This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MoMaps: Semantics-Aware Scene Motion Generation with Motion Maps

Jiahui Lei[1,2] Kyle Genova[1] George Kopanas[3] Noah Snavely[1] Leonidas Guibas[1]
[1] Google DeepMind [2] University of Pennsylvania [3] Google
leijh@cis.upenn.edu, {jiahuilei,kgenova,gkopanas,snavely,guibas}@google.com

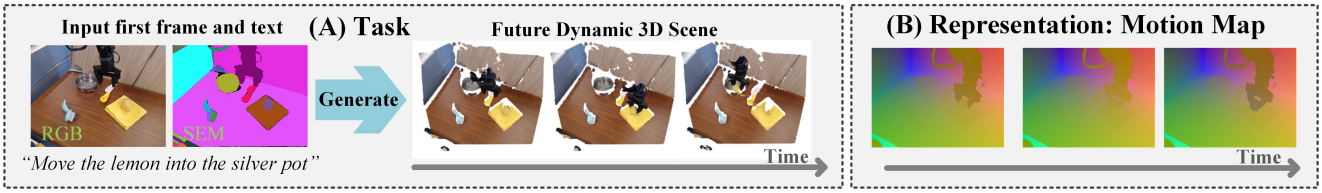


Figure 1. (A) Given the first time frame color and segmentation image and a text prompt, our model generates the future dynamic 3D scene. (B) 3D dynamic scenes as pixel-aligned curve/trajectory images, re-purposing an image diffusion model for 4D generation.

Abstract

This paper addresses the challenge of learning semantically and functionally meaningful 3D motion priors from real-world videos, in order to enable prediction of future 3D scene motion from a single input image. We propose a novel pixel-aligned Motion Map (MoMap) representation for 3D scene motion, which can be generated from existing generative image models to facilitate efficient and effective motion prediction. To learn meaningful distributions over motion, we create a large-scale database of MoMaps from over 50,000 real videos and train a diffusion model on these representations. Our motion generation not only synthesizes trajectories in 3D but also suggests a new pipeline for 2D video synthesis: first generate a MoMap, then warp an image accordingly and complete the warped point-based renderings. Experimental results demonstrate that our approach generates plausible and semantically consistent 3D scene motion.

1. Introduction

Motion is ubiquitous in the visual world. Understanding, reconstructing, forecasting, and interrogating how objects and entities move in 3D is an important task for computer vision and is critical for applications that involve interaction with physical environments such as augmented reality, autonomous driving, or robotics. Of special interest to us is motion forecasting, prediction, and generation, as required by an agent operating in a 3D environment. Despite its importance, there is a lack of approaches that learn 3D generative motion priors at scale. Existing methods either treat dynamic scene priors as a 2D video generation problem, focus on localized object-centric priors from large-scale object databases, or handle only very short 3D trajectories in constrained, small-scale setups.

In this work, we aim to learn semantically and functionally meaningful 3D motion priors from real-world videos at scale, and to use these priors for semantics-aware 3D motion generation. Recent generative motion methods either consider 2D motions (e.g., TAPIR [10]) or treat 3D trajectories as sets of very short curves, using PointNet or Transformer architectures over them (e.g., GeneralFlow [41], Track2Act [4]). In contrast, we represent motion in a video through a set of pixel-aligned 3D motion snippets we call **Motion Maps**, or **MoMaps** for short (Fig. 2-A). Each MoMap is a motion snapshot—a view of the evolving scene from a fixed camera over a time interval (typically about 50 frames at 3 FPS). The pixels of each MoMap frame encode (in RGB channels) the XYZ 3D locations of points seen by that camera in the fixed reference camera coordinate system, and the entire MoMap records their spatial evolution during the time interval captured by the MoMap.

MoMaps facilitate motion learning in two ways. First, they disentangle camera motion from object motion, reducing the dimensionality of the problem (only dynamic foreground changes over time). Second, their image-like structure allows us to build upon diffusion-based image generative models, such as Stable Diffusion [30, 32], that have been trained on vast amounts of data—effectively repurposing them from the task of dense photometric generation to that of dense position and motion prediction, considering the full context (both foreground and background) of the scene. Such repurposing has been used, for example, in depth prediction by the Marigold method [18]. Additionally, pixel-aligned motion images can directly bind with powerful 2D segmentation masks such as SAM [19, 31], enabling us to exploit semantic and instance information when predicting motion, as motion and semantics are often strongly correlated.

Prior motion prediction methods typically rely on syn-

thetic data (e.g., Kubric [13]) for training, where the motions involved do not capture the richness of the physical world. In contrast, our goal is to learn semantically meaningful real-world motions. Leveraging recent advances in point tracking and 4D reconstruction, our work presents a large dataset of 3D motions extracted from real-world videos and represented as MoMaps. Towards this goal, we developed a full-stack data pipeline based on video depth models [7, 16, 36], 3D point tracking [28, 38], video object segmentation [8, 31] and the MoSca 4D reconstruction system [22] to generate 3D tracks and semantic maps for more than 50K real videos taken from the **HOI4D** [25] (egocentric) and **BRIDGE** [35] (robotics) datasets.

Learning a 3D motion prior from large-scale real videos and building a generative model to predict future 3D trajectories from a single initial frame has proven valuable for various tasks, including robotics [41] and video model prompting [12]. As the first work to explore dense 3D trajectory generation, we showcase another novel downstream application: 2D video frame synthesis. Once a MoMap is generated, we can render the dense 3D trajectories frame by frame into target cameras, resulting in a partial video. A lightweight image diffusion model with flatten-cross-attention can then complete these partially warped 2D frames, directly synthesizing 2D RGB and semantic videos. This pipeline highlights a new direction for video generation that explicitly embeds dynamic 3D bias and 3D motion consistency into the video formation process. Because MoMap retains an image-like structure, most techniques that operate with 2D image diffusion models can be adapted to MoMaps, offering increased flexibility and opening up a broad range of future enhancements for 3D motion generation, for example, more fine-grained motion generation control through vision-language models.

In summary, our key contributions are: **(1)** A general framework for representing 3D motion in dynamic scenes that can be used for encoding real videos as well as generating plausible future motions conditioned by scene view semantics and language. **(2)** An image-like representation of motions, MoMaps, that disentangles camera and object motions and allows the use of large pre-trained image diffusion models for motion prediction. **(3)** A large database of MoMaps derived from open-source datasets. **(4)** An application of MoMaps to a new paradigm for generating future video frames from an image by first generating MoMaps and then completing renderings of warped point clouds.

2. Related Work

Motion Generation in 2D and 3D The closest related work to ours typically focuses on predicting 2D or 3D point trajectories, often within robotics settings [3, 29, 37, 39, 41]: TAPIR [10] uses a U-Net and Fourier encodings of 2D tracks to predict future 2D trajectories from large-scale real videos. GeneralFlow [41] adopts a PointNeXT geometric backbone to process a point cloud from the first video frame, and employs a trajectory-conditioned VAE to predict future 3D trajectories for queried object points. Im2Flow2Act [39] leverages Animatediff [15] to predict 2D tracks and visibilities obtained from TAPIR [10], focusing on object manipulation tasks. Track2Act [3] uses a transformer-based diffusion model over sets of 2D object tracks; it generates these tracks from the first frame and then estimates a rigid SE(3) transformation to manipulate the object. Similarly, ATM [37] learns policies on top of a comparable transformer-based diffusion model for 2D track generation. In a concurrent effort, ARM4R [29] pre-trains a causal transformer to predict future 3D tracks (at a coarser resolution in an egocentric coordinate system) on EPIC-Kitchens, later fine-tuning on robot joint trajectories for policy learning. Zhang et al. [43] address action-conditioned 3D future prediction for specific non-rigid objects reconstructed from visual data. In contrast to these works, our model focuses on long-term, dense, and camera-disentangled 3D motion generation. It also exploits priors from pre-trained image diffusion models, extending beyond the robotics domain and offering a more general-purpose solution.

Video Generation The field of video generation has advanced rapidly with models such as SORA [26], Veo [33], CogVideo [40], and COSMOS [1], etc., which primarily capture priors over pixel sequences across multiple frames. Their approaches often rely on diffusion-based methods applied to stacks of frames [6, 11, 23, 27] or autoregressive models that predict future frames [9, 21, 24]. Some works [20, 34] first generate future 2D video frames and then extract 3D motion for downstream tasks such as object manipulation. However, in all these methods, motion priors are implicitly learned through pixel-level color changes rather than explicit 3D trajectories. By contrast, our approach learns a prior directly on dense 3D tracks, which reside on a simpler and smoother manifold compared to pixel-space motion representations. Additionally, recent works [5, 12, 14] demonstrate that 3D tracks reconstructed from reference videos or obtained via user prompts can improve controllability in video generation. Yet these methods still rely on reconstructing 3D motion from existing sequences, whereas we focus on generating new 3D motions outright.

4D Reconstruction Reconstructing real-world, semantically meaningful 3D motion from video is a key step in learning 3D motion priors. Modern vision foundation models such as video depth estimators [7, 16, 36], 2D/3D point trackers [10, 17, 28, 38], and the induced 4D reconstruction systems [2, 22, 36, 42] play a crucial role in this pro-

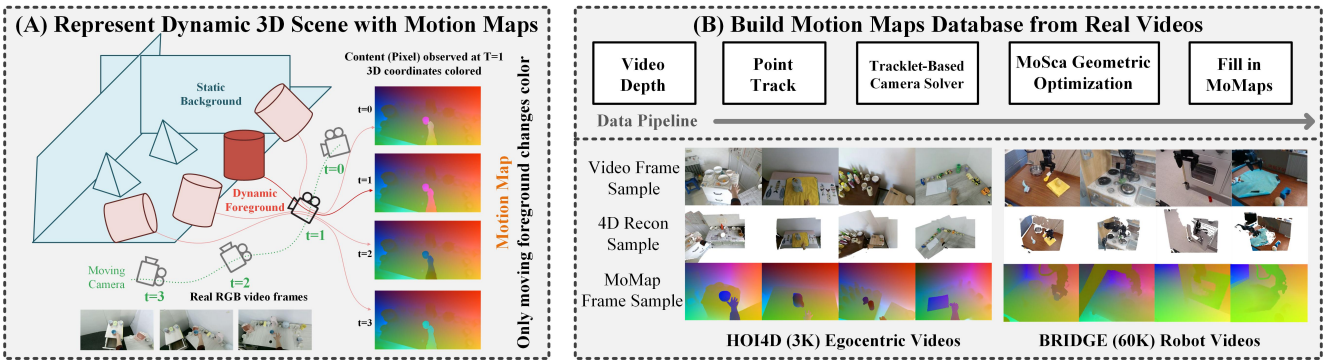


Figure 2. **Motion Maps**: (A) Dynamic 3D scenes can be represented as one or more Motion Maps – curve/trajectory images. (B) We develop a full-stack data pipeline to recover a large dataset of MoMaps from many real videos.

cess. They enable camera pose estimation, geometry reconstruction, and the establishment of dense 3D correspondences, which collectively encode the 3D motion present in a scene. These recent advances in 4D reconstruction serve as the foundation for constructing large-scale databases, facilitating the learning of predictive and generative motion priors.

3. Method

3.1. Motion Map Representation

The first challenge is to represent real-world dynamic 3D scenes in a way that neural networks can easily output or predict. It is crucial to capture motion in 3D space rather than in 2D. Most prior methods model motion as 2D tracks—either for reconstruction tasks (e.g., BootsTAPIR) or for generating 2D tracks (e.g., TAPIR [10], ATM [37], and Track2Act [3]). However, 2D tracks require the model to handle complexities like occlusion. Therefore, we aim to represent the dynamic scene directly as 3D trajectories. A prior method, General-Flow [41], attempts to model individual 3D trajectories as a set of polylines, using a PointNet-like structure to build a VAE that generates short-term 3D trajectories. However, this unstructured set representation cannot handle dense predictions for every pixel and has limited network capacity. Inspired by recent work repurposing image diffusion models for depth prediction (e.g., Marigold), we propose a novel scene representation, called Motion Maps, that is pixel-aligned and can be produced by a repurposed image diffusion model. As shown in Fig. 2(A), a Motion Map (MoMap) is an image of 3D trajectories/curves, or intuitively, is a list of XYZ position maps aligned onto the single-view reference image. Given a reference time (e.g., $t = 1$), we store the 3D trajectory of each pixel observed at this reference time in the corresponding pixel location. All 3D locations are defined with respect to the fixed reference frame at this time, which separates the egocentric (camera) motion from scene object motion. Other ways to see it are that MoMap forms a "3D trajectory image" of all pixels anchored at the reference time; or that MoMap is an XYZ recoloring across time of the same reference XYZ point map. The MoMap representation offers several advantages:

- **Pixel-aligned**: MoMap shares the same underlying structure as a standard image, making it straightforward to generate using large 2D models and allowing us to leverage pretrained image priors.
- **Dense**: Unlike a set of sparse trajectories, MoMap is dense and stores information for every pixel at the reference time, thus capturing both background and foreground elements critical for understanding motion.
- **Smooth and Compressible**: Because MoMap stores 3D trajectories, which are often smooth and low-rank, it can be further compressed into compact latent maps.
- **Camera Motion Disentanglement**: By "freezing" the reference frame, MoMap isolates only the meaningful (foreground) motion, removing the camera's egocentric motion.

Note that one MoMap only captures the content of pixels observed at the reference time, a more complete dynamic 3D scene can be represented by the union of multiple MoMaps anchored at different reference frames with camera poses. This paper only studies the generation of one MoMap and leaves the complete multi-MoMap joint generation as future work (Sec. 5).

3.2. MoMap Database from Real Videos

A non-trivial question arises: where do we learn meaningful motion priors? Many previous works rely on synthetic data (e.g., Kubric and PointOdyssey) to train 2D/3D tracking models that reconstruct trajectories from real observations, or they use manually/semi-automatically labeled small datasets (e.g., General-Flow) to generate short 3D trajectories. However, synthetic data often contains random, semantically/functionally meaningless motion that is only suitable for low-level vision tasks such as tracking or flow, while manual or semi-automatic labeling does not scale well to large video datasets. Instead, we leverage recent advances in 4D reconstruction (MoSca [22]) to reconstruct many real-world 4D scenes, and then convert them into our

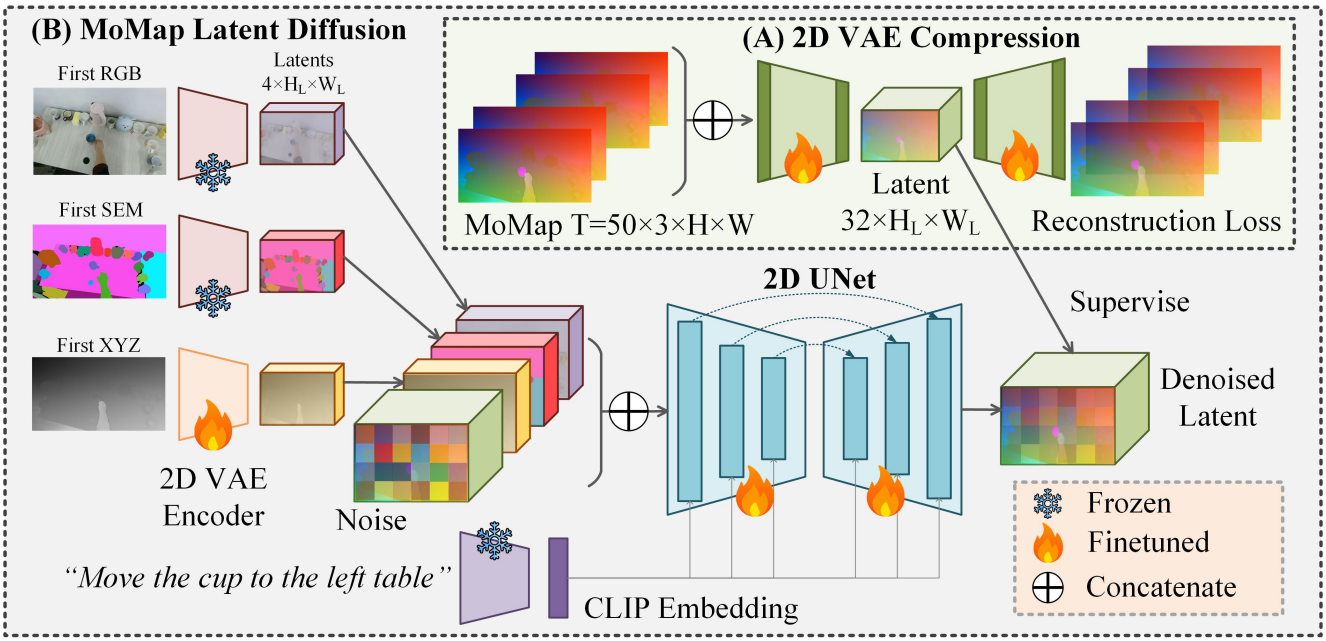


Figure 3. **Method Overview**: (A) A MoMap can be compressed to compact latent via initializing and finetuning a Stable-Diffusion VAE. (B) Given a starting frame and language condition, MoMap can be generated by finetuning the SD UNet.

MoMap representation.

Specifically, as shown in Fig. 2(B), the data preparation stage processes all frames of a raw 2D real video with several steps:

1. We first apply an off-the-shelf video depth model, DepthCrafter, to infer the depth of each frame.
2. We then query a 3D tracker (SpaTracker) densely on every pixel of a target reference time frame (randomly sampled for each video) to build the corresponding MoMap.
3. For videos captured by a moving camera, we run a tracklet-based bundle adjustment to solve for camera poses.
4. Because no current 3D tracker can reliably predict 3D locations under occlusions, we must run an optimization procedure to address occluded intervals in each 3D trajectory. We adapt the geometric optimization stage from MoSca [22] to optimize the dense foreground 3D trajectories.
5. Finally, we fill in the MoMaps at the reference time using these optimized dense 3D trajectories.
6. In addition to this geometric reconstruction process, we also apply a VOS model (DEVA [8]) for dense instance-level video object segmentation, allowing us to incorporate semantic information into the later 3D motion generation.

We applied this data processing pipeline to **3K HOI4D [25]** videos (human-object, egocentric camera) and **60K BRIDGE [35]** videos (robotics, static camera) as shown in Fig. 2(B).

3.3. MoMap Generation

3.3.1. Task Definition

Given a large database of MoMaps, we aim to learn a prior for 3D motions. As shown in in Fig. 1(A), the task is as follows: *given a single RGB image (the first frame) and optionally a text prompt, the model should generate the 3D dense motion of every pixel in that frame, from the input time step into the future time steps.*

We also aim to leverage semantic information from instance segmentation, since points belonging to the same instance typically share a similar motion pattern. Additionally, we assume access to a reliable monocular depth model so that the system does not need to learn 2D-to-3D lifting from scratch. Consequently, the first-frame RGB image can be augmented with an instance segmentation map and an XYZ map derived from the monocular depth model.

3.3.2. MoMap Compression

Thanks to the pixel-aligned property of the MoMap representation, our approach is to exploit existing 2D image diffusion models (e.g., StableDiffusion) to generate MoMaps. However, these image models generally operate on inputs and outputs of shape $H \times W \times 3$, while MoMaps have shape $H \times W \times T \times 3$, where T is the number of frames. Notably, T is typically large (e.g., 50 or 60) in our setting, as we want to generate semantically and functionally meaningful long-term 3D motions. The key question, therefore, is how to adapt these powerful 2D diffusion models to generate MoMaps. We found that it is necessary to encode the $H \times W \times 3$ MoMaps into compact latent feature maps of shape $H_L \times W_L \times C_L$. The underlying insight is that

real-world motion is often smooth and low-rank, making it compressible in both space and time. Concretely, we set $H_L = H/8$ and $W_L = W/8$ for the spatial dimensions, and compress the temporal dimension $T = 50$ (times the 3 spatial channels) into $C_L = 32$ channels. As illustrated in Fig. 3, we increase the input, output, and latent dimensions of the 2D VAE. By carefully initializing the network at the input, intermediate, and output layers, we can effectively transfer weights from a pre-trained VAE, and then finetune with MoMap data. This design allows the architecture to function similarly to the standard 3-channel RGB (as averaging across time of XYZs), leveraging the pre-trained knowledge acquired from large-scale image datasets.

3.3.3. MoMap Diffusion

We exploit the pixel-aligned nature of MoMap by repurposing a large pre-trained image generation model (e.g., Stable Diffusion) to generate these "trajectory images". Once the MoMap compression networks are trained, we can convert the ground-truth MoMap into a compact latent representation and use the U-Net to perform generation. Concretely, as shown in Fig. 3(B), we modify the input and output layers of a pre-trained U-Net from Stable Diffusion while initializing the remaining layers with weights from SD. The input conditions—the first-frame RGB image, the instance segmentation image (encoded as random RGB colors per patch), and the first-frame XYZ map—are all encoded by 2D VAE encoders into low-resolution latents, which are concatenated with noise and fed into the U-Net. Language prompts are incorporated in exactly the same manner as in standard Stable Diffusion. Note that such a lightweight (equivalent to one single image generation) diffusion process can efficiently generate long ($T = 50$) frames in a single sampling process, exploiting the pixel aligned nature of the MoMap representation.

3.4. Application

3.4.1. Video Synthesis

MoMap captures the future 3D motion of every pixel, which is inherently related to predicting future 2D video frames (commonly approached as a video generation problem). We further explore how to transform the 3D MoMap into a complete 2D video, as illustrated in Fig. 4(A). Given a generated MoMap and target camera intrinsics and extrinsics, we render the MoMap using 3D Gaussian Splatting to obtain a partial 2D RGB and semantic video (top row). This rendering is partial because a single MoMap contains only the pixels visible in the first frame, resulting in holes (shown in black). We then finetune another Stable Diffusion single-image model to complete these partial videos using a simple cross-view flattening attention module (similar to CAT3D), ultimately reconstructing the full video (bottom row).

At a high level, this approach explicitly embeds a dynamic 3D inductive bias into video generation. Unlike most video models, which focus on memorizing pixel-level color changes and tend to be computationally heavy, we directly learn how objects move in 3D and form the images using a dedicated renderer (Gaussian Splatting). A separate diffusion model subsequently corrects artifacts and fills in missing regions. As a result, the video generation is handled by two relatively lightweight image models.

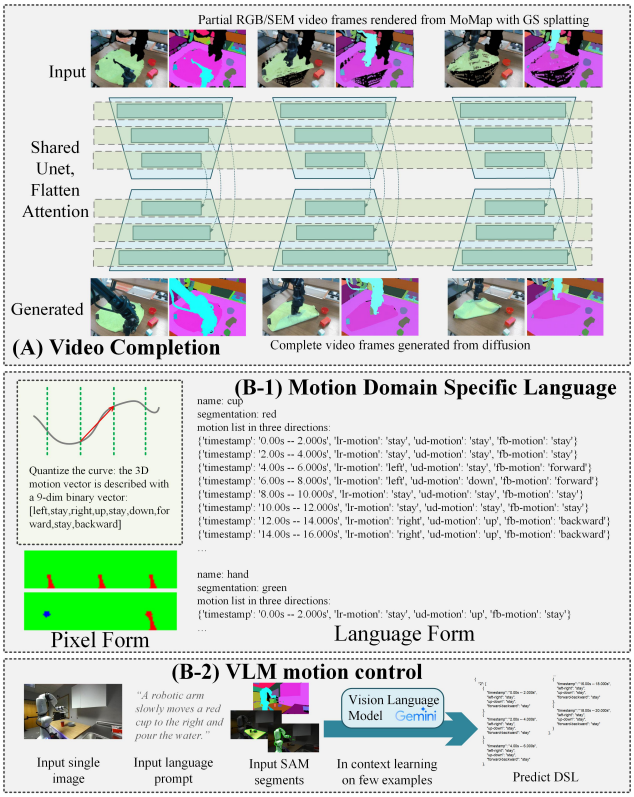


Figure 4. **Application**: (A) 2D video generation via render MoMap and then complete; (B-1) motion DSL representation. (B-2) Infer motion DSL with VLM for finer generation control.

3.4.2. VLM Control

Real-world motion often spans long time periods, making effective conditioning challenging. Relying solely on global text conditions (as in Stable Diffusion) may be insufficient and would require highly accurate text annotations in the dataset (Fig. 6-B). Consequently, there is a need for alternative or augmented conditioning strategies to achieve robust and contextually aligned control over motion generation. We propose leveraging Vision-Language Models (VLMs), which possess strong semantic understanding and commonsense reasoning. To connect high-level VLM outputs with low-level MoMap diffusion, we introduce a domain-specific language (DSL), illustrated in Fig. 4-(B-1). The centroid of each semantic patch is quantized into nine directional flags (e.g., "left", "right", "stay") and converted into a structured language format (e.g., JSON) for each patch. Alternatively, these patch-level directional flags can be grounded

at the pixel level, forming 2D conditions. Thus, this DSL seamlessly combines both a structured language format and pixel-grounded information.

Rather than inserting text prompts directly into the MoMap diffusion network, we remove the text condition and instead rely on pixel-based DSL (2D pixel conditions) to guide generation. During inference, we let a VLM (e.g., Gemini) generate the DSL from the input's first frame as well as the global text prompt via in-context learning, using a few ground-truth examples from the training videos (Fig. 4-B-2). In this setup, VLM's strong reasoning and language capabilities allow it to interpret flexible prompts—even those substantially different from the training dataset—and determine what moves and how. The DSL it produces is then converted into pixel-form conditions for the diffusion model's U-Net (trained without global text conditioning) as an input condition, enabling fine-grained, semantically informed control over 3D motion generation.

4. Experiment

4.1. Baselines Comparison

We compare our 3D motion generation approach with several baselines that also generate 3D trajectories. The closest related work is GeneralFlow [41], which produces short 3D trajectories (typically $T = 4$ frames) from an input-colored point cloud. We train three variants of the G-Flow-large model on our BRIDGE robot dataset:

- **GFlow-Original**: Uses the same configuration as the original GeneralFlow [41] but extends the trajectory prediction length from 4 to 50.
- **GFlow-Rollout**: Keeps the original configuration without changing the prediction length. To achieve T=50, it performs iterative "roll-outs" by taking the last generated frame as the new first-frame input for each subsequent generation round.
- **GFlow-Extend**: An upgraded version of GeneralFlow that increases the querying chunk size from 128 to 20,000 and the scene point cloud samples from 4,096 to 10,000. The network also takes semantic colors as additional point features. Furthermore, the trajectory VAE latent dimension is increased from 16 to 32, and more layers are added to boost overall capacity.

We also built another baseline, **MoMap-VAE**, which uses the same overall architecture but omits the diffusion U-Net in the middle. Instead, it is trained as a conditional VAE that directly generates the MoMap.

Generating a long motion (T=50) from a single starting frame is highly ill-posed, so the most direct and arguably maybe the most accurate way to assess performance is via side-by-side qualitative comparisons of the generated 3D motions as in Fig. 5. From the comparisons, we observe: **GFlow-Original**: Tends to produce diverging trajectories in long-term predictions. Objects often lose their original rigid shape at later time steps. **GFlow-Rollout**: Performs poorly in this setting because it can only generate very short time spans (T=4) and must roll out repeatedly to reach T=50. This causes even more severe divergence over time. **GFlow-Extend**: Benefiting from a larger query chunk size and more network capacity, it generates smoother and more consistent motion. However, due to limited semantic and contextual understanding of Point-NeXT, it misidentifies which objects should move. For instance, in the second-to-last row, it moves two blocks when only one should be moving. **MoMap-VAE** vs. **Ours** (MoMap Diffusion): While MoMap-VAE can produce reasonable long-term motions, our diffusion-based approach generates more natural-looking hand and object motions, reflecting a richer learned prior over realistic 3D trajectories.

4.2. Metrics and Quantitative Comparison

Evaluating the generated results quantitatively is challenging because the problem is highly ill-posed and many plausible generations can be consistent with the provided initial conditions. For each starting frame, there is only one ground-truth observed trajectory, but many plausible solutions exist. We follow the previous work (GeneralFlow [41]) which generates N = 10 samples per input, but computing the metric by comparing the **closest** of the generated samples with the ground truth. Beside detailed reconstruction erros, we introduce several additional coarse metrics that capture various aspects of the motion:

1. Foreground Mask IoU (**fg_mask_iou** ↑): Measures whether the generated motion correctly identifies the moving regions. A "moving mas" is constructed by identifying pixels with significant 3D trajectory displacement. The Intersection over Union (IoU) between the ground truth and generated moving masks is then computed.
2. Foreground Absolute Trajectory Error under Dynamic Time Warping (**ate_dtw**↓): Computes ATE (reported in GeneralFlow) after optimizing temporal alignment with Dynamic-Time-Warping (DTW), capturing scenarios where the timing of motions differs but the overall trajectory is correct.
3. Foreground Curve Distance-Matrix Signature Error (**D_sig**↓): Creates a translation and rotation-invariant signature by computing a T x T distance matrix between positions at any two-time steps. The difference between ground truth and generated matrices measures SE(3) invariant curve similarity.
4. Foreground Local Distance Difference (**local_dist_diff**↓): Evaluates the preservation of local structure by computing distances between K-nearest neighbors in the ground truth and generated 3D trajectories. Measures whether the generated motion preserves local rigidity and avoids collapsing into noise.

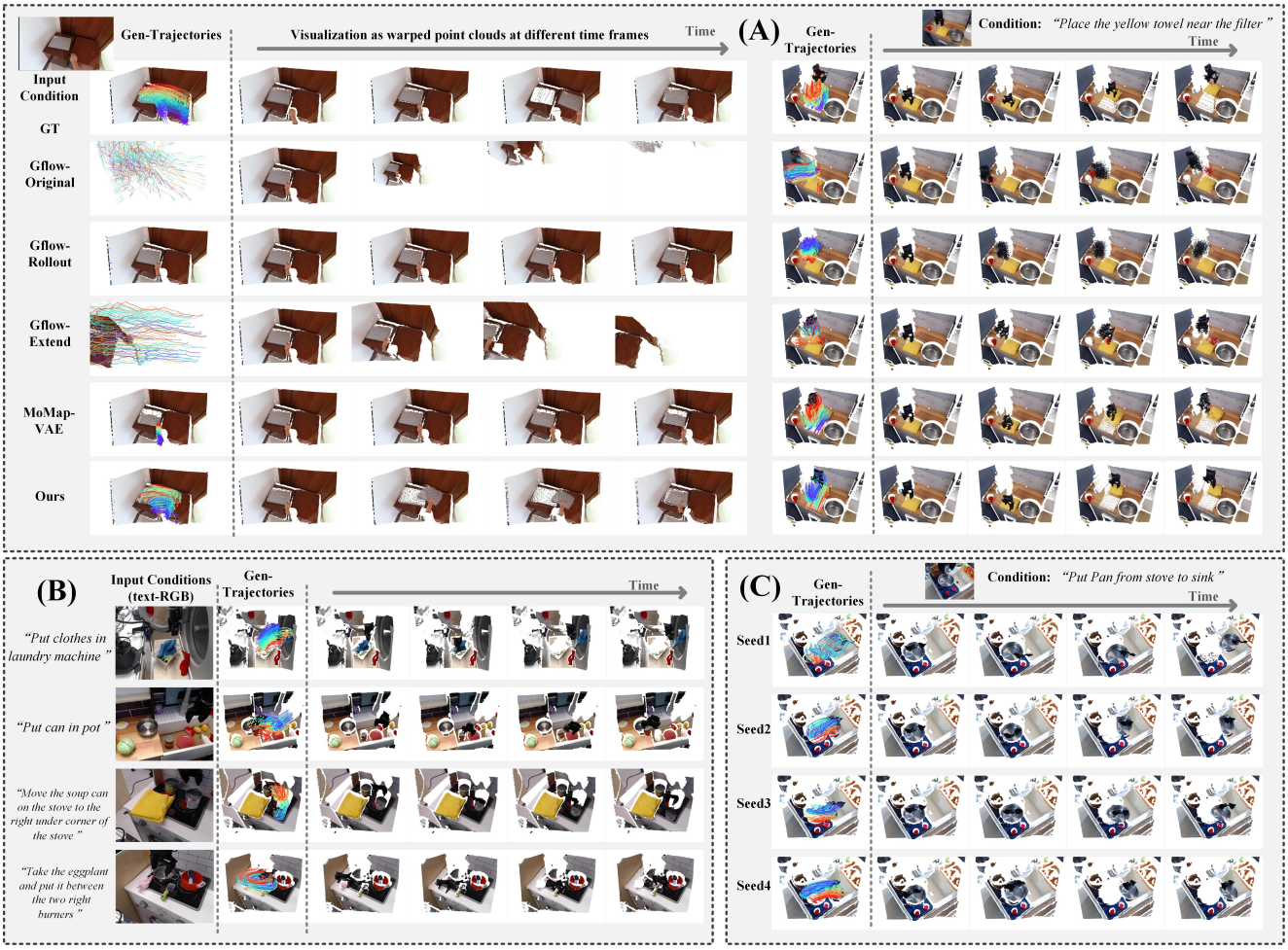


Figure 5. **Qualitative Results**: (A) comparison with baselines. (B) More generation results. (C) Diverse generations from the same condition with different random seeds.

5. Foreground Patches Nearest Patch Accuracy (**cross_patch_nearest_acc**↓) tracks which patch is closest (centroid-wise) to each moving foreground patch at each time step. Accuracy measures whether the generated motion maintains the same nearest-patch relationships as the ground truth.
6. Foreground dT Quantized Direction Accuracy (**quantize_acc_dT**↑): Coarsely evaluates motion direction by quantizing a 3D trajectory over a delta time interval (dT) into 9 directions (left-stay-right, forward-stay-backward, up-stay-down). Measures the accuracy between ground truth and generated motion directions for each pixel in the foreground mask.

These metrics capture a variety of aspects, ranging from fine-grained reconstruction errors to coarse object-level motion directions and proximities, enabling a more comprehensive evaluation of 3D trajectories. The metrics of Ours and the baselines on a small testset are reported in Tab. 1.

We observe that our method outperforms the baselines in most metrics, especially in quantized directional accuracy and the D_sig_fg↓ metric. From both qualitative and quantitative findings, we note that our generated trajectories exhibit stronger semantic coherence and greater internal consistency, particularly in capturing aggregate motion. In contrast, GFlow appears to have lower noise in individual point trajectories (reflected by higher reconstruction metrics) but exhibits inconsistent noise across adjacent points belonging to the same object, leading to visually detectable "object breakdowns."

4.3. Application

Video synthesis. We present our video synthesis results in Fig.6-A, following the approach in Sec.3.4.1, showing that our lightweight model achieves decent performance.

VLM control. In Fig.6-B, we showcase VLM-controlled generation, described in Sec.3.4.2. Notably, the VLM-controlled approach produces more accurate motion than relying solely on a global text condition, which can fail in certain scenarios.

Dataset	BRIDGE					HOI4D				
Metric\Method	**GFlow-Ori.**	**GFlow-Roll.**	**GFlow-Ext.**	**MoMap VAE**	**Ours**	**GFlow-Ori.**	**GFlow-Roll.**	**GFlow-Ext.**	**MoMap VAE**	**Ours**
fg_mask_iou↑	0.8044	0.5659	0.7597	**0.8271**	0.8128	0.0667	0.0000	0.0667	0.3400	**0.4492**
ate_dtw↓	0.0747	0.1164	0.0812	0.0751	**0.0689**	1.6119	0.1241	0.7281	0.1265	**0.1112**
D_sig↓	0.0536	0.0873	0.0591	0.0538	**0.0463**	1.3888	0.1193	0.2922	0.1030	**0.0886**
local_dist_diff↓	0.0124	0.0154	0.0073	0.0073	**0.0058**	0.0154	**0.0086**	0.0092	0.0094	0.0092
patch_nearest_acc↑	0.8387	0.7649	0.8244	0.8398	**0.8691**	0.6926	0.7631	0.7554	0.7495	**0.7923**
quantize_acc_1↑	0.6758	0.6271	0.6623	0.6856	**0.7212**	0.2116	0.6014	0.3279	0.6015	**0.6157**
quantize_acc_4↑	0.5890	0.4702	0.5568	0.6148	**0.6554**	0.3653	0.2710	0.3656	0.3000	**0.4278**
quantize_acc_16↑	0.7294	0.4503	0.7049	0.7293	**0.7752**	0.4319	0.1347	0.4068	0.3980	**0.4610**

Table 1. Quantitative comparison on BRIDGE and HOI4D testset.

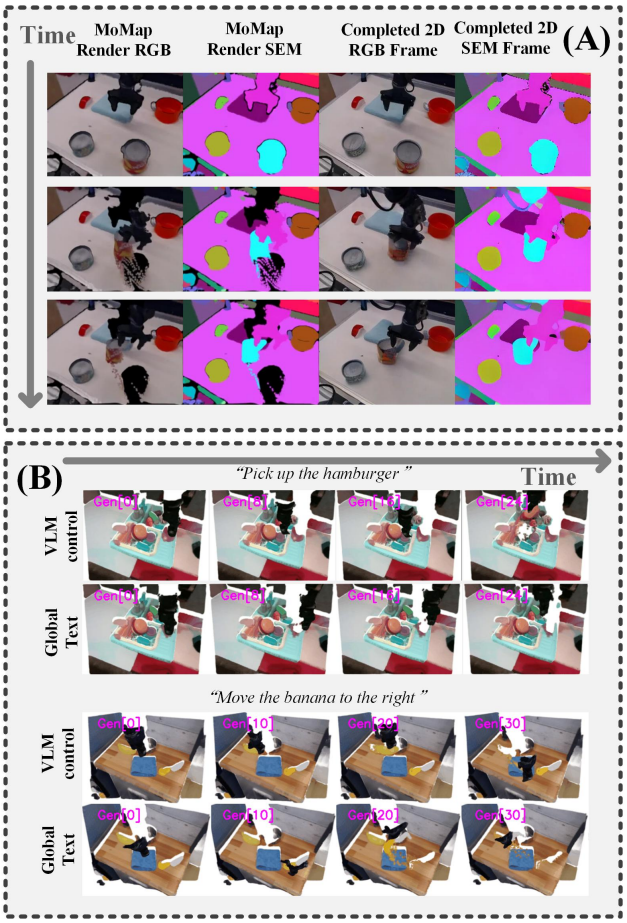


Figure 6. **Application Results**: (A) 2D video generation results. (B) VLM finer control can generate reasonable results even when global text condition fails.

4.4. Ablation Study

We validate the design of our MoMap diffusion by ablating various components in its architecture: (1) remove the input XYZ condition; (2) remove the input semantic image condition; (3) remove the VAE compression of the MoMaps and let the UNet work on the original MoMap space instead of the latent space; (4) train the diffusion from scratch without initialization from stable-diffusion pre-trained weights; and (5) train both the MoMap VAE and UNet from scratch. Tab. 2 presents the results of these ablation experiments, illustrating that our full model achieves the best performance.

Metric\Ablation	Full	no xyz	no sem	no vae	no unet init	no unet&vae init
fg_mask_iou↑	**0.8128**	0.7962	0.8079	0.1305	0.7974	0.7450
ate_dtw↓	**0.0689**	0.0738	**0.0689**	0.1759	0.0792	0.0992
D_sig↓	**0.0463**	0.0484	0.0468	0.1082	0.0520	0.0579
local_dist_diff↓	0.0058	0.0060	0.0056	0.1348	**0.0055**	0.0094
patch_nearest_acc↑	**0.8691**	0.8605	0.8659	0.7533	0.8510	0.8306
quantize_acc_1↑	**0.7212**	0.7142	0.7187	0.1940	0.7085	0.6566
quantize_acc_4↑	**0.6554**	0.6415	0.6522	0.3977	0.6260	0.5925
quantize_acc_16↑	**0.7752**	0.7589	0.7719	0.5311	0.7434	0.7267

Table 2. Ablation comparison on BRIDGE dataset.

5. Limitations and Conclusion

In this paper, we explored a new problem: learning to generate future 3D motion for an entire scene using large-scale real-world videos. We introduced an image-like *MoMap* representation, which enables the repurposing of established 2D image diffusion models for 3D motion generation. By assembling a large database of MoMaps from real videos, we demonstrated the feasibility of synthesizing semantically and functionally meaningful 3D scene motion, also highlighting its potential impact on 2D video synthesis.

Despite these promising results, our work represents an initial step in the realm of dense 3D motion generation at scale, and several challenges and directions remain:

1. **Multiview MoMap joint generation.** This paper primarily focuses on generating a single MoMap anchored at the first frame, which only partially captures the scene. A key future direction lies in jointly and consistently generating multiple MoMaps anchored at different frames or viewpoints to aggregate a more complete scene.
2. **Enhanced motion control and VLMs.** Currently, the language prompts are integrated similarly to Stable Diffusion, providing limited motion control. Achieving more fine-grained and semantically meaningful motion control using advanced vision-language models is a promising area of future research.
3. **Scaling to larger and more diverse data.** Our current datasets, though sizable, are domain-specific (e.g., hand-object or robotics). Extending these techniques to large-scale, general-purpose videos presents an important avenue for broader impact.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 2

[2] Abhishek Badki, Hang Su, Bowen Wen, and Orazio Gallo. L4P: Low-level 4D vision perception unified. 2025. 2

[3] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024. 2, 3

[4] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[5] Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. Gs-dit: Advancing video generation with pseudo 4d gaussian fields through efficient dense 3d point tracking. *arXiv preprint arXiv:2501.02690*, 2025. 2

[6] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[7] Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv:2501.12375*, 2025. 2

[8] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023. 2, 4

[9] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024. 2

[10] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 1, 2, 3

[11] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023. 2

[12] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. *arXiv preprint arXiv:2412.02700*, 2024. 2

[13] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 2

[14] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. *arXiv preprint arXiv:2501.03847*, 2025. 2

[15] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023. 2

[16] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. *arXiv preprint arXiv:2409.02095*, 2024. 2

[17] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker: It is better to track together. 2023. 2

[18] 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 (CVPR)*, 2024. 1

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[20] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. *arXiv preprint arXiv:2310.08576*, 2023. 2

[21] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 2

[22] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic gaussian fusion from casual videos via 4D motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2, 3, 4

[23] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. 2

[24] Chang Liu, Rui Li, Kaidong Zhang, Yunwei Lan, and Dong Liu. Stablev2v: Stablizing shape consistency in video-to-video editing. *arXiv preprint arXiv:2411.11045*, 2024. 2

[25] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2, 4

[26] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 2

[27] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 2

[28] Tuan Duc Ngo, Peiye Zhuang, Chuang Gan, Evangelos Kalogerakis, Sergey Tulyakov, Hsin-Ying Lee, and Chaoyang Wang. Delta: Dense efficient long-range 3d tracking for any video. *arXiv preprint arXiv:2410.24211*, 2024. 2

[29] Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. *arXiv preprint arXiv:2502.13142*, 2025. 2

[30] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[31] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2

[32] 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. 1

[33] Abhishek Sharma, Adams Yu, Ali Razavi, Andeep Toor, Andrew Pierson, Ankush Gupta, Austin Waters, Aäron van den Oord, Daniel Tanis, Dumitru Erhan, Eric Lau, Eleni Shaw, Gabe Barth-Maron, Greg Shaw, Han Zhang, Henna Nandwani, Hernan Moraldo, Hyunjik Kim, Irina Blok, Jakob Bauer, Jeff Donahue, Junyoung Chung, Kory Mathewson, Kurtis David, Lasse Espeholt, Marc van Zee, Matt McGill, Medhini Narasimhan, Miaosen Wang, Mikołaj Bińkowski, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Nando de Freitas, Nick Pezzotti, Pieter-Jan Kindermans, Poorva Rane, Rachel Hornung, Robert Riachi, Ruben Villegas, Rui Qian, Sander Dieleman, Serena Zhang, Serkan Cabi, Shixin Luo, Shlomi Fruchter, Signe Nørly, Srivatsan Srinivasan, Tobias Pfaff, Tom Hume, Vikas Verma, Weizhe Hua, William Zhu, Xinchen Yan, Xinyu Wang, Yelin Kim, Yuqing Du, and Yutian Chen. Veo. 2024. 2

[34] Weiliang Tang, Jia-Hui Pan, Wei Zhan, Jianshu Zhou, Huaxiu Yao, Yun-Hui Liu, Masayoshi Tomizuka, Mingyu Ding, and Chi-Wing Fu. Embodiment-agnostic action planning via object-part scene flow. *arXiv preprint arXiv:2409.10032*, 2024. 2

[35] 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. 2, 4

[36] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 2

[37] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 2, 3

[38] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 2

[39] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 2

[40] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[41] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 1, 2, 3, 6

[42] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024. 2

[43] Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. *arXiv preprint arXiv:2410.18912*, 2024. 2