IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

FlowDreamer: A RGB-D World Model with Flow-based Motion Representations for Robot Manipulation

Jun Guo[1,2], Xiaojian Ma[1], Yikai Wang[3], Min Yang[1,4], Huaping Liu[2], and Qing Li[1]

***Abstract*—This paper investigates training better visual world models for robot manipulation, *i.e.*, models that can predict future visual observations by conditioning on past frames and robot actions. Specifically, we consider world models that operate on RGB-D frames (RGB-D world models). As opposed to canonical approaches that handle dynamics prediction mostly implicitly and reconcile it with visual rendering in a single model, we introduce FlowDreamer, which adopts 3D scene flow as explicit motion representations. FlowDreamer first predicts 3D scene flow from the past frame and action conditions with a U-Net, and then a diffusion model will predict the future frame utilizing the scene flow. FlowDreamer is trained end-to-end despite its modularized nature. We conduct experiments on 4 different benchmarks, covering both video prediction and visual planning tasks. The results demonstrate that FlowDreamer achieves better performance compared to other baseline RGB-D world models by 7% on semantic similarity, 11% on pixel quality, and 6% on success rate in various robot manipulation domains.**

***Index Terms*—Deep Learning in Grasping and Manipulation; Visual Learning; Deep Learning Methods**

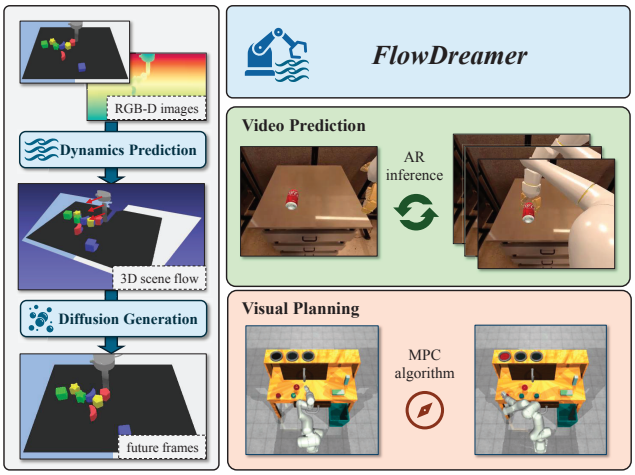

Fig. 1. **Proposed RGB-D world model with flow-based motion representations.** FlowDreamer adopts a two-stage prediction framework, which explicitly predicts scene flow as motion representations. FlowDreamer achieves better results on future frame prediction and visual planning tasks in various robot manipulation domains.

I. Introduction

We study developing better visual world models for robot manipulation tasks. In robotics, a visual world model [1] needs to perform the following steps: 1) **dynamics prediction**: predict the future motion given the current sensory observations (about robot and environment states) and robot action; and 2) **visual rendering**: render the visual observations after the motion happens. A visual world model captures the underlying world dynamics and can be used as a learnable simulator to help produce and evaluate motion plans of robot manipulators in company with model-based planning algorithms [2], [3], [4]. The use of visual world models alleviates the need for precise scene modeling and simulation [5], [6], making it a promising research direction. In this paper, we specifically focus on world models that operate on RGB-D frames (RGB-D world models), which are commonly adopted sensory observations in robot manipulation tasks.

Existing visual world models have undergone rapid development in recent years. Starting from early approaches that utilize recurrent neural networks (RNNs) [7], [8], [9], [10], powerful diffusion-based generative models [11], [12], [13], [14] have gained popularity in recent world models [15], [16], [17]. However, regardless of the architectures of these models, they mostly reconcile the two aforementioned steps (dynamics prediction and visual rendering) in a single model. These design choices not only reduce the model's transparency but also, as our later experiments show, impair its future prediction performance. We hypothesize that models trained solely with frame prediction loss tend to prioritize improving the fidelity of rendered visual appearances while placing less emphasis on accurate dynamics prediction. This highlights the importance of exploring methods that explicitly model dynamics prediction.

To this end, we propose FlowDreamer, a RGB-D world

Manuscript received July 3, 2025; revised October 27, 2025; accepted December 10, 2025.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. This paper is jointly supported by the National Natural Science Fund for Distinguished Young Scholars under grant No. 62025304, and the National Natural Science Foundation of China Young Scientists Fund under Grant No. 62306163. *Jun Guo, Xiaojian Ma, and Yikai Wang contributed equally to this work. (Corresponding authors: Xiaojian Ma; Huaping Liu; Qing Li.)*

[1]Jun Guo, Xiaojian Ma, Min Yang, and Qing Li are with the State Key Laboratory of General Artificial Intelligence (BIGAI), Beijing, China (e-mail: guo-j24@mails.tsinghua.edu.cn, jeasinema@gmail.com; yang.m2003@outlook.com; dylan.liqing@gmail.com)

[2]Jun Guo and Huaping Liu are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China (e-mail: hpliu@tsinghua.edu.cn)

[3]Yikai Wang is with the School of Artificial Intelligence, Beijing Normal University, Beijing, China (e-mail: yikaiw@outlook.com)

[4]Min Yang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

model that explicitly models dynamics prediction to enhance the predictive capability of world models. FlowDreamer adopts a two-stage framework to predict the environment dynamics and render the visual observations separately. Specifically, FlowDreamer introduces explicit modeling of 3D dynamics by leveraging 3D scene flow [18], which is a versatile representation that describes the motion of objects within a scene. In the first stage, a scene flow prediction module independently predicts the dynamic changes induced by given actions. This module is trained with a scene flow prediction loss to ensure robust supervision of the scene dynamics, thereby endowing the world model with an enhanced understanding of dynamics in 3D space. In the second stage, we employ a conditional diffusion model [13], [12] that predicts the next visual observation based on the current observation and the motion information provided by the scene flow prediction module. Despite its modularized nature, our model can be trained in an end-to-end fashion.

We validate the effectiveness of our method on multiple benchmarks commonly used in robotic manipulation. On action-conditioned benchmarks (*e.g.*, RT-1 [19], Language Table [20]), our approach achieves better visual performance comparable to normal RGB-D world models, with a 7% increase in semantic similarity and 11% on pixel quality. Evaluations on VP^2 visual planning benchmark [21] with RoboDesk [22] and Robosuite [23] tasks reveal a 6% increase in the success rate in manipulation tasks.

In summary, our main contributions are as follows:

1) We propose FlowDreamer, a two-stage RGB-D world model with dynamics and a visual prediction module, which can be trained in an end-to-end fashion despite its modularized nature.
2) We introduce a scene flow prediction module that adopts 3D scene flow on RGB-D space as a general motion representation to supervise dynamics prediction and enhance dynamics modeling.
3) We perform comprehensive evaluations across several benchmarks, demonstrating the efficacy of our approach in both visual performance and visual planning tasks.

II. Related Work

A. Generative World Models in Robotics

World models [1], also known as dynamic models, refer to techniques that predict the future state based on current observations and given conditions. In robotics, control strategies can leverage the generative capability of these models for planning, learning, or reducing the cost associated with interactions in the real environment. A world model typically requires the robot's current observation in the form of latent states, sensor measurements, or visual information as input and produces future observations while also potentially predicting higher-level information such as rewards, values, or subsequent actions.

Existing work in this domain can be broadly categorized into two types, namely *control-based* world models and *instruction-based* world models. In *control-based* world models [3], [4], [2], [24], [25], [26], the conditioning inputs are robot control signals, including joint forces or velocities, end-effector positions, or movement commands. These models can serve as simulators for the environment and facilitate approaches such as model-based reinforcement learning [27], [7], [8], [28], [29], [30], [31], [9] and model predictive control [10], [32]. Many studies [33], [17], [15] have also trained world models within games with players' interaction signals as input, employing neural networks as the game engine. In contrast, *instruction-based* world models [16], [34], [35], [36], [37], [38], [2] take task-related instructions, *e.g.*, task identifiers or natural language descriptions, as conditions to generate a sequence of future states or actions. This approach not only envisions future world states but also simulates the execution of policies, thereby allowing inverse dynamics to derive robot control parameters from observations [35], [37] or enabling fine-tuning into a vision-language-action (VLA) model [39], [40], [38], [41]. This paper proposes a *control-based* world model that uses robot actions as conditions. Unlike many prior control-based models that predict future frames in an end-to-end manner [3], [4], [2], our primary contribution is to decompose the task into two distinct stages: explicit dynamics prediction and visual rendering. We employ scene flow as an intermediate representation to simulate and predict environmental dynamics, ultimately generating images that are interpretable by human observers. This explicit motion representation can be visualized, offering a direct insight into the model's understanding of scene dynamics and providing a degree of interpretability into the model's learned physics.

B. Dynamics Modeling

Most existing world models are trained in an end-to-end manner with supervising signals to directly predict future states without explicitly modeling the dynamical processes. Explicit modeling of the dynamics underlying environmental state changes can enhance the interpretability of the world model and improve prediction accuracy. In the field of video generation, there are several works [42], [43], [44], [45], [46] that model the motion between frames to better control and enhance the performance of video generation models. Early works [47], [48], [49] employed SE(3) transformations as an intermediate representation of object motion. This approach is limited to rigid bodies and requires the world model to possess object-centric awareness, which requires extra segmentations of training data. An alternative representation is *flow* [50], [35], which is referred to as optical flow [51], [52] in 2D images and as scene flow [18] in 3D space. Flow is defined as the displacement of every point in the observation space from one timestep to the next, which provides a universal and flexible means to represent various forms of motion, including non-rigid objects. Our approach utilizes 3D scene flow in the RGB-D space as the representation of motion, which can be obtained by integrating optical flow from consecutive frames with the corresponding depth information. While prior works have explored flow for tasks like learning from actionless videos [35] or as a manipulation interface [50], to the best of our knowledge, this is the first work to utilize 3D scene flow as an explicit intermediate representation for a generative RGB-D

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

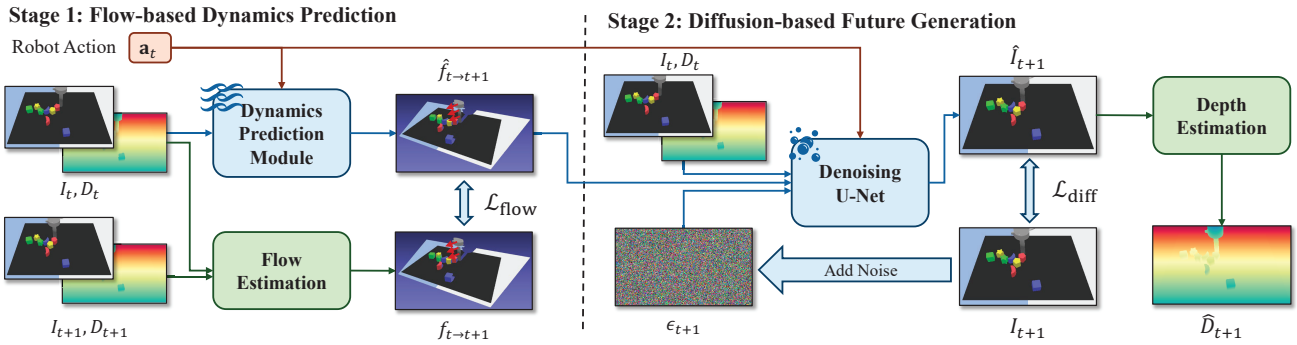

Fig. 2. **Overview of FlowDreamer.** At stage 1, FlowDreamer receives the RGB-D frame and the robot action as input to explicitly predict the scene flow as motion representations. At stage 2, FlowDreamer leverages a denoising U-Net to generate high-resolution next-step future observation via diffusion.

world model in robotics. Our work investigates this direction, hypothesizing that decoupling dynamics from rendering via 3D scene flow can improve prediction accuracy.

III. FlowDreamer: A RGB-D World Model

In this section, we illustrate the framework of our FlowDreamer. Fig. 2 depicts the overall framework of our method. FlowDreamer is a two-stage action-conditioned RGB-D world model, which receives the current RGB-D observation (I_t, D_t) and the robot action $\mathbf{a}_t$ as the input and predicts the future RGB observations I_{t+1}. At stage 1, we start from an RGB-D observation image and a robot action, predicting the 3D scene flow $f_{t\to t+1}$ between the current and the next frame (Sec. III-B). At stage 2, we apply a diffusion denoising network condition on the RGB-D image (I_t, D_t) and the predicted scene flow $\hat{f}_{t\to t+1}$ to generate the observation I_{t+1} at the next timestep from a random noise ϵ_{t+1} (Sec. III-C).

A. Background: Latent Diffusion Models

Our future generation module is built upon Latent Diffusion Models (LDMs) [11], which are a class of generative models that have demonstrated state-of-the-art performance in high-fidelity image synthesis. LDMs operate by first encoding high-dimensional images I into a lower-dimensional latent space $\mathbf{z}$ using a pre-trained autoencoder. A diffusion process [14], [13], [12] then iteratively adds noise to these latent representations. A neural network, typically a U-Net, is trained to reverse this process, learning to denoise the latent vector conditioned on inputs like text or, in our case, previous states and actions. During inference, a random latent vector is iteratively denoised to generate a new sample, which is then decoded back into pixel space. For a detailed formulation, we refer the reader to the original works [14], [13], [12].

B. Dynamics Prediction

Different from end-to-end world models, FlowDreamer applies a dynamics prediction module to explicitly predict the transition between consecutive frames. In robot manipulation tasks, we hope to find an intermediate representation as auxiliary information to depict the dynamics of the robot and objects. Innovated by [48], [35], [50], we choose 3D scene flow as the intermediate representation, which is general and versatile and can be collected by various flow estimation networks. In point clouds, scene flow is generally defined by the displacement of point coordinates. We can easily project an RGB-D image into a point cloud with camera intrinsics. Assume the point in 3D space has a coordinate (x, y, z) at timestep t and (x', y', z') at timestep $t+1$, the 3D scene flow is defined as follows:

$$f_{t\to t+1} = (x' - x, y' - y, z' - z). \tag{1}$$

As pixels at the same index between consecutive frames do not always represent the same 3D point, the vital challenge to get the scene flow is to find the correspondence between frames. For simulation data, we can directly obtain the scene flow from the simulator backend according to the rigid transformations of every object. For real-world data, we can apply an off-the-shelf 3D scene flow estimator named RAFT-3D [53] to estimate the scene flow. If the real-world data has no depth information, we can estimate the depth by a pretrained video depth estimator [54].

We apply a dynamics prediction module to predict the 3D scene flow from time t to $t+1$, given the RGB-D observation (I_t, D_t) and the robot action $\mathbf{a}_t$. The backbone of our dynamics prediction model is a conditional U-Net [55]. The input RGB-D frame is processed through an encoder backbone with 4 downsampling blocks. The robot action a_t is projected to an embedding and integrated into the U-Net's bottleneck features via cross-attention layers. The decoder then uses 4 upsampling blocks with skip connections from the encoder to predict the 3D scene flow $\hat{f}_{t\to t+1}$. The loss function $\mathcal{L}_{\text{flow}}$ is defined as the mean square error (MSE) between the module output $\hat{f}_{t\to t+1}$ and the estimated scene flow $f_{t\to t+1}$:

$$\mathcal{L}_{\text{flow}} = \text{MSELoss}(\hat{f}_{t\to t+1} - f_{t\to t+1}). \tag{2}$$

C. Future Generation

After getting the predicted scene flow, we can further predict future observations through a generative model. In FlowDreamer, we fine-tune the pre-trained Stable Diffusion [11] to build our future generation module, which we denote as ϵ_θ. The input of the generation module contains the current RGB-D observation (I_t, D_t), the robot action $\mathbf{a}_t$, and the predicted 3D scene flow $\hat{f}_{t\to t+1}$. The action is included as a condition in this second stage for two key reasons. First, the predicted scene flow can be imperfect, and the action serves as

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

a precise, error-free instruction that helps the model correct for these inaccuracies. Second, scene flow derived from a single-view RGB-D image represents local information and may not capture the full motion of partially-observed objects like the robot arm; providing the action $\mathbf{a}_t$ ensures this information is not lost. The output of the module is the next RGB observation I_{t+1}. We modify the denoising U-Net by adding extra input channels to its first convolutional layer to accept the conditioning information. The robot action $\mathbf{a}_t$ is provided as a conditioning signal to the cross-attention layers of the U-Net, replacing the standard text embeddings. We use the pre-trained variational autoencoder in Stable Diffusion [11] to compress the RGB observation I_t into latent space $\mathbf{z}_t$. The module output $\mathbf{z}_{t+1}$ is also in latent space and can be decoded to image space by the pretrained decoder. During the generative diffusion process for this next frame, we denote the noised latent at diffusion step k as $\mathbf{z}_{t+1}^k$. Depth map D_t and the scene flow $\hat{f}_{t\to t+1}$ are firstly downsampled to the same shape of $\mathbf{z}_t$ by several convolutional layers, and then channel-wise concatenated with $\mathbf{z}_t$ as the input of the diffusion model:

$$\mathbf{z}_t = \text{VAE}(I_t), \tag{3}$$

$$\mathbf{c}_t = \text{downsample}(D_t, \hat{f}_{t\to t+1}), \tag{4}$$

$$\hat{\epsilon}^k = \epsilon_\theta(\text{concat}(\mathbf{z}_{t+1}^k, \mathbf{z}_t, \mathbf{c}_t), \mathbf{a}_t, k), \tag{5}$$

where k is the diffusion step, and z^k is the latent to be denoised. We do not directly predict the depth map D_{t+1}. Our model is characterized as an RGB-D world model because it takes RGB-D data as input to understand the 3D scene structure for dynamics prediction. However, the generative process focuses on the next RGB frame I_{t+1} because the decoder of the pre-trained diffusion model is not designed for metric depth generation, which would compromise the quality of the generated depth. To autoregressively imagine the future, we leverage a pre-trained depth estimation model [56] with a simple DPT [57] head to predict the metric depth D_{t+1} from I_{t+1}.

In summary, our FlowDreamer trains the dynamics prediction module and the denoising U-Net jointly, and the overall loss function is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{diff}} + \alpha\mathcal{L}_{\text{flow}}, \tag{6}$$

where $\mathcal{L}_{\text{diff}}$ is the standard diffusion loss, $\mathcal{L}_{\text{flow}}$ is defined in Eqn. 2, and α is a coefficient to control the weight of the two objectives.

IV. Experiments

In this section, we conduct extensive experiments in four different benchmarks to verify the performance of FlowDreamer. We aim to answer three questions:

1) Is FlowDreamer effective on video prediction compared with other RGB-D world models? (Sec. IV-A)
2) Can FlowDreamer facilitate model predictive control for robot manipulation tasks? (Sec. IV-B)
3) How does the predicted scene flow in FlowDreamer contribute to the future prediction? (Sec. IV-C)

Except for the results shown in the paper, we encourage the reader to view our supplementary video, which provides qualitative results and comparisons.

A. Video Prediction

To evaluate the future generation performance of FlowDreamer, we conduct the video prediction experiments, which require the world model to generate a full trajectory given the first frame of the video and the action trajectory. To generate longer sequences, we apply the model autoregressively, starting with the first ground truth frame and generating subsequent frames for the full duration of the provided action trajectory. The prediction horizon is therefore variable, matching the trajectory length of each sample in the test set. On average, this corresponds to a horizon of 30 frames for the SimplerEnv RT-1 dataset and 40 frames for the Language Table dataset. By comparing the similarity between predicted frames and ground truth frames, we can verify the future prediction capability of world models.

Datasets. We conduct video prediction experiments on RT-1 [19] and Language Table [20] benchmarks to evaluate the methods. As the real-world data do not contain the depth information, we collect trajectories and evaluate in the simulator to obtain the accurate metric depth and scene flow. For RT-1 environment, we use SimplerEnv [58] as the simulator. For the Language Table environment, we use the official simulation environment. Collected trajectories are split into training, validation, and test sets without overlap.

Baselines. To evaluate the performance of FlowDreamer compared to other RGB-D world models, we design three different baselines: *Vanilla*, *MinkNet*, *SepTrain*, and *FlowOnly*.

- *Vanilla* is a single-stage RGB-D diffusion model, which receives current RGB-D observations and the robot action as input and predicts RGB images at the next timestep. We compare it with our FlowDreamer to measure the contribution of the dynamics prediction module.
- *MinkNet* is a two-stage world model that replaces the backbone of the dynamics prediction module from U-Net to MinkowskiNet [59], a 4D sparse convolutional network for point clouds. We compare it with our FlowDreamer to demonstrate the effectiveness of the RGB-D representation.
- *SepTrain* is a two-stage world model, which trains the dynamics prediction module and the denoising U-Net separately. We compare it with our FlowDreamer to evaluate the performance of end-to-end training.
- *FlowOnly* is identical to our full FlowDreamer model except that the robot action $\mathbf{a}_t$ is removed as a condition for the Stage 2 denoising U-Net. The denoising U-Net must predict the future frame based only on the current state (I_t, D_t) and the predicted scene flow $\hat{f}_{t\to t+1}$.

Metrics. To evaluate the video prediction performance, we use PSNR [60], SSIM [61], LPIPS [62], FID [63], and FVD [64] as the assessment metrics, and additionally calculate the latent L2 distance extracted by DINOv2 (denoted as DINOv2 L2) and the latent cosine similarity extract by CLIP (denoted as CLIP score) to estimate the semantic similarity. PSNR measures

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I

VIDEO PREDICTION RESULTS ON THE SIMPLERENV RT-1 AND LANGUAGE TABLE BENCHMARK. WE CATEGORIZE THE METRICS INTO THREE GROUPS: SEMANTIC SIMILARITY, PIXEL SIMILARITY, AND MEDIA QUALITY. **BOLD** NUMBERS INDICATE THE BEST RESULTS, AND UNDERLINED NUMBERS INDICATE THE SECOND BEST RESULTS. NUMBERS IN THE BRACKET INDICATE THE STANDARD DEVIATION ACROSS 5 RANDOM SEEDS.

Dataset	Method	Semantic Similarity		Pixel Similarity			Media Quality	
		DINOv2 L2	CLIP score	PSNR	SSIM	LPIPS	FID	FVD
SimplerEnv RT-1	Vanilla	12.59 (0.24)	0.8999 (0.0508)	20.59 (0.33)	0.7831 (0.0257)	0.1304 (0.0158)	71.81 (13.97)	365.07 (39.45)
	MinkNet	12.16 (0.21)	0.9038 (0.0491)	20.58 (0.35)	0.7942 (0.0265)	0.1252 (0.0162)	57.97 (12.89)	325.34 (37.52)
	SepTrain	11.45 (0.25)	0.9131 (0.0477)	21.23 (0.31)	0.8135 (0.0248)	0.1097 (0.0145)	45.40 (11.53)	**245.91** (35.19)
	FlowOnly	12.48 (0.30)	0.9011 (0.0612)	19.83 (0.43)	0.7714 (0.0324)	0.1471 (0.0202)	88.77 15.42)	414.23 (56.16)
	FlowDreamer (Ours)	**10.99** (0.23)	**0.9189** (0.0465)	**21.76** (0.30)	**0.8196** (0.0241)	**0.0993** (0.0139)	**43.58** (11.21)	268.39 (36.04)
Language Table	Vanilla	10.73 (0.26)	0.9473 (0.0412)	25.68 (0.38)	0.9273 (0.0213)	0.0627 (0.0112)	26.64 (5.15)	110.46 (18.99)
	MinkNet	10.01 (0.22)	0.9614 (0.0398)	25.20 (0.36)	0.9228 (0.0231)	0.0642 (0.0115)	33.55 (5.88)	88.39 (15.14)
	SepTrain	10.25 (0.24)	0.9507 (0.0405)	25.98 (0.37)	0.9278 (0.0210)	0.0571 (0.0109)	21.28 (5.97)	87.31 (14.81)
	FlowOnly	10.66 (0.28)	0.9494 (0.0441)	24.86 (0.41)	0.9195 (0.0232)	0.0664 (0.0183)	42.09 (6.11)	137.62 (20.13)
	FlowDreamer (Ours)	**9.39** (0.21)	**0.9688** (0.0381)	**26.89** (0.29)	**0.9401** (0.0188)	**0.0476** (0.0097)	**20.03** (4.55)	**66.92** (13.10)

TABLE II

VIDEO PREDICTION RESULTS ON RT-1 REAL-WORLD DATASET. WE CATEGORIZE THE METRICS INTO THREE GROUPS: SEMANTIC SIMILARITY, PIXEL SIMILARITY, AND MEDIA QUALITY. **BOLD** NUMBERS INDICATE THE BEST RESULTS. NUMBERS IN THE BRACKET INDICATE THE STANDARD DEVIATION ACROSS 5 RANDOM SEEDS.

Method	Semantic Similarity		Pixel Similarity			Media Quality	
	DINOv2 L2	CLIP score	PSNR	SSIM	LPIPS	FID	FVD
Vanilla	15.67 (0.32)	0.8618 (0.0533)	17.97 (0.43)	0.5401 (0.0307)	0.1882 (0.0202)	13.16 (2.36)	195.40 (15.29)
FlowDreamer (Ours)	**15.07** (0.30)	**0.8801** (0.0504)	**19.69** (0.44)	**0.5982** (0.0289)	**0.1764** (0.0177)	**10.45** (1.87)	**162.31** (13.58)

the distance between the predicted video and the ground-truth video in the pixel space, and SSIM evaluates the structural similarity between frames. LPIPS, FID, and FVD are model-based evaluation metrics that compare frames in latent feature space. LPIPS measures the distance in different feature spaces, while FID and FVD measure the distribution disparity between generated and ground-truth images or videos. DINOv2 and CLIP are large-scale pretrained models via unsupervised learning, which could robustly extract the semantic feature from the images. We assume that DINOv2 and CLIP feature distance could reflect the environment state information more than image quality metrics.

Results. Table I shows the quantitative results on RT-1 and Language Table benchmarks. We can observe that our FlowDreamer achieves the best performance on most of the metrics, including semantic similarity, pixel similarity, and media quality. The results demonstrate the effectiveness of our FlowDreamer. The *SepTrain* model achieves the second-best performance at most of the metrics, and the performance is very similar to our FlowDreamer. This indicates that end-to-end training is generally a better approach, while the contribution is less than that of other components. We notice that *Vanilla* and *MinkNet* have similar performances, though *MinkNet* has a similar two-stage framework and a dynamics prediction module. For results on *FlowOnly*, We observed a severe degradation in prediction quality. The *FlowOnly* model performed significantly worse than our full FlowDreamer model and, notably, its prediction quality was even substantially lower than that of the *Vanilla* baseline, which lacks an explicit dynamics module entirely. It demonstrates that the robot action modality plays a critical and complementary role in the scene flow. The failure of the FlowOnly model suggests that relying solely on a potentially imperfect predicted flow is insufficient.

Real-world experiments. We conduct real-world experiments to evaluate the feasibility of our pipeline in the real world. We conduct video prediction experiments on the real-world RT-1 robot manipulation dataset. The RT-1 real-world dataset contains more tasks, lighting conditions, and camera positions, making it a much harder task than on the simulation data. As there is no ground truth depth or scene flow, we leverage the estimation results as the training target. We compare FlowDreamer with the single-stage *Vanilla* world model.

Table II shows the performance of video prediction. FlowDreamer still performs better than *Vanilla*, while the discrepancy becomes lower than simulation data. We observe that the ground truth scene flows estimated by RAFT-3D are not accurate, which would introduce noise into the training target. Our FlowDreamer is affected by inaccurate supervision and produces inaccurate flow predictions. However, our performance still outperforms *Vanilla*, as *Vanilla* cannot even keep the consistency of the background during generation.

B. Visual Planning

We further evaluate our FlowDreamer on visual planning tasks to show how FlowDreamer makes a difference in robot manipulation tasks. In visual planning tasks, the policy interacts with environments to minimize the difference between the observation and the goal image. For world models without any action output, model predictive control (MPC) [25], [7] methods can be applied to evaluate the performance.

Datasets. We choose VP^2 [21] as our visual planning benchmark. VP^2 is a control-centric benchmark that evaluates video prediction models by visual MPC methods. The environment contains four Robosuite [23] and seven RoboDesk [22] tasks. We run our experiments with 4 seeds on RoboDesk tasks and 3 seeds with Robosuite tasks. We strictly follow the experimental protocol of the VP^2 benchmark. All baselines

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

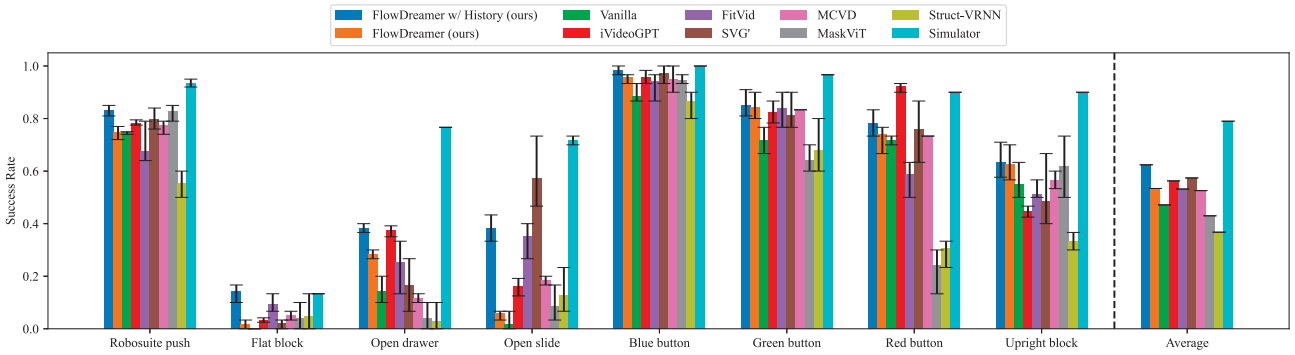

Fig. 3. **Visual planning results on the VP^2 benchmark.** We report the mean and the min/max performance of different methods over multiple runs with different random seeds. On the right, "Average" means the average success rate over all reported tasks.

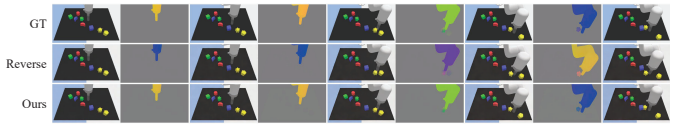

Fig. 4. **The qualitative results when flows are reversed.** With reversed (therefore incorrect) scene flow, the diffusion model in FlowDreamer can only utilize action condition, leading to worse performances on future frame prediction.

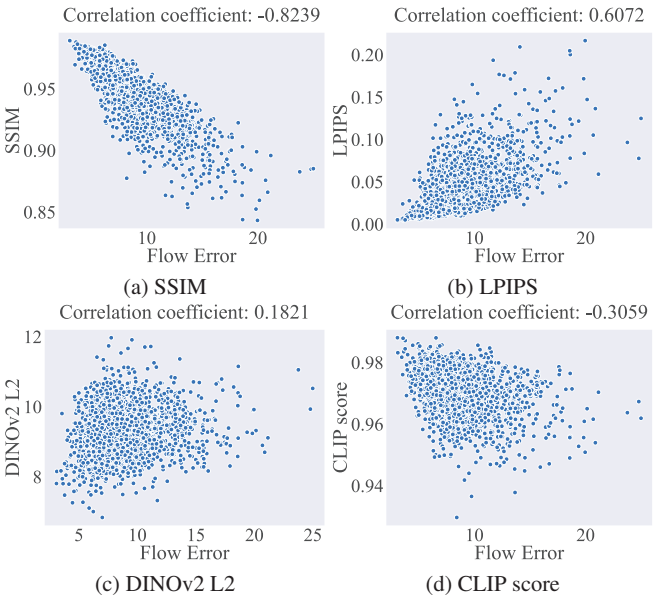

Fig. 5. **The correlation between the flow prediction error and image assessment metrics.** We show the scatter plots of SSIM (higher is better), LPIPS (lower is better), DINOv2 L2 (lower is better), and CLIP score (higher is better) vs. flow error and report the correlation coefficients.

and evaluation settings are adopted directly from VP^2 to ensure a fair comparison. For a complete and detailed description of all hyperparameters, such as the planning horizon, action space definitions for each task, and the number of optimization iterations, we refer the reader to the original VP^2 paper [21].

Baselines. We compare the visual planning performance with a single-stage RGB-D diffusion world model (denoted as *Vanilla*). Moreover, following iVideoGPT [4], we choose all video generation models provided by VP^2 paper as our baselines, including FitVid [65], SVG' [66], MCVD [67], Struct-VRNN [68], and MaskViT [69]. We also compare our performance with iVideoGPT itself.

Metrics. Following iVideoGPT [4], we report the minimum, maximum, and average success rate of our method between different random seeds. The reported baseline results are provided by previous works [4], [21], and we additionally report our performance in the same setting. For Robosuite push tasks, a cost below 0.05 is considered a success.

Results. Fig. 3 shows the visual MPC results on the VP^2 benchmark. From the results, we found that our FlowDreamer always performs better than the *Vanilla* model, which implies that our proposed framework works well on visual planning tasks. For other video prediction models, FlowDreamer achieves a comparable performance on the average metric, while all video prediction baselines have a context of two frames. To ensure a fair comparison with prior work on the VP^2 benchmark with two-frame history, we developed a multi-frame version of FlowDreamer. As shown in Fig. 4, this model now achieves state-of-the-art performance on 6 of 8 tasks and suboptimal performance on the remaining tasks. This demonstrates the effectiveness of our flow-based dynamics prediction, especially when provided with sufficient historical context.

Our initial analysis suggested that a single-frame input limited the model's ability to infer second-order dynamics like object velocity, which is critical for tasks like Robosuite push. This put the model at a disadvantage compared to baselines using a longer history. Our new experiments confirm this hypothesis decisively. By extending FlowDreamer to use a two-frame history, the model gains the ability to perceive motion and consequently achieves state-of-the-art performance as shown in our updated results. This highlights that the core contribution of our work—explicit dynamics prediction via scene flow—is highly effective and benefits significantly from sufficient temporal context.

C. Additional Analysis on Flow Prediction

In this section, we conduct further analysis to figure out the effect of the predicted flow. We first reverse the direction of input flows at stage 2 while the robot action remains unchanged. The reversed flow maintains a similar structure to the original flow but possesses a completely incorrect

direction. This can help us determine whether the model truly utilizes the directional information from the flow to generate the next frame, or if it merely uses the flow information as a mask to identify regions of motion. Fig. 4 visualizes the resulting prediction. We can observe that the robot did not really take contrary actions due to the action input at stage 2, while its performance becomes worse and cannot lead to the goal state. This result shows that the scene flow predicted at stage 1 provides auxiliary information to better generate the future.

Then, we calculate the correlation between the flow prediction error and other image assessment metrics. Fig. 5 shows the correlation coefficient r between flow prediction error and other metrics, and the scatter plot for SSIM, LPIPS, DINOv2 L2, and CLIP score. We notice that the flow error has a high correlation with SSIM and a reasonable correlation with LPIPS, which demonstrates the effectiveness of the predicted flow in video prediction. For DINOv2 L2 and CLIP scores, the correlations are weak, where we infer that the semantic metrics extracted by DINOv2 and CLIP do not indicate the relatively minor prediction error from the ground truth well.

V. Conclusion and Limitations

We introduce FlowDreamer, an action-conditioned RGB-D world model with flow-based motion representations. FlowDreamer leverages 3D scene flow as a versatile motion representation and applies a separate dynamics prediction module to learn environment dynamics from scene flow. Despite its modularized nature, FlowDreamer can be jointly trained in an end-to-end manner. Experiments on 4 different benchmarks, including video prediction and visual planning, demonstrate the superiority of our FlowDreamer compared to other RGB-D world models.

Limitations. Though with promising results, our work has several limitations that open avenues for future research.

First, the performance of our model is inherently tied to the quality of the depth map and 3D scene flow. In scenarios with significant occlusions, transparent or reflective objects, estimating scene flow is challenging. Errors in the Stage 1 flow prediction can propagate and lead to inaccurate future frame generation in Stage 2. Developing methods to make the generation process more robust to noisy flow predictions is an important area for improvement.

Then, for real-world applications, our method relies on an off-the-shelf flow estimator (RAFT-3D) or depth estimator when ground truth is unavailable. The domain gap between the training data of these estimators and the target robotic environment can introduce errors. Fine-tuning these components or developing self-supervised methods for flow estimation could mitigate this issue.

Finally, we did not include an empirical comparison against object-centric models. These two classes of methods operate under fundamentally different design philosophies and assumptions, making a direct comparison potentially inequitable. Object-centric world models often require stronger priors, which can lead to high performance but restrict their generalizability on non-rigid objects. Our approach, in contrast, is intended as a more universal representation designed to handle a wider range of scenarios with fewer assumptions. A thorough evaluation of tasks involving non-rigid objects is an important direction for future work.

References

[1] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

[3] F. Zhu, H. Wu, S. Guo, Y. Liu, C. Cheang, and T. Kong, "Irasim: Learning interactive real-robot action simulators," *arXiv preprint arXiv:2406.14540*, 2024.

[4] J. Wu, S. Yin, N. Feng, X. He, D. Li, J. Hao, and M. Long, "ivideogpt: Interactive videogpts are scalable world models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[5] T. Dai, J. Wong, Y. Jiang, C. Wang, C. Gokmen, R. Zhang, J. Wu, and L. Fei-Fei, "Automated creation of digital cousins for robust policy learning," *arXiv preprint arXiv:2410.07408*, 2024.

[6] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5616–5626.

[7] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[8] Ł. Kaiser, M. Babaeizadeh, P. Miłos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.*, "Model based reinforcement learning for atari," in *International Conference on Learning Representations (ICLR)*, 2020.

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

[10] N. A. Hansen, H. Su, and X. Wang, "Temporal difference learning for model predictive control," in *International Conference on Machine Learning (ICML)*, 2022.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning (ICML)*, 2015.

[15] J. Bruce, M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, *et al.*, "Genie: Generative interactive environments," in *International Conference on Machine Learning (ICML)*, 2024.

[16] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, *et al.*, "Cosmos world foundation model platform for physical ai," *arXiv preprint arXiv:2501.03575*, 2025.

[17] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret, "Diffusion for world modeling: Visual details matter in atari," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[18] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *International Conference on Computer Vision (ICCV)*, 1999.

[19] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[20] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[21] S. Tian, C. Finn, and J. Wu, "A control-centric benchmark for video prediction," in *International Conference on Learning Representations (ICLR)*, 2023.

[22] H. Kannan, D. Hafner, C. Finn, and D. Erhan, "Robodesk: A multi-task reinforcement learning benchmark," 2021. [Online]. Available: https://github.com/google-research/robodesk

[23] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[24] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[25] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *International Conference on Robotics and Automation (ICRA)*, 2017.

[26] H. Shao, D. Xie, and Y. Huang, "A survey of intelligent sensing technologies in autonomous driving," *ZTE Communications*, vol. 19, no. 3, p. 56, 2021.

[27] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, 1991.

[28] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, 2020.

[29] R. Sun, H. Zang, X. Li, and R. Islam, "Learning latent dynamic robust representations for world models," in *International Conference on Machine Learning (ICML)*, 2024.

[30] J. Yu and Y. Chen, "A practical reinforcement learning framework for automatic radar detection," *ZTE Communications*, vol. 21, no. 3, p. 22, 2023.

[31] J. Shen, K. Jiang, and X. Tan, "Boundary data augmentation for offline reinforcement learning," *ZTE Communications*, vol. 21, no. 3, p. 29, 2023.

[32] N. Hansen, H. Su, and X. Wang, "Td-mpc2: Scalable, robust world models for continuous control," *arXiv preprint arXiv:2310.16828*, 2023.

[33] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[34] S. Huang, L. Chen, P. Zhou, S. Chen, Z. Jiang, Y. Hu, P. Gao, H. Li, M. Yao, and G. Ren, "Enerverse: Envisioning embodied future space for robotics manipulation," *arXiv preprint arXiv:2501.01895*, 2025.

[35] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum, "Learning to act from actionless videos through dense correspondences," in *International Conference on Learning Representations (ICLR)*, 2023.

[36] S. Zhou, Y. Du, J. Chen, Y. LI, D.-Y. Yeung, and C. Gan, "Robodreamer: Learning compositional world models for robot imagination," in *International Conference on Machine Learning (ICML)*, 2024.

[37] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pre-trained image-editing diffusion models," in *International Conference on Learning Representations (ICLR)*, 2024.

[38] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[39] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, "3d-vla: A 3d vision-language-action generative world model," in *International Conference on Machine Learning (ICML)*, 2024.

[40] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, *et al.*, "Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation," *arXiv preprint arXiv:2410.06158*, 2024.

[41] Y. Du, S. Yang, P. Florence, F. Xia, A. Wahid, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. P. Kaelbling, *et al.*, "Video language planning," in *International Conference on Learning Representations (ICLR)*, 2024.

[42] Z. Hao, X. Huang, and S. Belongie, "Controllable video generation with sparse trajectories," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[43] A. Mahapatra and K. Kulkarni, "Controllable animation of fluid elements in still images," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[44] H. Ni, C. Shi, K. Li, S. X. Huang, and M. R. Min, "Conditional image-to-video generation with latent flow diffusion models," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[45] D. Geng, C. Herrmann, J. Hur, F. Cole, S. Zhang, T. Pfaff, T. Lopez-Guevara, C. Doersch, Y. Aytar, M. Rubinstein, *et al.*, "Motion prompting: Controlling video generation with motion trajectories," *arXiv preprint arXiv:2412.02700*, 2024.

[46] X. Shi, Z. Huang, F.-Y. Wang, W. Bian, D. Li, Y. Zhang, M. Zhang, K. C. Cheung, S. See, H. Qin, *et al.*, "Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling," in *ACM SIGGRAPH Conference Proceedings*, 2024.

[47] I. Nematollahi, E. Rosete-Beas, S. M. B. Azad, R. Rajan, F. Hutter, and W. Burgard, "T3vip: Transformation-based 3d video prediction," in *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[48] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3d dynamic scene representations for robot manipulation," in *Conference on Robot Learning (CoRL)*, 2020.

[49] A. Byravan and D. Fox, "Se3-nets: Learning rigid body motion using deep neural networks," in *International Conference on Robotics and Automation (ICRA)*, 2017.

[50] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," in *Conference on Robot Learning (CoRL)*, 2024.

[51] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, 1981.

[52] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981.

[53] Z. Teed and J. Deng, "Raft-3d: Scene flow using rigid-motion embeddings," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang, "Video depth anything: Consistent depth estimation for super-long videos," *arXiv preprint arXiv:2501.12375*, 2025.

[55] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention (MICCAI)*, 2015.

[56] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[57] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *International Conference on Computer Vision (ICCV)*, 2021.

[58] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[59] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[60] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, 2008.

[61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Transactions on Image Processing (TIP)*, 2004.

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[63] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[64] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[65] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, "Fitvid: Overfitting in pixel-level video prediction," *arXiv preprint arXiv:2106.13195*, 2021.

[66] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, "High fidelity video prediction with large stochastic recurrent neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[67] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Mcvd-masked conditional video diffusion for prediction, generation, and interpolation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[68] M. Minderer, C. Sun, R. Villegas, F. Cole, K. P. Murphy, and H. Lee, "Unsupervised learning of object structure and dynamics from videos," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[69] A. Gupta, S. Tian, Y. Zhang, J. Wu, R. Martín-Martín, and L. Fei-Fei, "Maskvit: Masked visual pre-training for video prediction," in *International Conference on Learning Representations (ICLR)*, 2023.