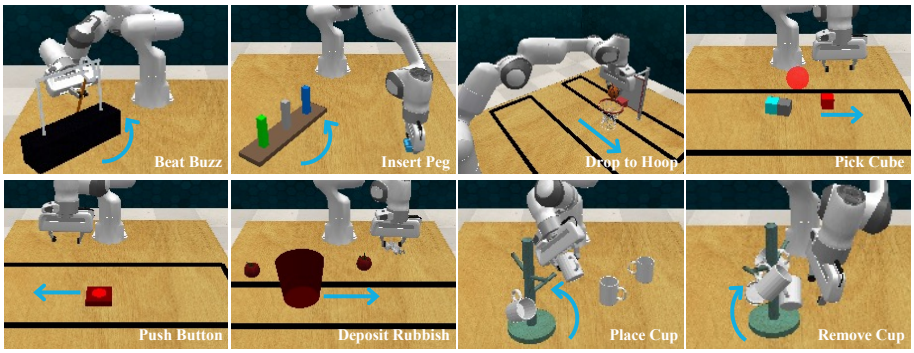


Fig. 4: Examples of diverse and challenging dynamic tasks in our PhysMani-Bench. The blue arrows illustrate the complex motions of dynamic targets.

During inference, given visual observations, our framework continuously predicts the 3D scene's future dynamics and uses them to infer future action keyposes, which are then converted into joint commands via inverse kinematics for execution. More details of the network and training are in Appendix A.2.

3.4 Implementation for Fast Computation

To speed up the overall computation of our framework, we reuse static computation graphs during online optimization of future dynamics. In our setting, the number of 3D Gaussians remains fixed across frames, allowing the optimization to be executed within a static computation graph. By avoiding dynamic graph construction, we significantly reduce CUDA kernel launch overhead and improve GPU utilization. More details of the kernel launch time and overall update latency with and without reusing static graph is in Appendix A.3.

4 PhysMani-Bench

As there is a lack of benchmarks for manipulating targets in general dynamic settings, we carefully extend the widely used RLBench [27] and construct 8 groups of general dynamic tasks, each with normal-speed and high-speed variants, resulting in 16 challenging tasks in total, as briefly described below and illustrated in Figure 4. These tasks cover a diverse range of dynamic reaching, grasping, placing, inserting, pushing and tool-use, with dynamic targets exhibiting complex motions such as linear, circular, and rotational trajectories. While evaluating on a very large-scale benchmark with hundreds or even thousands of tasks is desirable, it is non-trivial. Although recent PhysInOne [66] has begun to realize this scale for visual physics reasoning, scaling our physics-principled evaluation framework to a comparable magnitude is left for future work.

– *beat the rotating buzz at normal/high speed*: The robot must grasp and steer the loop along the rotating wire, maintaining position without making contact.

- *insert ring onto rotating peg at normal/high speed*: The robot must pick up a ring, align and insert it onto a vertical peg attached to a rotating base.
- *drop basketball into moving hoop at normal/high speed*: The robot must pick up a ball and then drop it into a hoop attached to a moving backboard.
- *pick moving cube on table at normal/high speed*: The robot must pick up a moving cube among several other static cubes on the table.
- *push moving button at normal/high speed*: The robot must press a button moving on the table.
- *deposit rubbish into moving bin at normal/high speed*: The robot must pick up the rubbish among other items and deposit it into a moving bin.
- *place cup onto rotating rack at normal/high speed*: The robot must pick up a cup and hang it on the rotating rack.
- *remove cup from rotating rack at normal/high speed*: The robot must grasp a cup from the rotating rack and place it on the table.

For every task, 4 static cameras are mounted around the table, the same as RLBench. The speeds of moving targets are set comparable to the robot's maximum end-effector speed (up to 2 m/s for the Franka Emika Panda 7-DoF arm). This dynamic scenario is highly challenging; reactive control alone is inadequate.

For each task, we collect 160 successful trajectories, randomly split them into training/validation/test sets in a 100/20/40 ratio. The trajectories are gathered using scripted expert policies which produce collision-free and valid executions. All methods are trained on the same training split and evaluated online in simulation using identical configurations in the test split. More details of our PhysMani-Bench are in Appendix A.4.

5 Experiments

We compare against the following representative baselines and evaluate the success rate (**SR**) and future frame prediction quality as primary metrics:

- **Act3D** [19]: It is a voxel-based 3D action prediction method that formulates manipulation as a sequence of perception-driven affordance predictions.
- **3DDA** [30]: It is a diffusion-based imitation learning policy that generates continuous action trajectories conditioned on 3D scene representations. For a fair comparison, we adopt 10 denoising steps to predict its policies, matching the same time budget used by 3DFA and our method.
- **3DFA** [20]: It improves 3DDA by replacing DDPM-based diffusion with Rectified Flow for fast training and inference.
- **3DFA-OF**: A motion-cue variant of 3DFA in which 2D optical flow is concatenated with the RGB input (5 channels in total) and projected back to 3 channels by a convolutional layer before the 3DFA encoder.
- **ManiGaussian** [47]: It is based on dynamic Gaussian splatting to learn 3D scene dynamics as an auxiliary task for improving action prediction.

Table 1: The average success rate (SR) (%) across all 16 tasks of our PhysMani-Bench. The standard deviation is computed over the top 3 models of the last 5 checkpoints.

	Mean SR	**Beat Buzz**	**Insert Peg**	**Drop to Hoop**	**Pick Cube**	**Push Button**	**Deposit Rubbish**	**Place Cup**	**Remove Cup**
Act3D [19]	27.1 ± 2.7	46.7 ± 5.9	58.3 ± 4.7	6.7 ± 1.2	56.7 ± 6.6	49.2 ± 11.2	81.7 ± 5.1	0.0 ± 0.0	21.7 ± 8.5
pi0.5 [9]	8.3 ± 0.2	16.7 ± 3.1	0.8 ± 1.2	0.8 ± 1.2	18.3 ± 10.5	34.2 ± 4.7	8.3 ± 1.2	0.0 ± 0.0	22.5 ± 6.1
ManiGaussian [47]	22.5 ± 0.8	40.0 ± 8.9	35.8 ± 4.2	6.7 ± 6.2	32.5 ± 12.7	18.3 ± 2.4	67.5 ± 2.0	1.7 ± 1.2	61.7 ± 10.5
3DDA [30]	35.1 ± 1.7	60.0 ± 4.1	68.3 ± 6.6	9.2 ± 2.4	63.3 ± 6.6	39.2 ± 6.6	67.5 ± 4.1	15.8 ± 3.1	58.3 ± 3.1
3DFA [20]	37.8 ± 0.9	82.5 ± 2.0	38.3 ± 6.2	30.8 ± 3.1	70.0 ± 6.1	55.8 ± 4.2	53.3 ± 4.2	17.5 ± 4.1	63.3 ± 2.4
3DFA-OF	37.5 ± 1.0	79.2 ± 4.7	45.0 ± 5.4	27.5 ± 5.4	60.8 ± 5.9	54.2 ± 3.1	58.3 ± 4.2	14.2 ± 5.1	58.3 ± 11.2
PhysMani(Ours)	45.9 ± 0.8	78.3 ± 4.2	37.5 ± 6.1	71.7 ± 3.1	84.2 ± 3.1	57.5 ± 3.5	60.0 ± 2.0	15.0 ± 3.5	61.7 ± 6.6
	High Speed **(H)→**	**Beat Buzz (H)**	**Insert Peg (H)**	**Drop to Hoop (H)**	**Pick Cube (H)**	**Push Button (H)**	**Deposit Rubbish (H)**	**Place Cup (H)**	**Remove Cup (H)**
Act3D [19]	–	19.2 ± 5.9	8.3 ± 3.1	3.3 ± 1.2	12.5 ± 2.0	25.8 ± 11.6	31.7 ± 16.6	0.0 ± 0.0	12.5 ± 2.0
pi0.5 [9]	–	3.3 ± 1.2	0.0 ± 0.0	0.0 ± 0.0	5.8 ± 1.2	19.2 ± 2.4	0.8 ± 1.2	0.0 ± 0.0	1.7 ± 2.4
ManiGaussian [47]	–	22.5 ± 3.5	1.7 ± 1.2	0.8 ± 1.2	6.7 ± 4.7	1.7 ± 1.2	37.5 ± 5.4	0.0 ± 0.0	25.0 ± 5.4
3DDA [30]	–	45.8 ± 4.2	25.0 ± 10.2	2.5 ± 2.0	10.0 ± 2.0	20.0 ± 7.4	25.8 ± 5.9	15.0 ± 0.0	36.7 ± 7.2
3DFA [20]	–	52.5 ± 9.4	11.7 ± 5.1	9.2 ± 3.1	21.7 ± 2.4	21.7 ± 3.1	19.2 ± 4.2	13.3 ± 3.1	43.3 ± 6.2
3DFA-OF	–	48.3 ± 8.5	20.8 ± 4.7	6.7 ± 2.4	20.8 ± 3.1	15.8 ± 6.2	28.3 ± 3.1	18.3 ± 6.2	44.2 ± 5.9
PhysMani(Ours)	–	49.2 ± 7.7	25.0 ± 2.0	46.7 ± 4.2	18.3 ± 2.4	20.0 ± 5.4	46.7 ± 4.2	19.2 ± 4.2	43.3 ± 3.1

– **pi0.5** [9]: It is a compact and sophisticated VLA model that predicts action chunks from visual observations and task instructions.

Since our goal is to manipulate targets across general and diverse dynamic scenarios, we evaluate all 16 tasks using a single model of each method rather than training separate models for each task. Specifically, we fine-tune pi0.5 on the entire training set with LoRA [26]. All baselines and our method are well-trained using the same training set of PhysMani-Bench in the same action space, ensuring a fair comparison. For each method, we report success rate (SR) and the standard deviation using the best 3 models of its last five checkpoints in Tables 1&2. More implementation details can be found in the Appendix A.2.

5.1 Evaluation on PhysMani-Bench

Dynamic Manipulation: In Table 1, PhysMani achieves the best performance across most tasks, obtaining a mean SR score 8.1 points higher than the next best method 3DFA. Notably, PhysMani uses 3DFA as its backbone and adopts the same training strategy, differing only in its incorporation of 3D scene future dynamics. It shows the effectiveness of PhysMani with the performance gains directly attributed to our accurate future dynamics predictions (Table 2).

Augmenting 3DFA with 2D optical flow as an explicit motion cue (3DFA-OF) yields no performance gain over the baseline 3DFA and still falls significantly short of PhysMani. This indicates that 2D motion cues alone provide limited dynamic information. Instead, the substantial improvements seen in PhysMani stem from integrating the future dynamics generated by our world model.

The strong VLA model pi0.5 performs poorly on almost all high-speed variants, mainly because it fails to capture the 3D spatial representation and the complex future dynamics required for precise manipulation.

Future Frame Prediction: Table 2 reports PSNR, SSIM, LPIPS, logRMSE (on depth images) and trajectory error to evaluate future frame prediction for

Table 2: Quantitative comparison of future frame prediction on PhysMani-Bench.

	PSNR ↑	logRMSE ↓	SSIM ↑	LPIPS ↓	Traj. Err. ↓
ManiGaussian [47] (next 1^{st} frame)	14.32	-	0.402	0.889	0.388
FreeGave [37] (next 1^{st} frame)	19.47	0.208	0.799	0.208	0.010
PhysMani(Ours) (next 1^{st} frame)	26.90	0.060	0.877	0.123	0.008
FreeGave [37] (next 5^{th} frame)	18.33	0.253	0.780	0.234	0.047
PhysMani(Ours) (next 5^{th} frame)	22.42	0.134	0.829	0.157	0.039
FreeGave [37] (next 10^{th} frame)	17.89	0.269	0.770	0.250	0.086
PhysMani(Ours) (next 10^{th} frame)	20.00	0.189	0.786	0.196	0.074

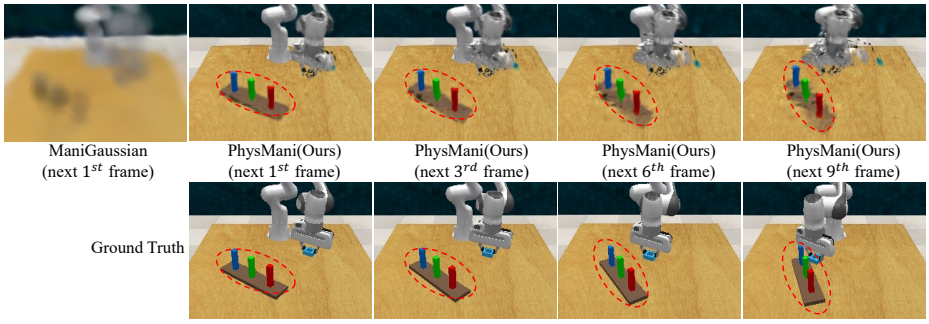
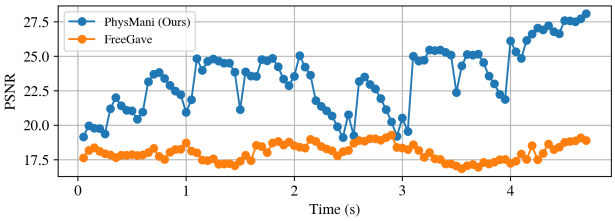


Fig. 5: Qualitative comparison of future frame prediction. Red circles highlight that our method can accurately predict the movement of dynamic targets.

our method and ManiGaussian; the other baselines in Table 1 do not support this. We further compare our method with FreeGave [37], evaluating it online using current and historical observations within the same iteration budget.

For every timestep of each test trajectory across 16 dynamic tasks, PhysMani renders RGB/D predictions from all four cameras for the next 10 timesteps. Since ManiGaussian predicts only one future frame, we render only the next timestep. All predictions are compared with ground truth. Trajectory error assesses whether predictions capture physically meaningful motion, *i.e.*, the Euclidean distance between predicted and ground-truth target centers. Although ManiGaussian is designed for future-frame prediction, it performs relatively poorly because its predictions are inaccurate (PSNR 14.32; Table 2 and Figure 5). In contrast, our physics-principled module enables high-quality future-frame predictions over 10 timesteps.

Fig. 6: The PSNR scores at each timestep, which is the mean over the next 10 future frames. PhysMani requires 205 ms/frame, whereas FreeGave takes 607 ms/frame.



For target trajectory prediction, consecutive frames are captured at 50 ms intervals, meaning the 1st, 5th, and 10th future frames translate to horizons of 50, 250, and 500 ms. Table 2 shows that PhysMani maintains remarkably low errors

Table 3: The success rate (SR) (%) across all 4 real-world dynamic tasks.

	Mean SR	Pick from Belt	Place on Belt	Place on Rack	Remove from Rack
3DFA [20]	45.3	75.0	37.5	6.3	62.5
PhysMani(Ours)	62.5	81.3	62.5	25.0	81.3

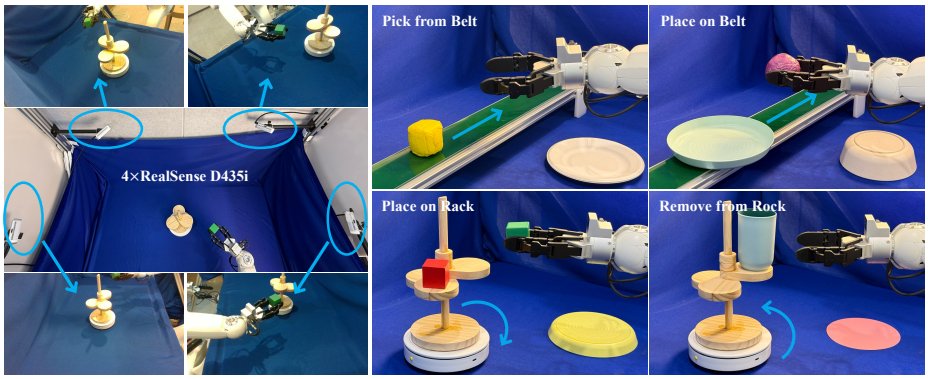


Fig. 7: Illustration of the physical robot setup and real-world dynamic tasks. The robot is Astribot S1 [18].

of 0.008, 0.039, and 0.074 m across these timesteps. In contrast, ManiGaussian produces a substantial error of 0.388 m after just a single 50 ms step.

Figure 6 illustrates that, for a single episode under equal compute constraints, PhysMani runs 3.0× faster and yields higher PSNR scores than FreeGave. Full test set results (Table 2) further demonstrate that under strict online time limits, our pipeline is highly efficient at predicting moving target dynamics from streaming observations compared to FreeGave. This validates the effectiveness and necessity of our modifications to FreeGave, as detailed in Section 3.2.

Generalization Ability: To further evaluate the generalization ability of our policy model pretrained on PhysMani-Bench, we conduct additional experiments on multiple dynamic tasks that extend our original tasks of PhysMani-Bench by varying object shapes and colors, as well as using substantially different speeds and initial positions. Additional results are provided in Appendix A.5.

Overall, all these results demonstrate the physical plausibility of our future predictions and explain the higher success rate over ManiGaussian in Table 1.

5.2 Evaluation on Real-world Tasks

We further evaluate our approach on physical robots in four real-world dynamic tasks as described below. Our system uses a two-fingered robotic arm in a tabletop workspace observed by four fixed RGB-D cameras (Figure 7). All images are recorded at a resolution of 240 × 320, and the control frequency is set to 5 Hz, following common practice in recent real-world manipulation benchmarks [59].

– *picking up a moving toy*: The robot must grasp a toy such as a plastic onion and lemon on a moving belt, and then place it at a target location.

Table 4: The success rate (SR) (%) across all 16 tasks of all ablation experiments. The standard deviation is computed over two runs of the last checkpoint.

	Mean SR	Beat Buzz	Insert Peg	Drop to Hoop	Pick Cube	Push Button	Deposit Rubbish	Place Cup	Remove Cup
(1) removing $\boldsymbol{D}_t$	37.9 ± 0.1	76.2 ± 1.2	43.8 ± 6.2	30.0 ± 2.5	65.0 ± 2.5	47.5 ± 2.5	65.0 ± 0.0	21.2 ± 1.2	58.8 ± 1.2
(2) removing $\boldsymbol{L}$	40.8 ± 1.1	78.8 ± 3.8	31.2 ± 1.2	66.2 ± 1.2	72.5 ± 0.0	58.8 ± 3.8	62.5 ± 10.0	10.0 ± 2.5	58.8 ± 8.8
The Full PhysMani	45.4 ± 0.9	76.2 ± 3.8	43.8 ± 1.2	70.0 ± 2.5	85.0 ± 0.0	53.8 ± 1.2	61.2 ± 1.2	16.2 ± 1.2	50.0 ± 2.5
	High Speed (H)→	Beat Buzz (H)	Insert Peg (H)	Drop to Hoop (H)	Pick Cube (H)	Push Button (H)	Deposit Rubbish (H)	Place Cup (H)	Remove Cup (H)
(1) removing $\boldsymbol{D}_t$	–	48.8 ± 1.2	21.2 ± 1.2	6.2 ± 1.2	18.8 ± 6.2	17.5 ± 5.0	26.2 ± 8.8	20.0 ± 5.0	40.0 ± 0.0
(2) removing $\boldsymbol{L}$	–	42.5 ± 2.5	7.5 ± 5.0	28.8 ± 1.2	21.2 ± 6.2	18.8 ± 6.2	52.5 ± 5.0	6.2 ± 1.2	37.5 ± 5.0
The Full PhysMani	–	55.0 ± 5.0	20.0 ± 7.5	38.8 ± 3.8	21.2 ± 1.2	18.8 ± 1.2	53.8 ± 1.2	15.0 ± 2.5	47.5 ± 7.5

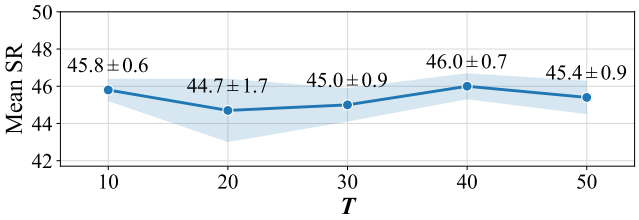


Fig. 8: Mean SR of PhysMani over all 16 tasks under different world model optimization iterations T.

– *placing a toy onto a moving belt*: The robot must pick up a toy such as a plastic onion and lemon, and then place it onto a moving belt.
– *placing a cube onto a rotating rack*: The robot must pick up a cube and then place it on a rotating rack.
– *removing a cup from a rotating rack*: The robot must grasp a cup from the rotating rack and place it on the table.

These tasks involve challenging dynamics, with target speeds set to be comparable to the robot's end-effector speed limits. For each task, we collect 180 training and 60 validation episodes. We train a single model per method across all four tasks, rather than separate models for each task. Each trained model is then evaluated online on the physical system over 64 trials, comprising 16 varying settings such as different target speeds, target objects, and target locations, *etc*. More details of the real-world tasks and experiments are in Appendix A.6.

Table 3 compares our method with the strongest baseline 3DFA on the four real-world dynamic tasks. Results for the other baselines are omitted due to their rather poor performance. Our method consistently outperforms 3DFA by a large margin, demonstrating its effectiveness in real-world dynamic manipulation.

5.3 Ablation Study

Our framework consists of a physics-principled 3D world model and a future-aware policy model. Ablations on PhysMani-Bench validate our design choices.

(1) Removing the 3D scene velocity basic components $\boldsymbol{D}_t$. In this setting, we only keep the neighboring Gaussian coordinates for the center point $\boldsymbol{p}_t$, and feed them into the learnable token based Transformer blocks. Basically, this is to directly evaluate the effectiveness of integrating 3D scene future dynamics.

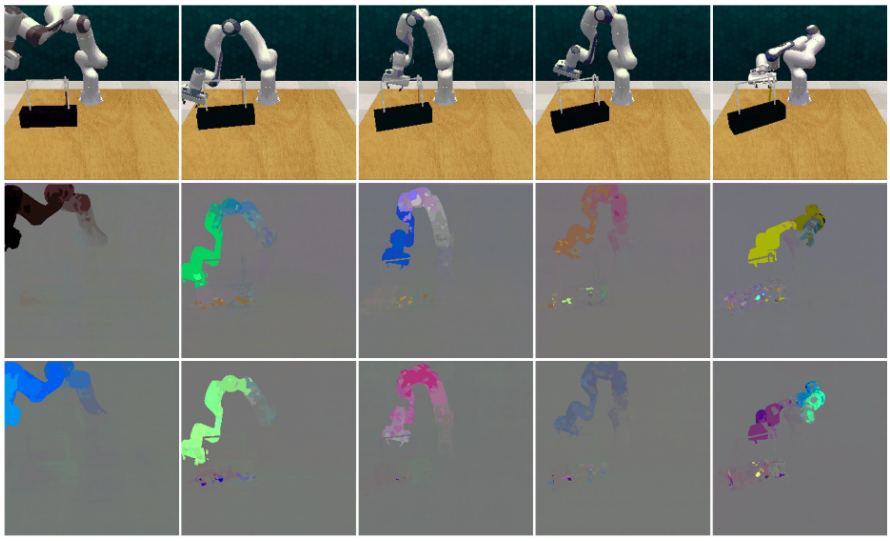


Fig. 9: Visualizations of the learned six basic velocity components.

Note that, if we entirely remove our 3D world model and the incorporation module, the resulting framework is exactly the 3DFA backbone, which already performs worse than our method, as shown in Tables 1&3. Therefore, there is no need to conduct a separate ablation experiment.

(2) Removing the design of learnable token $\boldsymbol{L}$: In Step #5 of incorporating 3D scene future dynamics (Section 3.3), the learnable token $\boldsymbol{L}$ provides flexibility to fuse valuable dynamics information within a local neighborhood. To validate this design, we remove $\boldsymbol{L}$ and instead simply adopt a self-attention mechanism to fuse predicted future dynamics within the neighborhood.

(3) Optimizing Gaussian velocity field with different iterations T: In our online Algorithm 1, we set $T = 50$ to sufficiently optimize the Gaussian dynamics, enabling more accurate future prediction. Intuitively, a smaller T reduces optimization time but may degrade dynamics learning. To evaluate this sensitivity, we conduct 4 additional ablation experiments with $T \in \{40, 30, 20, 10\}$.

Analysis: Table 4 shows that: (1) Removing the 3D scene basic velocity components $\boldsymbol{D}_t$ leads to the most significant drop in success rate, making performance comparable to the backbone 3DFA. This demonstrates that integrating future dynamics is crucial for tackling the challenging dynamic tasks. (2) Replacing the learnable token based mechanism with pure self-attention also noticeably degrades performance, highlighting the effectiveness of the simple and flexible learnable token $\boldsymbol{L}$ for fusing future dynamics within a local neighborhood. (3) Figure 8 shows that our method is particularly robust to different numbers of iterations $\boldsymbol{T}$ for optimizing the velocity field, enabling our framework to quickly learn dynamics and execute actions, which is essential for dynamic manipulation.

5.4 Analysis of Gaussian Velocity

We further analyze how the learned velocity components concretely contribute to effective manipulation. During online evaluation on PhysMani-Bench, at a

Table 5: Training/inference time and memory of different methods.

	Training Time (hours)	Training Memory (GB/GPU)	Inference Time (ms/frame)	Inference Memory (GB)
Act3D [19]	19	18.5	18.8	1.1
3DDA [30]	32	22.0	267.2	1.0
3DFA [20]	42	22.0	264.5	1.0
3DFA-OF	47	22.0	260.0	1.0
ManiGaussian [47]	60	23.5	897.7	5.9
pi0.5 [9]	149	15.2	132.7	7.4
PhysMani(Ours)	53	22.0	272.8	1.0

given time step t, we visualize the predicted six basic velocity components for future frames $\{10^{th}, 30^{th}, 50^{th}, 70^{th}, 90^{th}\}$, along with their corresponding rendered RGB images in Figure 9. The second row shows the three linear velocity components, while the third row shows the three angular velocity components. The visualizations indicate that the velocity components naturally focus on the moving robot arm and the rotating target object, while largely ignoring the static background. This provides an explicit and physics-grounded signal that directly guides the robot's actions.

5.5 Computation Efficiency

For inference, all models are evaluated on an Intel Core i7-13700F CPU paired with an NVIDIA RTX 4090 GPU. Table 5 reports the total time required to train each model. PhysMani requires 272.8 ms to infer a single keyframe action. These keyframe actions are then converted into joint trajectories via inverse kinematics. In real-world experiments, to decouple inference from execution, we utilize an action buffer with a temporal ensemble operating at 5 Hz. Since the online optimization of the world model takes ∼200 ms and runs in parallel, the policy model inference remains the overall system bottleneck. Together, these design choices enable fast and dynamic manipulation.

6 Conclusion

This work presents PhysMani, a general framework for dynamic object manipulation that explicitly models 3D scene future dynamics and conditions action prediction on them. By learning a divergence-free Gaussian velocity field online, our world model forecasts physically meaningful future dynamics with low latency, while the future-aware action policy model effectively exploits these predictions via local cross-attention. Across our newly introduced PhysMani-Bench and challenging real-world tasks, our method consistently outperforms state-of-the-art 3D action policy and VLA-based baselines, particularly in highly dynamic tasks. These results highlight the importance of physics-principled 3D world models for embodied control and provide a practical path toward general dynamic manipulation in open environments.

Acknowledgments

This work was supported in part by Research Grants Council of Hong Kong under Grants 15228626 & 15225522 & 15219125, and in part by Otto Poon Charitable Foundation Smart Cities Research Institute (8-CDCQ), in part by Research Center for Unmanned Autonomous Systems (1-CE3D), The Hong Kong Polytechnic University.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical AI. arXiv preprint arXiv:2501.03575 (2025)
3. Akinola, I., Xu, J., Song, S., Allen, P.K.: Dynamic grasping with reachability and motion awareness. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 9422–9429 (2021)
4. Ali, A., Bai, J., Bala, M., Balaji, Y., Blakeman, A., Cai, T., Cao, J., Cao, T., Cha, E., Chao, Y.W., et al.: World simulation with video foundation models for physical AI. arXiv preprint arXiv:2511.00062 (2025)
5. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025)
6. Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., Zada, S., Ephrat, A., Hur, J., Liu, G., Raj, A., et al.: Lumiere: A space-time diffusion model for video generation. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
7. Barcellona, L., Zadaianchuk, A., Allegro, D., Papa, S., Ghidoni, S., Gavves, E.: Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In: International Conference on Learning Representations. vol. 2025, pp. 56729–56763 (2025)
8. Bi, H., Tan, H., Xie, S., Wang, Z., Huang, S., Liu, H., Zhao, R., Feng, Y., Xiang, C., Rong, Y., et al.: Motus: A unified latent action world model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 35101–35113 (2026)
9. Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M.R., Finn, C., Fusai, N., Galliker, M.Y., et al.: pi0.5: a vision-language-action model with open-world generalization. In: Conference on Robot Learning (2025)
10. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M.R., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Smith, L., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: pi0: A Vision-Language-Action Flow Model for General Robot Control. In: Proceedings of Robotics: Science and Systems. LosAngeles, CA, USA (June 2025)
11. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)

12. Bruce, J., Dennis, M.D., Edwards, A., Parker-Holder, J., Shi, Y., Hughes, E., Lai, M., Mavalankar, A., Steigerwald, R., Apps, C., et al.: Genie: Generative interactive environments. In: International Conference on Machine Learning (2024)
13. Chen, Y., Liang, Y., Wang, J., Chen, T., Cheng, J., Gu, Z., Huang, Y., Jiang, Z., Li, W., Li, T., et al.: TeleWorld: Towards dynamic multimodal synthesis with a 4d world model. arXiv preprint arXiv:2601.00051 (2026)
14. Christen, S., Yang, W., Pérez-D'Arpino, C., Hilliges, O., Fox, D., Chao, Y.W.: Learning human-to-robot handovers from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9654–9664 (2023)
15. Coumans, E., Bai, Y.: PyBullet, a Python module for physics simulation for games, robotics and machine learning. `http://pybullet.org` (2016–2021), last accessed 28 Jun 2026
16. D'Ambrosio, D.B., Abeyruwan, S., Graesser, L., Iscen, A., Amor, H.B., Bewley, A., Reed, B.J., Reymann, K., Takayama, L., Tassa, Y., et al.: Achieving human level competitive robot table tennis. In: IEEE International Conference on Robotics and Automation. pp. 74–82 (2025)
17. Ding, J., Zhang, Y., Shang, Y., Zhang, Y., Zong, Z., Feng, J., Yuan, Y., Su, H., Li, N., Sukiennik, N., et al.: Understanding world or predicting future? a comprehensive survey of world models. ACM Computing Surveys **58**(3), 1–38 (2025)
18. Gao, G., Wang, J., Zuo, J., Jiang, J., Zhang, J., Zeng, X., Zhu, Y., Ma, L., Chen, K., Sheng, M., et al.: Towards human-level intelligence via human-like whole-body manipulation. arXiv preprint arXiv:2507.17141 (2025)
19. Gervet, T., Xian, Z., Gkanatsios, N., Fragkiadaki, K.: Act3D: 3D feature field transformers for multi-task robotic manipulation. In: Conference on Robot Learning (2023)
20. Gkanatsios, N., Xu, J., Bronars, M., Mousavian, A., Ke, T.W., Fragkiadaki, K.: 3D FlowMatch Actor: Unified 3D policy for single-and dual-arm manipulation. arXiv preprint arXiv:2508.11002 (2025)
21. Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y.W., Fox, D.: RVT: Robotic view transformer for 3D object manipulation. In: Conference on Robot Learning. pp. 694–710 (2023)
22. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
23. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 **2**(3), 440 (2018)
24. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse control tasks through world models. Nature **640**(8059), 647–653 (2025)
25. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
26. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
27. James, S., Ma, Z., Arrojo, D.R., Davison, A.J.: RLBench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters **5**(2), 3019–3026 (2020)
28. James, S., Wada, K., Laidlow, T., Davison, A.J.: Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13739–13748 (2022)
29. Kawaharazuka, K., Oh, J., Yamada, J., Posner, I., Zhu, Y.: Vision-language-action models for robotics: A review towards real-world applications. IEEE Access (2025)
30. Ke, T.W., Gkanatsios, N., Fragkiadaki, K.: 3D Diffuser Actor: Policy diffusion with 3D scene representations. In: Conference on Robot Learning (2024)
31. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3D Gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (Jul 2023)
32. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., et al.: OpenVLA: An open-source vision-language-action model. In: Conference on Robot Learning (2024)
33. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: Robocup: The robot world cup initiative. In: International Conference on Autonomous Agents. pp. 340–347 (1997)
34. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: HunyuanVideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
35. Li, J., Song, Z., Yang, B.: NVFi: Neural velocity fields for 3d physics learning from dynamic videos. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
36. Li, J., Song, Z., Yang, B.: TRACE: Learning 3D Gaussian physical dynamics from multi-view videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8820–8829 (2025)
37. Li, J., Song, Z., Zhou, S., Yang, B.: FreeGave: 3D physics learning from dynamic videos by Gaussian velocity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12433–12443 (2025)
38. Li, W., Zhao, H., Yu, Z., Du, Y., Zou, Q., Hu, R., Xu, K.: PIN-WM: Learning physics-informed world models for non-prehensile manipulation. arXiv preprint arXiv:2504.16693 (2025)
39. Li, X., He, X., Zhang, L., Wu, M., Li, X., Liu, Y.: A comprehensive survey on world models for embodied AI. arXiv preprint arXiv:2510.16732 (2025)
40. Li, Y., Wu, J., Tedrake, R., Tenenbaum, J.B., Torralba, A.: Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In: International Conference on Learning Representations (2019)
41. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: DeepSeek-V3 technical report. arXiv preprint arXiv:2412.19437 (2024)
42. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: LIBERO: benchmarking knowledge transfer for lifelong robot learning. In: Advances in Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
43. Liu, H., Lee, L., Lee, K., Abbeel, P.: Instruction-following agents with multimodal transformer. arXiv preprint arXiv:2210.13431 (2022)
44. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 34892–34916. Curran Associates, Inc. (2023)
45. Liu, J., Zhang, R., Fang, H.S., Gou, M., Fang, H., Wang, C., Xu, S., Yan, H., Lu, C.: Target-referenced reactive grasping for dynamic objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8824–8833 (2023)

46. Liu, X., Gong, C., et al.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations (2023)
47. Lu, G., Zhang, S., Wang, Z., Liu, C., Lu, J., Tang, Y.: ManiGaussian: Dynamic Gaussian splatting for multi-task robotic manipulation. In: European Conference on Computer Vision. pp. 349–366 (2024)
48. Ma, Y., Song, Z., Zhuang, Y., Hao, J., King, I.: A survey on vision–language–action models for embodied AI. IEEE Transactions on Neural Networks and Learning Systems (2026)
49. Marturi, N., Kopicki, M., Rastegarpanah, A., Rajasekaran, V., Adjigble, M., Stolkin, R., Leonardis, A., Bekiroglu, Y.: Dynamic grasp and trajectory planning for moving objects. Autonomous Robots **43**(5), 1241–1256 (2019)
50. Mees, O., Hermann, L., Rosete-Beas, E., Burgard, W.: CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. IEEE Robotics and Automation Letters **7**(3), 7327–7334 (2022)
51. Motamed, S., Culp, L., Swersky, K., Jaini, P., Geirhos, R.: Do generative video models understand physical principles? In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 948–958 (2026)
52. Noh, D., Kong, D., Zhao, M., Lizarraga, A., Xie, J., Wu, Y.N., Hong, D.: Latent adaptive planner for dynamic manipulation. In: Conference on Robot Learning. pp. 2430–2448 (2025)
53. Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Xu, C., Luo, J., Kreiman, T., Tan, Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., Levine, S.: Octo: An open-source generalist robot policy. In: Proceedings of Robotics: Science and Systems. Delft, Netherlands (2024)
54. Qwen Team: Qwen2.5-VL Technical Report. arXiv preprint arXiv:2502.13923 (2025)
55. Sapkota, R., Cao, Y., Roumeliotis, K.I., Karkee, M.: Vision-language-action models: Concepts, progress, applications and challenges. arXiv preprint arXiv:2505.04769 (2025)
56. Shridhar, M., Manuelli, L., Fox, D.: Perceiver-Actor: A multi-task transformer for robotic manipulation. In: Conference on Robot Learning. pp. 785–799 (2023)
57. Tang, J., Liu, J., Li, J., Wu, L., Yang, H., Zhao, P., Gong, S., Yuan, X., Shao, S., Zhang, L., et al.: Hunyuan-gamecraft-2: Instruction-following interactive game world model. arXiv preprint arXiv:2511.23429 (2025)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
59. Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., Kim, M.J., Du, M., et al.: BridgeData v2: A dataset for robot learning at scale. In: Conference on Robot Learning. pp. 1723–1736 (2023)
60. Wan Team, Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
61. Xie, H., Wen, B., Zheng, J., Chen, Z., Hong, F., Diao, H., Liu, Z.: DynamicVLA: A vision-language-action model for dynamic object manipulation. arXiv preprint arXiv:2601.22153 (2026)
62. Yang, A., Yu, B., Li, C., Liu, D., Huang, F., Huang, H., Jiang, J., Tu, J., Zhang, J., Zhou, J., Lin, J., Dang, K., Yang, K., Yu, L., Li, M., Sun, M., Zhu, Q., Men,

R., He, T., Xu, W., Yin, W., Yu, W., Qiu, X., Ren, X., Yang, X., Li, Y., Xu, Z., Zhang, Z.: Qwen2.5-1m technical report. arXiv preprint arXiv:2501.15383 (2025)
63. Ye, S., Ge, Y., Zheng, K., Gao, S., Yu, S., Kurian, G., Indupuru, S., Tan, Y.L., Zhu, C., Xiang, J., et al.: World action models are zero-shot policies. In: International Conference on Learning Representations the 2nd Workshop on World Models: Understanding, Modelling and Scaling (2026)
64. Zhang, D., Sun, J., Hu, C., Wu, X., Yuan, Z., Zhou, R., Shen, F., Zhou, Q.: Pure vision language action (VLA) models: A comprehensive survey. arXiv preprint arXiv:2509.19012 (2025)
65. Zhang, Y., Wang, R., Chen, X.: Dynamic behavior cloning with temporal feature prediction: Enhancing robotic arm manipulation in moving object tasks. IEEE Robotics and Automation Letters **10**(6), 5209–5216 (2025)
66. Zhou, S., Wang, H., Cheng, H., Li, J., Wang, D., Jiang, J., Jin, Y., Huang, J., Mao, S., Liu, S., Yang, Y., Song, H., Wei, S., Zhang, Z., Wang, B., Wang, Z., Zou, C., Yang, B.: PhysInOne: Visual Physics Learning and Reasoning in One Suite. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 33131–33142 (2026)
67. Zhou, Y., Sun, G., Miao, Y., Zhang, Y., Chen, X., Wang, H.: Spatiotemporal optimal trajectory planning for safe planar manipulation of a moving object. IEEE Transactions on Industrial Electronics **71**(7), 7466–7476 (2023)
68. Zhu, Z., Wang, X., Zhao, W., Min, C., Li, B., Deng, N., Dou, M., Wang, Y., Shi, B., Wang, K., et al.: Is Sora a world simulator? a comprehensive survey on general world models and beyond. arXiv preprint arXiv:2405.03520 (2024)
69. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183 (2023)