Learning 4D Embodied World Models

Anonymous CVPR submission

Paper ID 11899

Abstract

In this paper, we present a 4D embodied world model, which takes in an image observation and language instruction as input and predicts a 4D dynamic scene showing how the scene will change as the embodied agent performs actions. In contrast to previously learned world models which typically generate 2D videos, our 4D model provides detailed 3D information on precise configurations and shapes of objects in a scene over time. This allows us to effectively learn accurate inverse dynamic models for an embodied agent to execute a policy for interacting with the environment. To construct a dataset to train such 4D world models, we first annotate large-scale existing video robotics datasets using pre-trained depth and normal prediction models to construct 4D-consistent scenes of each video. To efficiently learn generative models on this 4D data, we propose to train a video generative model on this annotated dataset, which jointly predicts RGB-DN (RGB, Depth, and Normal) for each video. We then present an algorithm to directly convert generated RGB, Depth, and Normal images into a high-quality dynamic 4D mesh of the world. We illustrate how this enables us to predict high-quality meshes consistent across both time and space from embodied scenarios, render novel views for embodied scenes, and construct policies that substantially outperform those from prior 2D and 3D models of the world. Our code, model, and dataset will be made publicly available.

1. Introduction

Learned world models [20, 61, 64, 70], which simulate environmental dynamics, play a crucial role in enabling embodied intelligent agents. Such models enable flexible policy synthesis [14, 41], data simulation and generation [64, 73], and long-horizon planning [13, 28, 68]. However, while the physical world is three-dimensional in nature, existing world models operate in the space of 2D pixels. This limitation leads to an incomplete representation of spatial relationships, impeding tasks that require precise depth and pose information. For instance, without accurate depth and 6-DoF pose estimations, robotic systems struggle to determine the exact position and orientation of objects. Furthermore, existing 2D models can produce unrealistic results, such as inconsistent object sizes and shapes across time steps, which limits their use in data-driven simulations and robust policy learning.

In this paper, we explore how we can instead learn a 4D embodied world model, which directly simulates the dynamics of a 3D world. This approach allows us to generate realistic 3D interactions, such as grasping objects or opening drawers, with a level of detail that traditional 2D-based models cannot achieve. By modeling spatial and temporal dimensions, our model provides the depth and pose information essential for robotic manipulation.

However, the task of learning a 4D embodied world model is challenging as the dynamics of the world are extremely computationally expensive to train and learn, requiring models to generate outputs in three-dimensional space and time. To efficiently represent and predict the dynamics of the world, we propose a substantially more lightweight representation of the 4D world, consisting of predicting a sequence of RGB, depth, and normal maps of the scene. This combined representation accurately captures the appearance, geometry, and surface of a scene while being substantially lower dimensional than explicitly predicting world dynamics. Furthermore, such a representation shares substantial similarities to existing video space models, allowing us to directly use the generative capabilities and architecture improvements of existing video space models to effectively construct our 4D world model.

Given this intermediate representation, we present an efficient algorithm to reconstruct accurate 4D scenes from generated maps. For each frame, we use a combination of both depth and normal prediction to integrate a smooth 3D surface of the scene. We then use optical flow between generated frames to distinguish between background and dynamic regions in the reconstructed 3D scene across frames and add a loss function to reconstructions to enforce consistency across scenes over time. We find this enables us to construct fidelity 4D generated meshes for the scene suitable for downstream tasks such as policy prediction for the robot (Figure 1).

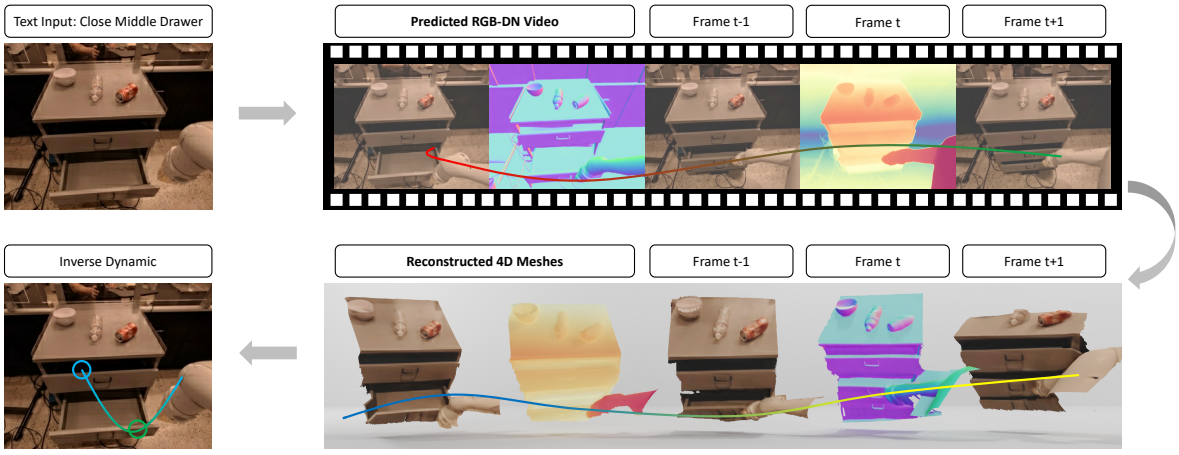

Figure 1. **4D Embodied World Models.** Our approach gets an input image and instructions and predicts RGB-D-N (RGB, Depth, Normal) maps. We propose a normal integration method that constructs a high-quality 4D mesh of the interaction from these predictions.

A key challenge for training a 4D world model is a lack of access to existing large-scale datasets with existing 4D annotations, or the high-quality image, depth, and normal annotations needed to train our approach. To construct such data, in principle, we can use pre-trained depth and normal estimators [30, 63] to obtain such estimates from video data, but these estimators are typically limited to predicting relative value maps from individual frames. As a result, when a scene changes, these estimators will produce depth and normal maps with inconsistencies across time. To tackle this challenge, we develop a data collection pipeline that leverages optical flow between frames in a video to enforce consistency between generated depth and normal maps across timesteps. In particular, we use optical flow to guide a depth and normal diffusion model across frames in a video, which we find is sufficient to ensure consistent depth and normal predictions across all timesteps without the need for expensive ground-truth annotations, facilitating the training of our world model on a large scale.

Overall, our paper has the following contributions: **(1)** We introduce a 4D embodied world model, and present an efficient representation of this model, in the form of RGB, depth, and normal maps, and illustrate how this representation can be used to construct a full 4D scene. **(2)** We present a pipeline to automatically extract 4D world model data from existing robot video datasets, leveraging an optical flow-guided depth and normal diffusion model. **(3)** We illustrate the approach's efficacy in generating consistent 4D meshes across different environments, substantially outperforming other baselines, and showing its downstream use as an effective policy.

2. Related Work

Embodied Foundation Models A flurry of recent work has focused on constructing foundation models for general purpose agents [17, 65]. One line of work has focused on constructing multimodal language models that operate over images [12, 19, 29, 40, 50, 60, 67] as well as 3D inputs [25, 26] and output text describing the actions of an agent. Other works have focused on the construction of vision-language-action (VLA) models that directly output action tokens [6, 32, 69]. Both of the previous approaches aim to construct foundation model policies (over text or continuous actions). In contrast, our work aims to instead construct a foundation 4D world model for embodied agents, which can then be used for downstream applications such as planning [13, 68] or policy synthesis [14, 41].

Learning World Models Learning dynamics model of the world given control inputs has been a long-standing challenge in system identification [42], model-based reinforcement learning [54], and optimal control [2, 74]. A large body of work focused on learning world models in the low dimensional state space [1, 16, 38], which while being efficient to learn, is difficult to generalize across many environments. Other works have explored how world models may be learned over pixel-space images [10, 11, 20, 21, 44], but such models are trained on simple game environments. With advances in generative modeling, a large flurry of recent research has focused on using video models as foundation world models [7, 47, 61, 64, 70, 72] but such models operate over the space of 2D pixels which does not fully simulate the 3D world. Most similar to our work, in Zhen et al. [69], a world model over 3D inputs is learned. In contrast to this work, our world model directly captures the dynamics of 3D scenes using a compact representation of RGB-DN video.

3. Learning 4D Embodied World Models

We introduce the 4D Embodied World Model, which predicts future RGB, depth, and normal maps based on a given input image and text. We leverage a pre-trained video diffusion

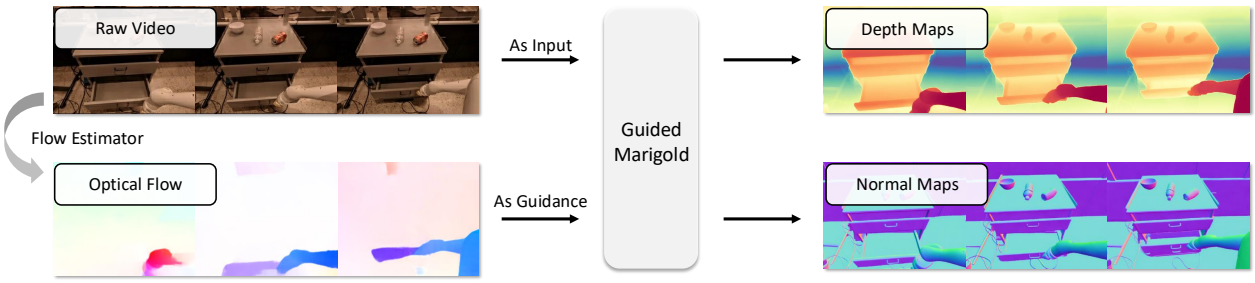

Figure 2. **Data Annotation Visualization.** An illustration for the data annotation pipeline.

model as the backbone to predict this rich set of 2D geometric information. Then, we propose an efficient method to convert the RGB-DN video into 4D scenes.

3.1. 4D Embodied Video Data Annotation

Learning 4D embodied world models requires large-scale 4D datasets, which are expensive to collect in the real world. In this section, we present a data annotation pipeline that enables us to automatically construct 4D datasets from existing video datasets. For an illustration of this process, see Fig. 2.

Given an input video $\mathcal{V}$, our data annotation protocol aims to automatically obtain 4D annotations of depth map sequence $\mathcal{D}$ and normal map sequence $\mathcal{N}$. Video depth/normal requires consistent predictions between consecutive frames [4, 43]. Therefore, we utilize flow as guidance [39]. By predicting the optical flow, we can guarantee that static areas of a scene remain temporally consistent between consecutive frames. Note that the optical flow method is effective in this case, as existing manipulation datasets [5, 6, 18, 27, 35, 36, 46, 57] are based on fixed camera poses.

To be specific, we employ RAFT [55], an efficient optical flow estimation model, to generate the optical flow $\mathcal{F}$. The optical flow between consecutive frames is computed as $\mathcal{F} = \mathrm{RAFT}(\mathcal{V})$. The next step involves generating the 3D annotations: depth $\mathcal{D}$ and surface normal $\mathcal{N}$. These 3D annotations provide richer visual context and enhance the understanding of the 4D environment represented by $\mathcal{V}$. Recent advances in monocular depth and normal estimators, typically based on models like Vision Transformers (ViT) or Diffusion Models, have shown strong generalization capabilities across diverse datasets.

Then, we leverage the diffusion model Marigold [30] to estimate $\mathcal{D}^i$ and $\mathcal{N}^i$ from each video frame $\mathcal{V}^i$. The frame is first encoded into a latent representation $\mathbf{v}^i$ using a variational autoencoder (VAE) [33, 51]: $\mathbf{v}^i = \mathrm{Encoder}(\mathcal{V}^i)$. At the same time, an initial depth or normal latent map $\mathbf{z}_T^i$ is sampled from a Gaussian distribution: $\mathbf{z}_T^i \sim \mathcal{N}(0, 1)$. Then we iteratively apply the learned denoiser U-Net [53] ϵ on each timestep t to reconstruct the $\mathbf{z}_0^i$ and $\mathcal{Z}^i$:

$$\mathbf{z}_{t-1}^i = \epsilon(\mathbf{v}^i, \mathbf{z}_t^i, t) \quad \text{and} \quad \mathcal{Z}^i = \mathrm{Decoder}(\mathbf{z}_0^i) \tag{1}$$

where $(\mathbf{z}, \mathcal{Z}) \in \{(\mathbf{d}, \mathcal{D}), (\mathbf{n}, \mathcal{N})\}$.

However, most existing approaches [3, 30, 63] are primarily designed for image-based inputs and often struggle when processing videos, resulting in unstable predictions. To overcome this limitation, we propose a novel technique that leverages optical flow as guidance to refine depth estimation. The key insight is that optical flow can capture how static backgrounds and objects move within a scene. Therefore, during the inference stage of the diffusion models, we introduce a loss function that enforces consistency between the backgrounds of consecutive frames. Specifically, we define a static region mask for the i-th frame based on the optical flow magnitude: $\mathcal{M}^i = (\|\mathcal{F}^i\| < c)$ where c is a predefined threshold. In practice, we erode the mask to ensure it covers a robust region. Finally, we define the loss function and integrate its gradient into Eq. 1 for every denoising step:

$$\mathcal{L}(\mathbf{z_t^i}) = \Big|\Big|\mathrm{Decoder}\big(\epsilon(\mathbf{v}^i, \mathbf{z}_t^i, t)\big) \circ (\mathcal{M}^i \cap \mathcal{M}^{i-1}) - \mathcal{Z}^{i-1} \circ (\mathcal{M}^i \cap \mathcal{M}^{i-1})\Big|\Big|^2 \tag{2}$$

$$\mathbf{z}_{t-1}^i = \epsilon_\theta\Big[\mathbf{v}^i, \Big(\mathbf{z}_t^i - w\nabla_{\mathbf{z_t^i}}\mathcal{L}\Big), t\Big] \tag{3}$$

where w is a guidance weight, $\circ$ represents the element-wise product and $\mathcal{M}_i \cap \mathcal{M}_{i-1}$ selects the overlapping stable regions between two consecutive frames. The pseudocode is presented in the appendix.

3.2. Preliminaries on Video Diffusion Models

Diffusion models [24, 52] are capable of learning the data distribution $p(x)$ by progressively adding noise to the data until it resembles a Gaussian distribution through a forward process. During inference, a denoiser ϵ is trained to recover the data from this noisy state. Latent video diffusion models [70] utilize a Variational Autoencoder (VAE) [33, 56], in the latent space of the data, maintaining high-quality outputs while more efficiently modeling the data distribution. In this section, we formulate the task of RGB $\mathcal{V}$, depth $\mathcal{D}$, and normal $\mathcal{N}$ map video generation as a conditional denoising generation task, i.e., we model the distribution $p(\mathbf{v}, \mathbf{d}, \mathbf{n}|\mathbf{v}_0, \mathcal{T})$, where $\mathbf{v}, \mathbf{d}, \mathbf{n}$ represent the predicted future latent sequences of RGB, depth, and normal maps, respectively. The condition $\mathbf{v}_0$ is a given RGB image latent, and $\mathcal{T}$ denotes the instruction provided by the user.

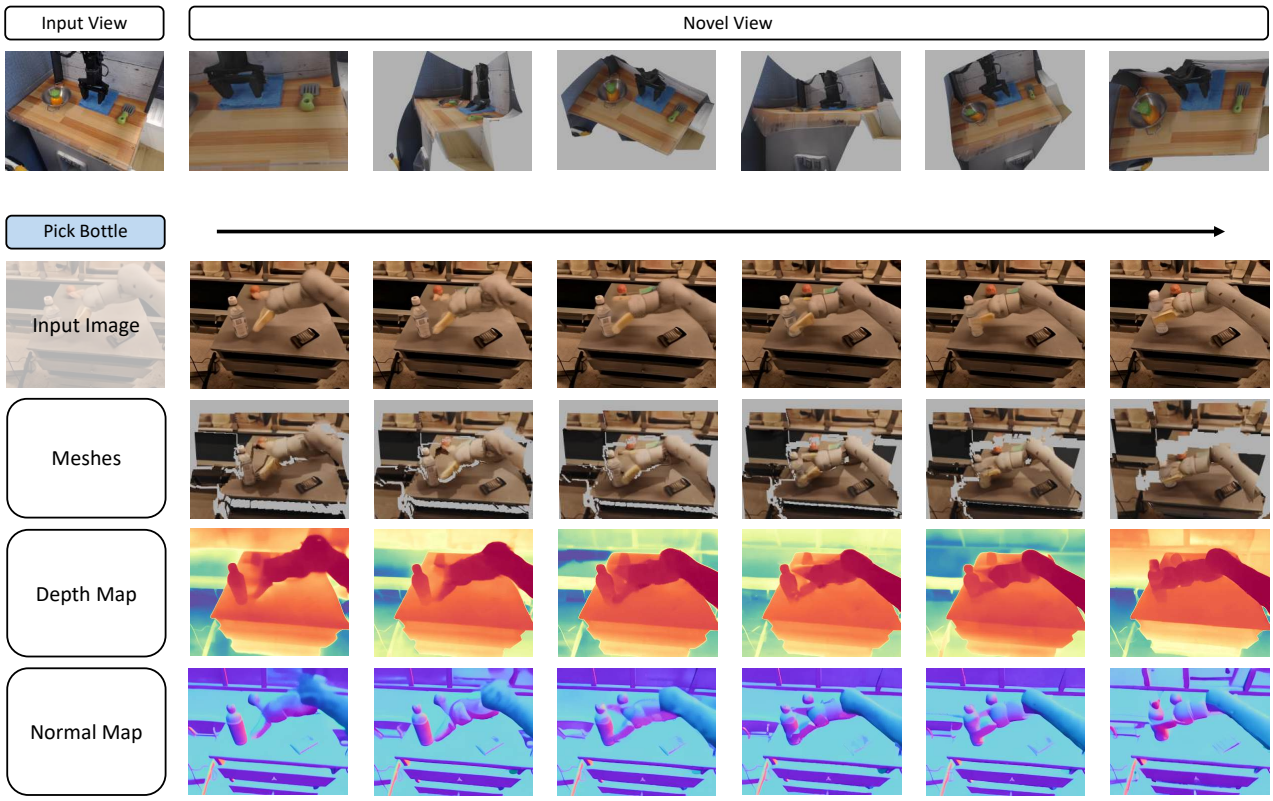

Figure 3. **4D Reconstruction and Video Generation.** The first row shows the results of our 4D reconstruction method from multiple views on a single frame. The following rows present the inference results of our method given the input image and instruction, along with the reconstructed meshes.

The forward diffusion process adds Gaussian noise to the latent $\mathbf{z} \in \{\mathbf{v}, \mathbf{d}, \mathbf{n}\}$ over T timesteps, defined as:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\alpha_t}\mathbf{z}_{t-1}, (1-\alpha_t)\mathbf{I}\right) \quad (4)$$

where $t \in \{1, 2, \ldots, T\}$ denotes the diffusion step, α_t is a parameter controlling the noise influence at each step, and $\mathbf{I}$ is the identity matrix. In the reverse process, the model aims to recover the original latent from the noise. A denoising network $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{v}_0, \mathcal{T})$ with learning parameters θ is trained to predict the noise added at each timestep. For simplicity, let $\mathbf{x}_t = [\mathbf{v}_t, \mathbf{n}_t, \mathbf{d}_t]$, which denotes the concatenation operation. The reverse process is defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{v}_0, \mathcal{T}) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t, \mathbf{v}_0, \mathcal{T}), \Sigma_\theta(\mathbf{x}_t, t)\right) \quad (5)$$

where

$$\mu_\theta(\mathbf{x}_t, t, \mathbf{v}_0, \mathcal{T}) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{v}_0, \mathcal{T})\right). \quad (6)$$

The variance term $\Sigma_\theta(\mathbf{x}_t, t)$ is typically constant. Once the denoised sequence of latent $\mathbf{z}_0$ is obtained, the model reconstructs the final video frames using the decoder network, mapping the latent back to the pixel space: $\mathcal{Z} = \text{Decoder}(\mathbf{z}_0)$.

During training, we randomly select samples from the dataset $(\mathcal{V}, \mathcal{D}, \mathcal{N}, \mathcal{T})$ and apply Eq.4 to add noise $\epsilon_\mathbf{v}$, $\epsilon_\mathbf{d}$, and $\epsilon_\mathbf{n}$ to the RGB-DN data at timestep t, minimizing the following objective:

$$L = \mathbb{E}_{\mathbf{v}_0, \mathcal{T}, t, \epsilon}\left[\left\|[\epsilon_\mathbf{v}, \epsilon_\mathbf{d}, \epsilon_\mathbf{n}] - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{v}_0, \mathcal{T})\right\|^2\right] \quad (7)$$

3.3. RGB, Depth and Normal Video Predictions

Training a diffusion model to model temporal RGB-DN data is a challenging task. To effectively train RGB video models, large-scale video datasets with billions of high-quality samples are used [70]. In contrast, even through automatic annotation, our dataset of RGB-DN data contains only around one million data points, which is insufficient to train a world model from scratch. To address this, we finetune the Open-Sora [70] as our RGB-DN prediction model and directly leverage the pre-trained knowledge inside the model to effectively bootstrap our 4D model.

To implement this, we use the temporal VAE [33, 56] in Open-Sora [70] to separately encode RGB, depth, and normal images for each frame of a video, without additional fine-tuning of the VAE. We then expand Open-Sora STDiT's

input and output channels threefold. Specifically, we adjust the first 3D convolutional input layer and the output linear layer to process concatenated RGB-DN video channels. We fine-tune all STDiT parameters to denoise videos, allowing the model to leverage pre-trained knowledge from video data to predict RGB, depth, and normal frames, capturing the 4D dynamics of a scene.

3.4. 4D Scene Reconstruction from RGB-DN Video

After obtaining the RGB-DN video, we further optimize the depth and reconstruct the surface to generate a full final dynamic mesh of the scene. Similar to prior works [30, 66], our depth representation for each image is given by a relative map in the range $[0, 1]$, and thus cannot directly reconstruct the entire scene. While past work has sidestepped this by assuming either a default scale for depth or by directly predicting metric depth, such reconstructions from depth are often coarse and often cause reconstructed planes or walls to be tilted.

We instead leverage the normal maps $\mathcal{N}^i$ and optical flow $\mathcal{F}^i$ between frames in a video to obtain precise per-pixel depth estimates from relative depth maps $\mathcal{D}^i$. Normal maps provide essential information about surface orientation, which is vital for enforcing geometric constraints and imposing surface smoothness and continuity during depth optimization. This spatial optimization leads to more accurate and reliable depth estimates that closely align with the true 3D geometry and capture fine surface details. Additionally, we use optical flow between frames to enforce 3D consistency over time, effectively optimizing in the temporal domain. Together, normals and optical flow offer spatial and temporal constraints that enhance depth prediction. Both maps, in combination with the coarse depth $\mathcal{D}^i$, allow us to optimize a refined depth map $\hat{\mathcal{D}}$ that corresponds to a consistent 4D scene.

To formalize the process and enforce consistency across frames, we can use the perspective camera model to set constraints on the depth and surface normal. In the coordinate system of the 2D image at frame i, a pixel position is given as $\boldsymbol{u} = (u, v)^T \in \mathcal{V}^i$, and its corresponding depth scalar, normal vector is $d \in \mathcal{D}^i, \boldsymbol{n} = (n_x, n_y, n_z) \in \mathcal{N}^i$. Under the assumption of a perspective camera whose focal length is f and the principal point is $(c_u, c_v)^T$, as proposed by [15], the log-depth $\tilde{d} = \log(d)$ should satisfy the following equations: $\tilde{n}_z \partial_u \tilde{d} + n_x = 0$ and $\tilde{n}_z \partial_v \tilde{d} + n_y = 0$ where $\tilde{n}_z = n_x(u - c_u) + n_y(v - c_v) + n_z f$. In addition, we can add, the assumption that assumes all locations are smooth surfaces [9], we can convert the above constraint to the quadratic loss function, allowing us to find the minimized depth map:

$$\min_{d} \iint_{\Omega} (\tilde{n}_z \partial_u \tilde{d} + n_x)^2 + (\tilde{n}_z \partial_u \tilde{d} + n_y)^2 \mathrm{d}u\mathrm{d}v. \quad (8)$$

Following [9], we can convert the above objective to an iteratively optimized loss objective. At iteration step t, we can compute the matrix $W(\tilde{d}_t)$ and iteratively optimize for a refined depth prediction $\tilde{d}_{t+1}$:

$$\tilde{d}_{t+1} = \arg\min_{\tilde{d}} (A\tilde{d} - b)^T W(\tilde{d}_t)(A\tilde{d} - b) \stackrel{\text{def}}{=} \arg\min_{\tilde{\mathcal{D}}} \mathcal{L}(\tilde{\mathcal{D}}, \mathcal{N}^i), \quad (9)$$

where A and b are defined by predicted normals and camera intrinsics.

The above approach optimizes depth frame by frame, which lacks temporal consistency across the dynamic scene. To address this, we compute optical flow between frames [55] $\mathcal{F} = \text{RAFT}(\mathcal{V})$ and enforce consistency of depth across frames. We define the static regions of each frame as the pixels with the magnitude of optical flow smaller than threshold $\mathcal{M}^i = \|\mathcal{F}^i\| \leq c$. We then define the dynamic parts of an image M_d^i as all regions not in $\mathcal{M}^i$. We further define the background of an image $\mathcal{M}_b^i$ as static regions that are fixed across image frames, $\mathcal{M}_b^i = \mathcal{M}^i \cap \mathcal{M}^{i-1}$

Since optical flow represents the movement of objects in the 2D-pixel space, we can retrieve the depth at any position from the previous frame to impose consistency constraints. To compute the depth values from the previous frame at positions corresponding to the current frame, we utilize the optical flow $\mathcal{F}^{i \to (i-1)}$. For each pixel (u, v) in frame i, the optical flow provides the displacement $(\Delta u, \Delta v)$, allowing us to find the corresponding pixel in frame $i - 1$ at position $(u - \Delta u, v - \Delta v)$. Based on this mapping, we define the $\mathcal{D}^{i \to (i-1)}$ such that: $\mathcal{D}^{i \to (i-1)}(u, v) = \mathcal{D}^{i-1}(u - \Delta u, v - \Delta v)$. We then introduce the loss function $\mathcal{L}_d$ for dynamic regions of an image:

$$\mathcal{L}_d(\tilde{\mathcal{D}}, \hat{\mathcal{D}}^{i-1}, \mathcal{M}_d^i, \mathcal{F}^i, \mathcal{F}^{i-1}) = \left\| \tilde{\mathcal{D}}^i \circ \mathcal{M}_d^i - \mathcal{D}^{i \to (i-1)} \circ \mathcal{M}_d^i \right\|^2. \quad (10)$$

In addition to the loss terms $\mathcal{L}$ defined previously, we

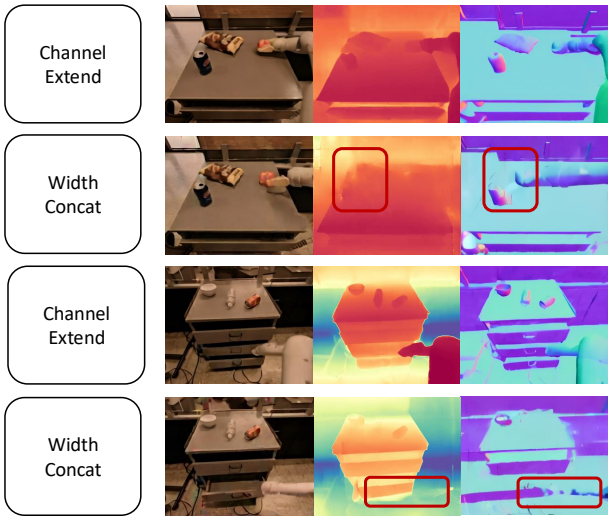

Figure 4. Channel concatenation improves visual quality and ensures consistency across maps.

Method	Chamfer L_1 RLBench	Chamfer L_1 RT-1
RGB to depth	0.2570	0.3013
4D Point Cloud Diffusion	0.1086	0.2211
Ours	**0.0945**	**0.2022**

Table 1. Comparison for 4D Generation on RLBench and RT-1

Method	Time Cost	Consistency Depth	Consistency Normal
Open-Sora	25.6 seconds	0.09267	0.04153
Marigold-LCM	**15.2 seconds**	0.09453	0.04647
Guided Marigold	~3.5 hours	**0.07299**	**0.03822**

Table 2. Comparison of Data Generation Methods

incorporate regularization loss $\mathcal{L}_g$ enforcing that optimized depths are similar to the generated depth map $\mathcal{D}^i$. We define the regularization loss $\mathcal{L}_g$ as:

$$\mathcal{L}_g(\mathcal{D}_1, \mathcal{D}_2, \mathcal{M}) = \|\mathcal{D}_1 \circ \mathcal{M} - \mathcal{D}_2 \circ \mathcal{M}\|^2 \quad (11)$$

We then define a regularization term on optimized depth maps over background regions of images $\lambda_{g2}\mathcal{L}g(\tilde{\mathcal{D}}, \hat{\mathcal{D}}^{i-1}, \mathcal{M}_b^i)$ enforcing that optimized depths for background regions of an image are consistent between frames and a regularization term over dynamic regions of images $\lambda_{g1}\mathcal{L}_g(\tilde{\mathcal{D}}, \mathcal{D}^i, M_d^i)$ enforcing that optimized depth of dynamic regions of an image match with predicted dynamic depths.

The overall loss objective we optimize is given by:

$$\arg\min_{\tilde{\mathcal{D}}} \ \mathcal{L}(\tilde{\mathcal{D}}, \mathcal{N}^i) + \lambda_b \mathcal{L}_g(\tilde{\mathcal{D}}, \hat{\mathcal{D}}^{i-1}, \mathcal{M}_b^i) + \quad (12)$$

$$\lambda_d \mathcal{L}_d(\tilde{\mathcal{D}}, \hat{\mathcal{D}}^{i-1}, \mathcal{M}_d^i, \mathcal{F}^i, \mathcal{F}^{i-1}) + \quad (13)$$

$$\lambda_{g2}\mathcal{L}_g(\tilde{\mathcal{D}}, \mathcal{D}^i, M_b^i) + \lambda_{g1}\mathcal{L}_g(\tilde{\mathcal{D}}, \mathcal{D}^i, M_d^i). \quad (14)$$

We initialize the starting depth $\tilde{d}_0 = \mathcal{D}^i$ with the generated depth map, and similar to prior work [9, 62], where we repeatedly optimize $\tilde{\mathcal{D}}^t$ across multiple iterations (using the previously optimized depth map $\tilde{\mathcal{D}}^{t-1}$ to define the new optimization objective).

Finally, we construct faces by connecting pixels to their nearby neighbors, resulting in a mesh derived from the optimized depth. Meshes provide a structured representation of the 3D geometry of a scene, enabling detailed surface reconstruction and facilitating further processing like rendering or physical simulations. For mesh denoising, we remove isolated vertices based on mean neighbor distances and eliminate small clusters using DBSCAN [22]. We also discard faces with abnormal normals or high edge-length variance, ensuring the final mesh is cleaner and more suitable for downstream tasks.

3.5. Inverse Dynamics Models from 4D Scenes

After generating 4D scenes, which encapsulate both spatial and temporal information, we extract geometric details that can significantly enhance downstream tasks in robotics. The detailed geometry captured by these scenes plays a crucial role in robotic grasping tasks.

To achieve this, we employ an inverse dynamics model built on the 4D meshes, predicting the appropriate robot action a_i based on the current state s_i, the predicted future state s_{i+1} and the instruction $\mathcal{T}$. Mathematically, this relationship is expressed as $a_i = \text{ID}(s_i, s_{i+1}, \mathcal{T})$. In our scenario, s_i represents the scene at time step i. Specifically, we sample the meshes to obtain point clouds, which are encoded by a PointNet [48] architecture within the inverse dynamics model to extract features. These features, combined with the instruction text embeddings, are further processed by an MLP to generate the final action.

4. Experiments

In this section, we evaluate the performance of our proposed model across several tasks. In Section 4.1, we present our experiments on 4D mesh prediction using the RLBench [27] and RT-1 [5] datasets. In Section 4.2, we conduct experiments on embodied novel view synthesis, using RLBench to assess our model's ability to generate novel views from monocular video inputs. In Section 4.3, we explore embodied action planning, applying our model to guide robotic arm policies for specific tasks. Finally, in Section 4.4, we present discussions and ablation studies that analyze the effect of different architectural and data generation choices on the quality and consistency of our video diffusion models. For additional experiments and results, see the Appendix.

4.1. 4D Scene Prediction

Since no prior work directly generates dynamic meshes from the first frame image and text inputs, we primarily compare our method to a 4D point cloud diffusion model. **Baseline**: our baselines include two main approaches. (1) The first is a 16-frame RGB diffusion model, where we obtain depth from the pre-trained depth estimator Marigold [30], and lift it to 3D via camera intrinsic and extrinsic parameters. (2) We also modify the Point-E [45] model by conditioning it on the mean of CLIP [49] features extracted from both text and image inputs, outputting a point cloud of size $T\times$num of points, where T is set to 4 due to computational constraints. **Dataset**: the datasets used for evaluation include RLBench [27] and our annotated RT-1 [5] dataset. **Metric**: for evaluation, we use the L_1 Chamfer Distance metric, which measures the distance between two point sets. The results are shown in Table 1.

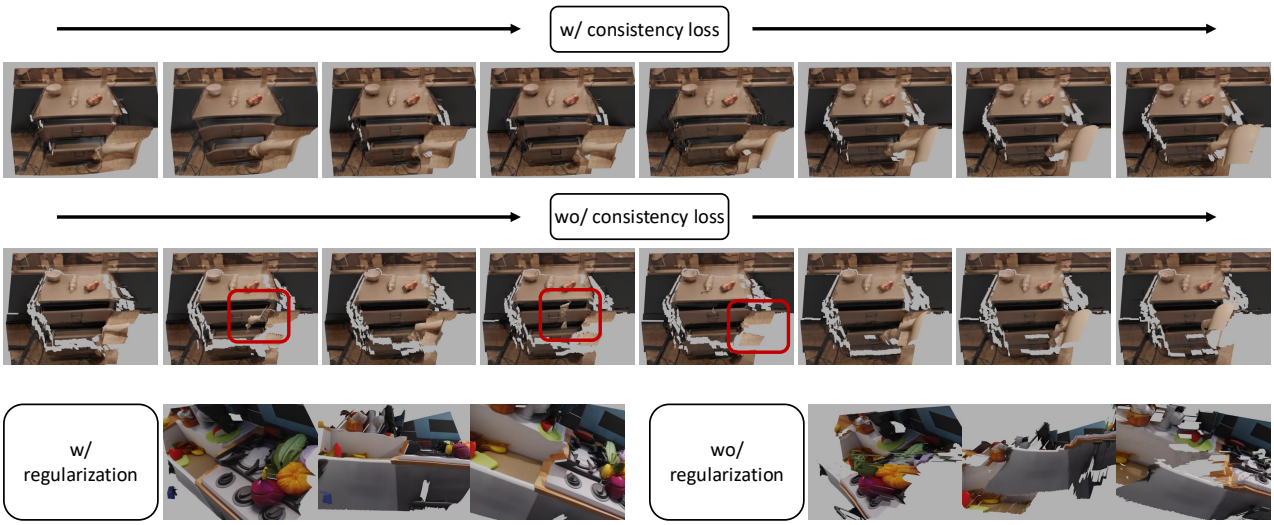

Figure 5. Effect of Consistency and Regularization Losses on 4D Mesh Reconstruction. The red boxes highlight the inconsistent regions.

As shown in the table, our method achieves the lowest Chamfer distances on both the RLBench [27] and RT-1 [5] datasets, indicating a more accurate reconstruction of 4D structures compared to the baselines. All methods perform better on RLBench [27], which is due to it being synthetic data, with less noise and perfectly accurate depth ground truth. The RGB-to-depth approach, while simple, suffers from larger errors due to the limitations of depth estimation from 2D images. The 4D point cloud diffusion method performs better, particularly on RLBench, but still lags behind our approach. Additionally, point cloud training is computationally expensive, restricting the number of frames used. In contrast, our model, by leveraging both image and text inputs, manages to generate more precise 3D representations, particularly in capturing fine-grained details in dynamic scenes. We show our qualitative results in Figure 3.

4.2. Embodied Novel View Synthesis

Our method performs monocular video to 4D tasks by predicting depth and normal sequences and generating meshes. **Baseline**: (1) we select Shape of Motion [58] as our primary baseline, a state-of-the-art video reconstruction approach that utilizes Gaussian splatting [31]. (2) Additionally, we include an RGB-to-depth approach, lifting depth to 3D and rendering point clouds for novel views. **Dataset**: since real-world datasets like RT-1 lack multiview camera information, we conduct experiments on RLBench. The input is a monocular front camera video, and we compare results from the overhead and left shoulder cameras. **Metric**: PSNR (reconstruction accuracy), SSIM (structural similarity), LPIPS (perceptual difference), CLIP Score (semantic match) [71], CLIP aesthetic (visual quality) [37], and Time costs.

Method	PSNR	SSIM	LPIPS	CLIP Score	CLIP aesthetic	Time costs
Shape of Motion	10.94	0.2402	**0.7382**	66.67	3.61	~2 hours
Ours	**12.99**	**0.4262**	0.6051	**83.02**	**3.73**	~ **1 minutes**

Table 3. Performance Comparison of Novel View Synthesis Methods on RLBench Dataset

4.3. Embodied Action Planning

Since our world model can predict future scenes, a direct application is to guide robotic arm policies. **Baseline**: we compare our method with re-implemented UniPi*, which is a 2D world model and uses an inverse dynamic policy to predict actions based on the videos. For this baseline, we fine-tune OpenSora [70] on RLBench data as the 2D world model and use a ResNet [23] in the image-based inverse dynamic model to encode both the current and predicted frames. Then, the encoded features and the text embeddings are passed through an MLP to obtain the 7-DoF action. **Dataset**: for simulation, we use RLBench [27] and collect 500 samples for each task to train our model. **Metric**: the success rate, which is returned by the simulator. For both the baseline and our method, we train the inverse dynamic model on collected RLBench data, with noise added to the corresponding modalities (image/3D point cloud). Given an initial state during inference, we first predict and record all future keyframes. In subsequent actions, we only query the inverse dynamic model to obtain the corresponding actions by the current state and the predicted future state. Table 5 reports the accuracy across different tasks.

The results show that our method outperforms video diffusion models across the selected tasks. This is because, in most tasks, 4D meshes or point clouds can reveal the geometry of objects, providing better spatial guidance for robotics planning, as seen in tasks like Close Box and Close Laptop. However, the performance of our model declines in tasks

Frames	Channel	RGB			Depth			Normal			Consistency
		FVD ↑	SSIM ↑	PSNR ↑	AbsRel ↓	δ_1 ↑	δ_2 ↑	Mean ↓	Median ↓	11.25° ↑	Edge-sim ↑
32	✗	20.12	70.23	19.32	30.22	59.21	80.28	54.22	40.87	6.41	28.27
32	✓	19.84	69.94	19.30	18.67	69.65	89.12	19.78	10.01	26.42	31.06
16	✗	**25.78**	71.89	21.86	**16.14**	76.59	91.54	26.59	15.73	24.36	19.67
16	✓	25.45	**74.98**	**21.94**	16.53	**77.13**	**92.15**	**16.23**	**7.78**	**38.50**	**32.98**

Table 4. Impact of the number of frames and channel concatenation. Frames refer to the number of input frames used in the model. "Channel" ✓ indicates concatenating RGB, depth, and normal maps along the channel dimension, while × refers to concatenation along the width.

Methods	Close Box	Close Laptop	Lift Block	Lamp Off
UniPi*	81.0	14.3	19.0	**57.1**
Ours	**95.2**	**28.6**	**23.8**	42.9

Table 5. Evaluation of action planning on RLBench dataset

like Lamp Off, due to the small size of the switch, which may not have been sampled. Overall, these results highlight the potential of combining 4D scene prediction with inverse dynamic models to improve robotics task execution.

4.4. Ablations

RGB-DN Video Diffusion Models We first conduct ablation studies on our video generation task. We perform experiments to explore the impact of different concatenation methods and the number of frames on the results. For the former, we compare two settings: (1) concatenating RGB-DN images along the width to form a larger image, or (2) using a VAE encoder [34, 56] to separately process RGB, depth, and normal maps, and concatenating them along the channel dimension before inputting them into the diffusion model. In the latter case, we also modify the input and output dimensions of the backbone network. Our evaluation metrics focus on the generation/reconstruction quality of RGB, depth, and normal maps. Additionally, we introduce an edge similarity metric to assess the consistency across RGB, depth, and normal maps at the same timestamp. Specifically, we convert them into gray-scale images, apply Canny edge detection [8], and compare the edge maps using SSIM [59].

Although concatenating images along the width results in better RGB reconstruction due to better utilization of the pre-trained Open-Sora model [70], it is less effective for depth and normal map predictions. Moreover, the inconsistency between RGB, depth, and normal maps prevents effective post-processing. As shown in Figure 4, concatenating along the channel dimension yields higher-quality depth and normal maps while maintaining consistency across the three value maps. This prevents issues such as a robotic arm appearing in different locations in the RGB and depth/normal maps.

Data Collection This part primarily compares the effects of guidance on depth and normal diffusion models during data generation. As shown in Table 2, "Marigold-LCM" refers to our use of the Marigold Latent Consistency Model [30] for independently predicting each frame. "Guided Marigold" represents our data generation method. We compare the time cost and the static part L_1 difference for these methods. As a reference, we provide the scores from our trained Open-Sora. Qualitative results are presented in the Appendix, where we observe that our proposed data generation method maintains the highest consistency, though at a significantly higher time cost. This also highlights the necessity of training a world model to rapidly predict and generate dynamic scenes.

Regularization and Consistency Loss in 4D Mesh Reconstruction. In this task, we evaluate the impact of our newly designed loss terms, as shown in Figure 5. The first two rows demonstrate the effect of the consistency loss, where we render frames from the same camera view at different time steps. The results show that the robot arm's movements are more coherent with the consistency loss applied. The last row highlights the role of the regularization loss. We display images of the same frame from three different views, revealing that this loss term helps improve the geometric accuracy of the reconstruction.

5. Conclusion

Our current approach has several limitations. First, while our RGB-DN representation of a 4D world model is cheap and easy to predict, it only captures a single surface of the world. To construct a more complete 4D world model, it may be interesting in the future to have a generative model that generates multiple RGB-DN views of the world, which can then be integrated to form a more complete 4D world model. In addition, we observe that generated RGB-DN maps from our 4D world model may not be fully consistent with each. Adding additional structure in the architecture or loss of function constraints at training time to help enforce consistency is a rich direction for future work.

Overall, our work provides some first steps towards the goal of constructing a 4D generative model of the world. We believe that such world models will be increasingly powerful and useful in the future, serving as a way to simulate the physical world and is an important step towards constructing intelligent embodied agents. Such models would then enable us to train policies in the real world in a fully offline manner, as well as roll out and imagine future plans in the world.

References

[1] Alessandro Achille and Stefano Soatto. A separation principle for control in the age of deep learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:287–307, 2018. 2

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. 2

[3] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3

[4] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019. 3

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3, 6, 7

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2, 3

[7] 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. 2

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 8

[9] Xu Cao, Hiroaki Santo, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Bilateral normal integration. In *European Conference on Computer Vision*, pages 552–567. Springer, 2022. 5, 6

[10] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022. 2

[11] Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*, 2017. 2

[12] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 2

[13] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 1, 2

[14] 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, 2024. 1, 2

[15] Jean-Denis Durou, Jean-François Aujol, and Frédéric Courteille. Integrating the normal field of a surface in the presence of discontinuities. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 261–273. Springer, 2009. 5

[16] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, pages 162–169, 2004. 2

[17] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, page 02783649241281508, 2023. 2

[18] Susan R Fussell, Leslie D Setlock, and Robert E Kraut. Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 513–520, 2003. 3

[19] Maitrey Gramopadhye and Daniel Szafir. Generating executable action plans with environmentally-aware language models. *arXiv preprint arXiv:2210.04964*, 2022. 2

[20] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 1, 2

[21] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. 2

[22] Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91(1):1–30, 2019. 6

[23] 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. 7

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[25] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 2

[26] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2

[27] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020. 3, 6, 7

[28] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1

[29] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3):6, 2022. 2

[30] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. 2, 3, 5, 6, 8

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 7

[32] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 2

[33] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3, 4

[34] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 8

[35] Danica Kragic, Mårten Björkman, Henrik I Christensen, and Jan-Olof Eklundh. Vision for robotic object manipulation in domestic settings. *Robotics and autonomous Systems*, 52(1): 85–100, 2005. 3

[36] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9–16, 2013. 3

[37] LAION-AI. Aesthetic predictor. https://github.com/LAION-AI/aesthetic-predictor, 2022. 7

[38] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018. 2

[39] Siyuan Li, Yue Luo, Ye Zhu, Xun Zhao, Yu Li, and Ying Shan. Enforcing temporal consistency in video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1145–1154, 2021. 3

[40] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022. 2

[41] 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. *arXiv preprint arXiv:2406.16862*, 2024. 1, 2

[42] Lennart Ljung and Torkel Glad. *Modeling of dynamic systems*. Prentice-Hall, Inc., 1994. 2

[43] Fei Luo, Lin Wei, and Chunxia Xiao. Stable depth estimation within consecutive video frames. In *Advances in Computer Graphics: 38th Computer Graphics International Conference, CGI 2021, Virtual Event, September 6–10, 2021, Proceedings 38*, pages 54–66. Springer, 2021. 3

[44] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022. 2

[45] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 6

[46] Antje Nuthmann and Teresa Canas-Bajo. Visual search in naturalistic scenes from foveal to peripheral vision: A comparison between dynamic and static displays. *Journal of Vision*, 22(1):10–10, 2022. 3

[47] OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2023. Accessed: 2024-10-01. 2

[48] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 6

[49] 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. 6

[50] Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Planning with large language models via corrective re-prompting. *arXiv preprint arXiv:2211.09935*, 2022. 2

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. 3

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

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

[55] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3, 5

[56] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3, 4, 8

[57] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang,

Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023. 3

[58] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 7

[59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 8

[60] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[61] Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024. 1, 2

[62] Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J. Black. ECON: Explicit Clothed humans Optimized via Normal integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6

[63] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 2, 3

[64] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 1, 2

[65] Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint arXiv:2303.04129*, 2023. 2

[66] Chongjie Ye, Lingteng Qiu, Xiaodong Gu, Qi Zuo, Yushuang Wu, Zilong Dong, Liefeng Bo, Yuliang Xiu, and Xiaoguang Han. Stablenormal: Reducing diffusion variance for stable and sharp normal. *arXiv preprint arXiv:2406.16864*, 2024. 5

[67] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023. 2

[68] Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Yilun Du, and Chuang Gan. Combo: Compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*, 2024. 1, 2

[69] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 2

[70] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. 1, 2, 3, 4, 7, 8

[71] SUN Zhengwentai. clip-score: CLIP Score for PyTorch. https://github.com/taited/clip-score, 2023. Version 0.1.1. 7

[72] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024. 2

[73] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024. 1

[74] Karl J. Åström and Björn Wittenmark. Adaptive control of linear time-invariant systems. *Automatica*, 9(6):551–564, 1973. 2