DYNAREP: LEARNING FUTURE VISUAL REPRESENTATIONS FOR OBJECT-CENTRIC DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most embodied world models predict futures in latent spaces optimized for pixel reconstruction, which may not be ideal for capturing manipulation-relevant dynamics. We present DynaRep, a world model that represents future world states in a visual representation space rich in semantic and geometric structure. To this end, we jointly model future DINO representations and 3D object flow to capture object-centric dynamics. To make dense DINO features tractable for generative modeling, we introduce DINO-VAE, which performs channel and temporal compression with causal temporal modeling to produce compact latents while preserving informative visual structure. Building on this compact representation space, we employ a unified diffusion transformer with joint attention to capture the bidirectional dependencies between future visual representations and 3D object flow. To further strengthen cross-modal coupling, DynaRep adopts asymmetric modality-specific noise schedules, allowing visual representations to stabilize earlier during denoising and continuously guide 3D object flow generation. Experiments on simulated and real-world manipulation data show that DINO-derived representations provide a more effective space for future modeling than pixel-reconstruction-oriented latents, leading to more geometrically consistent and task-relevant 3D object flow. These results demonstrate the value of predictive visual representations for modeling object-centric dynamics in embodied interaction.

1 INTRODUCTION

Effective robotic manipulation requires reasoning beyond the current observation to anticipate future object dynamics under interaction. 3D object flow explicitly represents the trajectories of object points in 3D space, providing a natural description of future physical dynamics (Xiao et al., 2024). It can also serve as a relatively embodiment-agnostic, high-level action representation (Wen et al., 2024), making it a promising interface between world prediction and robotic action.

However, existing flow-based world modeling still faces three key challenges. First, the representation space for future-state modeling is not fully aligned with physical dynamics prediction. Existing approaches typically inherit the World-Action Model (WAM) paradigm and model future scenes in pixel-reconstruction-oriented VAE latent spaces (Li et al., 2025; Zhang et al., 2026), which emphasize texture and low-level appearance rather than object semantics, task state, and spatial geometry that are more relevant to motion prediction (Yu et al., 2025; Yao et al., 2025). Second, pretrained visual features are difficult to model generatively. Although DINO provides rich visual features (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025), their high dimensionality and redundancy limit their direct use as compact, well-regularized latents for efficient diffusion modeling (Zheng et al., 2026; Bi et al., 2026). Third, existing joint generation frameworks do not fully exploit future world states to guide motion generation. Predicted visual representations can capture the semantic and geometric evolution of future scenes, yet existing modeling paradigms struggle to effectively translate such information into constraints and guidance for object motion (Lee et al., 2026), thereby limiting the conditional coupling between future visual states and object dynamics. Existing representation-based world models mainly focus on predicting future representations themselves (Zhou et al., 2025; Assran et al., 2025; Boduljak et al., 2025), while jointly modeling future visual representations and 3D object flow remains underexplored.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

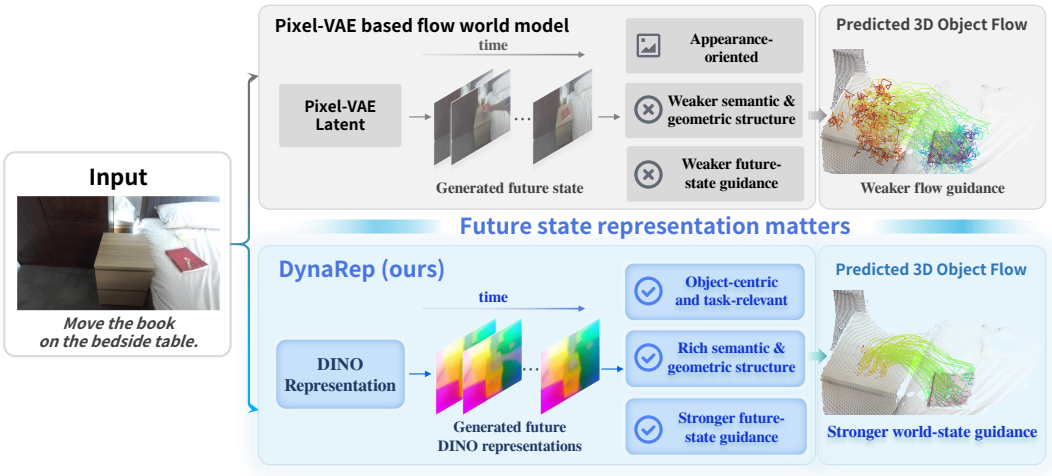

Figure 1: **Future-state representation matters for 3D object flow generation.** Compared with pixel-VAE-based flow world models, DynaRep models future states in the DINO representation space, which preserves richer semantic and geometric structure for dynamics modeling.

To address these challenges, we introduce DynaRep, a visual representation world model for object-centric dynamics that jointly models future DINO features and 3D object flow within a unified generative framework (Fig. 1). DynaRep first uses a DINO encoder to map visual observations into a representation space with rich semantic and geometric structure. To make high-dimensional DINO features more suitable for generation, we further introduce DINO-VAE, which combines temporal compression, channel compression, and latent regularization to produce a compact generation-friendly representation space. DynaRep then jointly generates future visual representations and 3D object flow using a unified diffusion transformer, enabling direct interaction between visual-state evolution and object dynamics. Finally, we introduce decoupled timestep sampling and modality-specific noise schedules, so that future visual representations can emerge earlier along the denoising trajectory and continuously guide flow generation in a latent-forcing manner.

This design provides three main advantages. First, DynaRep shifts future-state modeling from pixel-reconstruction-oriented latent spaces to a more task-relevant visual representation space, allowing the model to focus more on object semantics, spatial structure, and state changes rather than unnecessary appearance details, thereby providing a more informative world-state prior for flow. Second, DINO-VAE substantially compresses the temporal and channel dimensions, yielding a compact and generation-friendly latent space that enables more efficient generative world modeling while preserving key semantic and geometric information. Third, the asymmetric multimodal denoising process alleviates the limitation of conventional joint generation, where both modalities start from highly corrupted states, allowing progressively refined future visual states to continuously guide object motion generation. Meanwhile, the decoupled formulation allows flow to condition future scene evolution, enabling joint generation, inverse dynamics, and flow-conditioned forward dynamics within a single model. Overall, DynaRep tightly couples predictive visual states with object-centric dynamics in a unified generative framework.

Our main contributions are summarized as follows. First, we introduce DynaRep, a visual representation world model that jointly models future visual representations and 3D object flow within a unified generative framework, coupling future scene states with object-centric dynamics. Second, we propose DINO-VAE, which compresses high-dimensional DINO features through temporal and channel compression with latent regularization, yielding a compact and generation-friendly latent space while preserving key visual information. Third, we develop a modality-specific noise scheduling strategy with decoupled timesteps and asymmetric denoising trajectories, enabling latent-forcing-style visual guidance and bidirectional conditional modeling between visual states and flow. Finally, extensive experiments across simulated and real-world manipulation scenarios demonstrate that DynaRep achieves state-of-the-art 3D object flow generation performance, validating the effectiveness of representation space world modeling for object-centric dynamics.

2 Related Work

World-Action Models and Flow-based World Models. VLA models map multimodal observations directly to actions (Kim et al., 2024; Black et al., 2024; Brohan et al., 2023), leaving environment dynamics implicit. WAMs make such dynamics explicit by predicting future observations and acting on them (Du et al., 2023; Bruce et al., 2024; Yang et al., 2024), a paradigm recently scaled to open-domain manipulation by large video world models (NVIDIA, 2026; Li et al., 2026b). Point trajectories represent dynamics with limited dependence on visual appearance. Modern trackers estimate dense 2D or 3D point correspondences from video (Karaev et al., 2024; Lu et al., 2026; Karhade et al., 2025). Predicted trajectories have also been used as an embodiment-agnostic interface for action, serving as policy pre-training targets (Wen et al., 2024), interaction plans (Bharadhwaj et al., 2024), and representations of 3D affordances learned from human video (Yuan et al., 2024). Existing methods typically follow one of two approaches. Some generate future RGB video and recover trajectories with an off-the-shelf tracker (Li et al., 2025), allowing video-generation and tracking errors to compound. Others predict 3D trajectories directly without generating pixels (Lee et al., 2026; 2025; Zhang et al., 2026; Li et al., 2026a), but do not explicitly model future visual states. DynaRep instead jointly generates future visual representations and 3D object flow, enabling direct interaction between the two during prediction.

Visual Representations for Generative World Modeling. Most video world models operate in VAE latent spaces optimized for pixel reconstruction (Rombach et al., 2022), which devote substantial capacity to low-level appearance details that may be less relevant to downstream dynamics modeling. Self-supervised visual foundation models instead provide patch-level features with strong object semantics and cross-frame correspondence (Siméoni et al., 2025), and world models built on frozen visual features can support planning without reconstructing pixels (Zhou et al., 2025; Assran et al., 2025). However, these methods primarily predict future representations themselves, leaving how such predictive visual states can support 3D motion generation underexplored. A parallel line studies visual representations as generative spaces: aligning diffusion transformers with pretrained features improves optimization (Yu et al., 2025; Yao et al., 2025), while representation autoencoders show that visual representations can serve as effective latent spaces for generation (Zheng et al., 2026; Singh et al., 2026; Guo et al., 2026). We build on this direction to make dense DINO features tractable as a predictive state space for joint visual representation and 3D object flow modeling.

3 Method

3.1 Problem Formulation and Overview

Problem Formulation. Given the initial visual representation Z_0, the corresponding 3D point set P_0 of the target object, and a task description c, DynaRep jointly models future visual representations and 3D object flow within a unified framework. We treat 3D object flow as a high-level action representation. For the p-th point, its object flow at time t is defined as the 3D displacement relative to its initial position $F_{t,p} = P_{t,p} - P_{0,p}$. Let $Z_{1:T}$ denote the future visual representation sequence. DynaRep jointly generates future visual states and 3D object flow as

$$(\hat{Z}_{1:T}, \hat{F}_{1:T}) \sim \pi_\theta \left(Z_{1:T}, F_{1:T} \mid Z_0, P_0, c\right). \tag{1}$$

With the same bidirectional conditional modeling framework, DynaRep also supports inverse dynamics and flow-conditioned forward dynamics:

$$\hat{F}_{1:T} \sim \pi_\theta^{\mathrm{IDM}} \left(F_{1:T} \mid Z_{0:T}, P_0, c\right), \tag{2}$$

$$\hat{Z}_{1:T} \sim \pi_\theta^{\mathrm{FDM}} \left(Z_{1:T} \mid Z_0, F_{1:T}, P_0, c\right). \tag{3}$$

Overview. The overall framework of DynaRep is illustrated in Fig. 2. First, we use a frozen DINO encoder to extract visual features with rich semantic and spatial structure, and employ DINO-VAE to compress the high-dimensional DINO features into compact latent representations suitable for generative modeling (Sec. 3.2). DynaRep then jointly models future visual representations and 3D object flow within a unified generative framework (Sec. 3.3). Finally, we introduce modality-specific timestep sampling and noise schedules to model cross-modal conditional dependencies under different noise levels (Sec. 3.3).

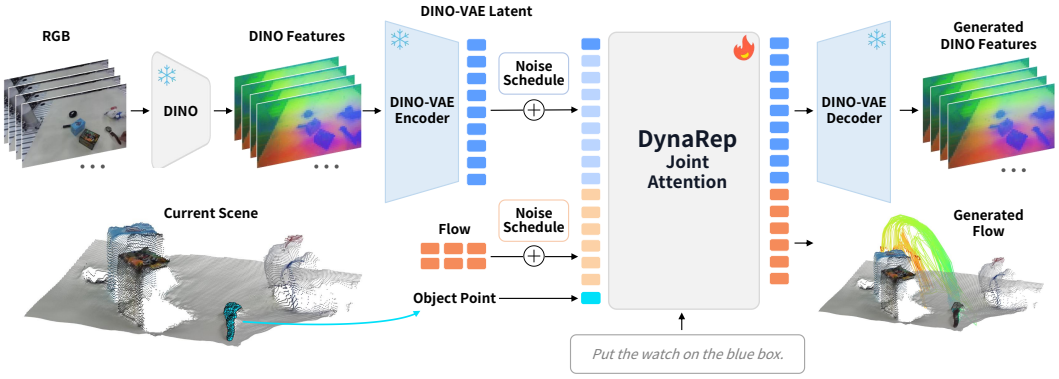

Figure 2: **Training framework of DynaRep.** RGB observations are encoded by DINO and compressed by DINO-VAE into compact visual latents. The visual and 3D object flow streams are independently corrupted with modality-specific noise schedules and jointly modeled by DynaRep, conditioned on the initial object point and language instruction. The model jointly predicts future DINO representations and 3D object flow.

3.2 DINO-VAE: Visual Representation Compression

Visual representation models such as DINO (Siméoni et al., 2025) provide rich object-level semantics and spatial-geometric structure, yet their raw features are not tailored to generative modeling. We therefore introduce DINO-VAE to compress DINO features into a compact latent space while preserving their semantic and geometric structure (Fig. 3).

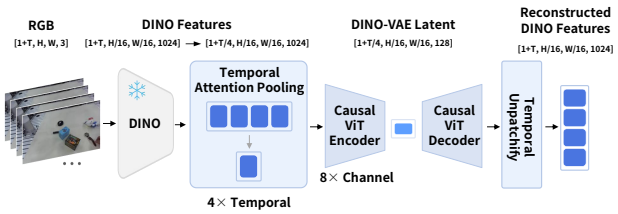

Figure 3: **Overview of the DINO-VAE architecture.**

Temporal and Channel Compression. Given an input video $V \in \mathbb{R}^{(1+T)\times H\times W\times 3}$, DINO extracts patch-level features $X \in \mathbb{R}^{(1+T)\times \frac{H}{16}\times \frac{W}{16}\times 1024}$. Since frame-wise DINO features exhibit substantial temporal redundancy, modeling every frame independently introduces unnecessary computational overhead. We therefore employ an attention-pooling module that aggregates every four consecutive frames into one temporal latent, yielding $X' \in \mathbb{R}^{(1+\frac{T}{4})\times \frac{H}{16}\times \frac{W}{16}\times 1024}$. Direct diffusion modeling in the high-dimensional DINO feature space is challenging. Much of the meaningful visual representation in DINO features lies in a lower-dimensional subspace, whereas corrupting and denoising the full ambient space forces the generative model to account for many directions that contribute little to the dominant representation structure, substantially increasing the optimization difficulty of diffusion-based generation. We therefore process X' with a three-layer causal ViT encoder that jointly performs causal temporal modeling and channel compression, mapping the original 1024-dimensional features into a low-dimensional representation latent $Z \in \mathbb{R}^{(1+\frac{T}{4})\times \frac{H}{16}\times \frac{W}{16}\times d_z}$. We use a lightweight architecture to preserve the representational structure of the original DINO features. The corresponding ViT decoder adopts a symmetric architecture and reconstructs the latent representation back to the original DINO feature space through temporal unpatchification, yielding $\hat{X} \in \mathbb{R}^{(1+T)\times \frac{H}{16}\times \frac{W}{16}\times 1024}$.

Training with Latent Regularization. The raw DINO feature distribution is not explicitly regularized for diffusion-based generation. We therefore adopt a β-VAE formulation to encourage a smoother and more continuous latent distribution. The DINO-VAE objective is

$$\mathcal{L}_{\text{DINO-VAE}} = \mathcal{L}_{\text{rec}} + \lambda_{\cos}\mathcal{L}_{\cos} + \beta\mathcal{L}_{\text{KL}}. \tag{4}$$

Here, $\mathcal{L}_{\text{rec}} = \|\hat{X} - X\|_2^2$ is the DINO feature reconstruction loss; $\mathcal{L}_{\cos}$ is a token-wise cosine consistency loss between the original and reconstructed DINO features that preserves the directional structure of DINO features; and $\mathcal{L}_{\text{KL}} = D_{\text{KL}}(q_\phi(Z \mid X) \,\|\, \mathcal{N}(0, I))$ regularizes the latent distribution. Overall, DINO-VAE compresses the high-dimensional and temporally redundant

DINO feature sequence into a compact visual representation latent space, providing a unified generative space for future representation prediction and subsequent visual–flow joint modeling.

3.3 DINO-FLOW DIT: UNIFIED VISUAL AND FLOW MODELING

Building on the compact visual latents produced by DINO-VAE, DynaRep jointly models future visual representations and object-centric dynamics in a unified generative framework, as shown in Fig. 2. Rather than using pixel-reconstruction-oriented VAE latents to represent future scenes, DynaRep uses DINO-derived visual latents alongside 3D object flow to model scene evolution and object motion.

Flow Representation. We use the definition of 3D object flow from Sec. 3.1. Since flow is represented directly in metric 3D space, its numerical scale can differ substantially from the Gaussian noise used in diffusion-based generative modeling. We therefore normalize flow with a global scale factor s_{flow} as $\tilde{F}_{t,p} = \frac{F_{t,p}}{s_{\text{flow}}}$, yielding a representation whose scale is better suited for diffusion-based generation. To associate each flow token with its corresponding geometric location, we further use the initial 3D coordinate $P_{0,p}$ as a geometric anchor. Specifically, we separately encode the global 3D position of each point in the initial frame and its local geometry relative to the object centroid, and fuse the two as the 3D geometric positional encoding for each flow token.

Cross-Modal Dynamics Modeling. We then concatenate the representation latent sequence produced by DINO-VAE with the flow token sequence, and jointly process them together with the clean initial object points P_0 using a unified diffusion transformer. Joint attention allows visual and flow tokens to interact bidirectionally: predicted future visual states can guide flow generation, while flow provides explicit object-motion information for predicting future scene states. To further capture cross-modal spatiotemporal correspondences, we introduce spatiotemporal RoPE. We assign spatial coordinates to each flow token based on the image-plane location of its initial 3D point $P_{0,p}$ in the DINO feature map, while aligning its temporal coordinate with the corresponding DINO latent timestep, thereby establishing explicit cross-modal spatiotemporal alignment within joint attention. Finally, separate output heads project the visual and flow representations back into their corresponding spaces, yielding the predicted future visual latents and 3D object flow.

Modality-Specific Noise Scheduling. To strengthen conditional modeling between visual representations and 3D object flow, we decouple diffusion timestep sampling and noise scheduling across the two modalities. During training, we independently sample diffusion timesteps τ_R and τ_F for the visual representation and flow streams, respectively. The sampled timesteps are subsequently mapped to modality-specific noise levels using the shifted schedule $S_s(\tau) = \frac{s\tau}{1+(s-1)\tau}$, yielding $\sigma_R = S_{s_R}(\tau_R)$ and $\sigma_F = S_{s_F}(\tau_F)$. We use separate noise schedules for the two modalities to account for their different generation difficulties. In particular, the flow schedule is biased toward higher noise levels, allowing the visual representation stream to form stable structure earlier and providing the basis for the latent-forcing trajectory used at inference. The forward corruption process is defined as

$$x^R_{\sigma_R} = (1-\sigma_R)x^R_0 + \sigma_R\epsilon_R, \qquad \epsilon_R \sim \mathcal{N}(0, I). \tag{5}$$

$$x^F_{\sigma_F} = (1-\sigma_F)x^F_0 + \sigma_F\epsilon_F, \qquad \epsilon_F \sim \mathcal{N}(0, I). \tag{6}$$

Independent timestep sampling exposes the model to varied visual–flow noise combinations, supporting both visual-conditioned flow generation and flow-conditioned visual prediction.

3.4 TRAINING AND INFERENCE

Joint Training Objective. Both the DINO visual representation latent and 3D object flow adopt an x-prediction parameterization. For high-dimensional representation latents, meaningful structure is often concentrated on a lower-dimensional data manifold; directly predicting the clean representation therefore avoids explicitly regressing all noise directions and reduces the optimization burden. The model jointly predicts the clean states as

$$(\hat{x}^R_0, \hat{x}^F_0) = f_\theta\left(x^R_{\sigma_R}, x^F_{\sigma_F}, \sigma_R, \sigma_F, c\right). \tag{7}$$

The corresponding velocity is computed as $\hat{v} = \frac{x_\sigma - \hat{x}_0}{\max(\sigma, \epsilon)}$, where ϵ is a small constant introduced for numerical stability. The training objective combines the velocity-matching losses of the two

Table 1: **3D object flow generation on WorldFlowBench.** We evaluate 3D point trajectories over a 49-frame horizon. ADE and FDE measure trajectory error in meters, while PWT reports the mean point-frame hit rate averaged over thresholds of 1 cm, 2 cm, and 5 cm.

Paradigm	Model	Inputs		Sim7			Real3			SpatialCog		
		Frames	Text	ADE↓	FDE↓	PWT↑	ADE↓	FDE↓	PWT↑	ADE↓	FDE↓	PWT↑
Video generation + 3D tracker	NovaFlow	1	✓	0.1964	0.3186	0.2897	0.0959	0.1290	0.2930	0.1657	0.2951	0.3200
	Cosmos3-Nano + Track4World	1	✓	0.1008	0.2021	0.4510	0.0881	0.1333	0.3095	0.1409	0.2571	0.3978
	Cosmos3-Nano + 4RC	1	✓	0.0963	0.1958	0.5308	0.0847	0.1219	**0.3318**	0.1412	0.2632	**0.4012**
	LingBot-VA + Track4World	1	✓	0.1051	0.2072	0.4247	0.0967	0.1453	0.2670	0.1338	0.2387	0.3644
	LingBot-VA + 4RC	1	✓	0.1035	0.2072	0.4919	0.0924	0.1344	0.2951	0.1311	0.2321	0.3755
Direct flow generation	MolmoMotion-AR (H1)	1	✓	0.1796	0.3297	0.2940	0.1818	0.3308	0.2501	0.2762	0.5331	0.1903
	μ_0	1	✓	0.1868	0.3073	0.0698	0.1164	0.1564	0.1480	0.2132	0.2705	0.0553
	EgoWAM	1	✓	0.1987	0.3365	0.0209	0.1301	0.1901	0.0562	0.2493	0.3838	0.0800
Ours	**DynaRep**	1	✓	**0.0530**	**0.0966**	**0.5561**	**0.0723**	**0.1023**	0.3258	**0.0828**	**0.1260**	0.3861

modalities:

$$\mathcal{L} = \lambda_R \mathcal{L}_R + \lambda_F \mathcal{L}_F, \tag{8}$$

where

$$\mathcal{L}_R = \|\hat{v}^R - v^R\|_2^2, \mathcal{L}_F = \|\hat{v}^F - v^F\|_2^2. \tag{9}$$

This design retains direct x-prediction while remaining consistent with the Rectified flow trajectory through velocity matching. We first pretrain the visual representation world model to learn future visual state prediction, and then introduce 3D object flow on top of the pretrained model and jointly train the generation of both modalities.

Unified Multi-Task Inference. After training, DynaRep supports multiple inference modes by configuring the noise states and denoising trajectories of the two modalities. In joint generation, the two streams share the same solver steps, while modality-specific noise schedules keep the visual representation stream ahead of flow in denoising progress, with flow remaining at a relatively higher noise level. This creates a latent-forcing-style asymmetric denoising trajectory, in which progressively refined future visual representations continuously provide informative state constraints for flow generation.

For visual-conditioned flow generation, ground-truth future visual representations are held clean while the model generates 3D object flow from noise, inferring object motion from the observed scene evolution. Conversely, for flow-conditioned visual generation, the given object flow is held clean while the model generates future visual representations. Thus, the same model supports joint generation, visual-conditioned flow generation, and flow-conditioned visual generation.

4 Experiments

4.1 Experimental Setup

We evaluate DynaRep on 3D object flow generation, DINO-VAE representation quality, component ablations, and conditional inference modes. The evaluation spans simulated and real-world manipulation data. For flow generation, we compare DynaRep with video-generation-and-tracking pipelines and direct-flow baselines. Appendix Sec. C provides details on the datasets, benchmark protocols, baseline implementations, evaluation metrics, and training hyperparameters.

4.2 Evaluation of Flow Generation

Quantitative Analysis. We first evaluate DynaRep on 3D object flow generation. Given the initial frame, the initial 3D point set of the target object, and a task description, the model generates the full future object flow sequence. Table 1 compares DynaRep with video generation pipelines followed by 3D tracking, as well as methods that directly predict object flow.

DynaRep achieves strong trajectory prediction performance across all three benchmarks, demonstrating its ability to accurately model future object motion. On Sim7, DynaRep achieves an ADE of 0.0530 m, an FDE of 0.0966 m, and a PWT of 0.5561. It also maintains stable and competitive performance on the real-world Real3 and SpatialCog benchmarks, indicating robustness to more complex appearance variations, interaction patterns, and motion distributions. Compared

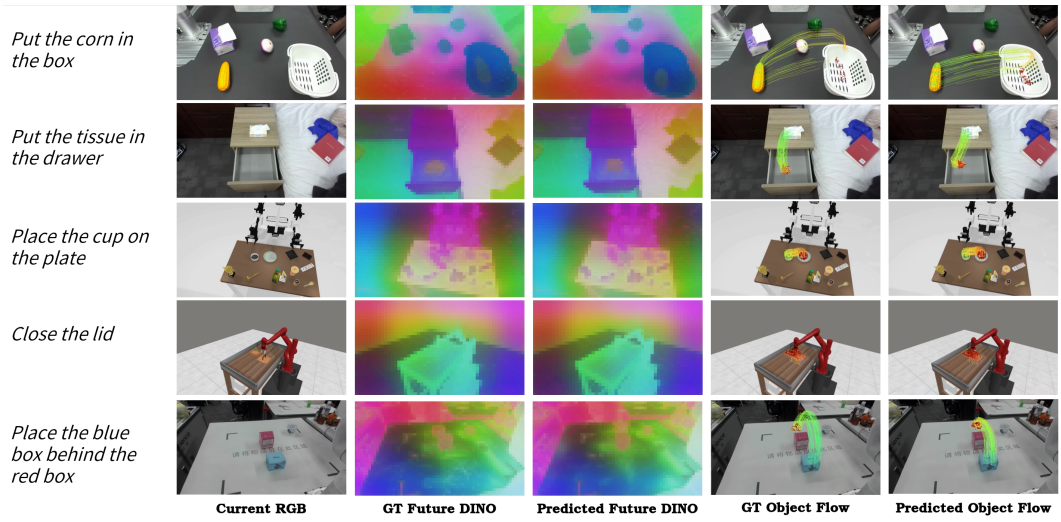

Figure 4: **Joint generation of future DINO representations and 3D object flow.** DynaRep jointly predicts future DINO representations and 3D object flow from the current observation and task instruction, producing future visual states and object trajectories that closely match the ground truth.

with methods that jointly generate future states and flow in pixel-reconstruction-oriented VAE latent spaces, DynaRep models future dynamics in the DINO representation space, providing richer semantic and geometric state constraints for flow generation. Overall, these results show that joint modeling in a visual representation space improves the trajectory accuracy and stability of 3D object flow generation.

We further conduct out-of-distribution (OOD) evaluation on the RoboMind dataset, which is not used during training, with results reported in Table 2. DynaRep maintains stable 3D object flow generation under substantial shifts in visual appearance and scene distribution, and exhibits stronger generalization than pixel-reconstruction-oriented baselines. These results further suggest that DINO representations reduce reliance on low-level appearance by capturing object semantics and spatial structure, providing a more transferable state representation for object-centric dynamics modeling across diverse scenes.

Table 2: **OOD 3D flow generation on RoboMind.** We report ADE, FDE, and mean PWT.

Method	ADE↓	FDE↓	PWT↑
NovaFlow	<u>0.1171</u>	0.2401	**0.4106**
MolmoMotion-AR (H1)	0.1986	0.3054	0.0711
μ_0	0.1639	<u>0.2125</u>	0.0595
DynaRep	**0.0718**	**0.1299**	<u>0.3453</u>

Qualitative Analysis. We also inspect DynaRep's predictions qualitatively. Fig. 4 visualizes the generated 3D object flow and future DINO representations across diverse manipulation scenarios. In the examples shown, the generated flow closely follows the target object's motion direction and overall trajectory while maintaining geometric consistency over long prediction horizons. Meanwhile, the predicted future visual representations preserve clear object structure and evolve consistently with the corresponding flow, demonstrating that DynaRep can jointly model future visual states and object-centric dynamics within a unified framework.

4.3 Analysis of DINO-VAE Representations

DINO-VAE aims to compress DINO features while retaining geometric and semantic information relevant to world modeling. We therefore analyze its representation quality from two perspectives. First, we use frozen downstream decoders to evaluate the geometric and semantic information retained in the reconstructed features. Second, we directly probe the latent space to assess whether high-level semantic structure survives compression. Detailed evaluation protocols and implementation settings are provided in Appendix C.4.

Table 3: **Reconstruction quality of DINO-VAE.** Frozen depth-estimation and semantic-segmentation probes are used to evaluate geometric and semantic information preserved after compression. Under the same latent budget, DINO-VAE reconstruction remains close to the original DINOv3-L features and consistently outperforms matched-budget PCA reconstruction.

		Depth			Segmentation	
Features fed to the frozen probe	cos ↑	AbsRel ↓	RMSE (m) ↓	δ_1 ↑	mIoU ↑	$\text{IoU}_{\text{Target}}$ ↑
DINOv3-L	1.000	0.0471	0.0987	0.9777	0.9204	0.6977
PCA, matched budget	0.851	0.1714	0.2410	0.6873	0.8753	0.5471
DINO-VAE reconstruction (ours)	0.959	0.0592	0.1160	0.9660	0.8982	0.6141

Table 4: **Semantic structure of the DINO-VAE latent space.** Top-1 probing accuracy on UCF101 shows that DINO-VAE preserves rich semantic information under $4\times$ temporal and $8\times$ channel compression. Single and TTA denote single-view and test-time-augmented evaluation.

Features fed to the frozen probe	**Temporal**	**Channel**	Single ↑	TTA ↑
CogVideoX VAE	$4\times$	—	14.51	14.14
Wan2.2 VAE	$4\times$	—	16.94	16.29
AToken	$4\times$	—	30.83	30.29
DINOv3-L	$1\times$	$1\times$	92.84	93.26
DINO-VAE reconstruction (ours)	$1\times$	$1\times$	90.35	90.70
DINO-VAE latent (ours)	$4\times$	$8\times$	79.17	81.79

Reconstruction Quality Evaluation. We assess geometry with depth estimation, semantics with segmentation, and feature-level similarity with the cosine similarity between reconstructed and original DINO features. Each decoder has fewer than $1M$ parameters and is trained only on original DINOv3-L features. We apply these frozen decoders to original DINO features, DINO-VAE reconstructions, and matched-budget PCA reconstructions. PCA serves as a linear compression baseline for DINO features. As shown in Table 3, DINO-VAE reconstruction retains performance close to the original DINO features on both depth estimation and semantic segmentation, while substantially outperforming PCA reconstruction. This indicates that DINO-VAE enables high-quality reconstruction of DINO features while preserving geometric and semantic information.

Semantic Structure of the Latent Space. Since DynaRep performs generative modeling directly in the DINO-VAE latent space, we further evaluate the semantic information retained in the latent representation. Following the DINOv3 protocol, we train a linear probe on UCF101 for classification. As shown in Table 4, the original DINOv3-L features achieve a Top-1 accuracy of 92.84%, while the compressed DINO-VAE latent achieves 79.17%. In contrast, pixel-reconstruction-oriented video VAE latents from CogVideoX and Wan2.2 achieve only 14.51% and 16.94% (Guo et al., 2026). This substantial gap indicates that DINO-VAE retains considerably richer semantic structure in its latent space. Overall, even under strong compression, the DINO-VAE latent preserves high-level semantic information, providing an effective generative space for the subsequent joint modeling of visual representations and flow.

4.4 Ablation Studies

We further conduct ablation studies to assess the contribution of key components to 3D object flow generation, including DINO-VAE, future DINO generation, modality-specific noise scheduling, and visual representation world model pretraining, as summarized in Table 5. Removing **future DINO generation** and conditioning flow only on first-frame DINO features increases ADE/FDE from $0.0315/0.0738$ to $0.0517/0.1147$, while PWT drops from 0.6570 to 0.5483. This shows that

Table 5: **Ablation studies of DynaRep.**

Method	**ADE↓**	**FDE↓**	**PWT↑**
DynaRep w/o pretrain	0.0597	0.1496	0.5500
DynaRep w/o DINO gen	0.0517	0.1147	0.5483
DynaRep w/o noise schedule	0.0477	0.1072	0.5440
DynaRep w/o DINO-VAE	0.0439	0.1095	0.6160
DynaRep (ours)	**0.0315**	**0.0738**	**0.6570**

explicitly predicting DINO representations provides temporally evolving visual states that more effectively constrain flow generation. Removing **DINO-VAE** and directly using the original DINO features also degrades performance, validating the benefit of a compact representation latent space.

Replacing **modality-specific noise scheduling** with a joint schedule further reduces performance, indicating that asymmetric denoising helps visual representations provide more effective conditioning for flow. In addition, joint training from random initialization underperforms pretrained initialization, suggesting that **pretraining future DINO generation** provides a useful dynamics prior for flow modeling. Overall, all ablations lead to performance degradation, demonstrating the complementary contributions of these components in DynaRep.

4.5 Bidirectional Dynamics Inference

We evaluate the bidirectional conditional inference capability of DynaRep from two complementary directions: visual-conditioned inverse dynamics and flow-conditioned forward dynamics.

Visual-Conditioned Inverse Dynamics. In the inverse dynamics setting, future DINO representations are provided as conditions, while the corresponding 3D object flow is generated from noise. This evaluates whether DynaRep can recover object motion from future visual-state evolution. Since Sim7 provides ground-truth 3D trajectories, we quantitatively evaluate performance using ADE, FDE, and PWT, and compare against tracking-based methods such as Track4World. The quantitative results are summarized in Table 6. The results show that DynaRep can effectively infer 3D object flow consistent with the observed scene evolution. We further provide qualitative results on real-world robotic manipulation data in the appendix, as shown in Fig. 8.

Table 6: **Inverse dynamics on Sim7.** We evaluate 3D object flow recovery from future visual observations using ADE, FDE, and PWT, comparing DynaRep-IDM with Track4World.

Method	ADE ↓	FDE ↓	PWT ↑
Track4World	0.1079	0.1726	0.4636
DynaRep-IDM (ours)	**0.0370**	**0.0849**	**0.6220**

Flow-Conditioned Forward Dynamics. We further evaluate conditional generation in the opposite direction. Given the initial visual state and future 3D object flow, DynaRep treats flow as a clean dynamics condition and generates the corresponding future DINO representations, thereby modeling future scene evolution under prescribed object motion. We qualitatively analyze the generated future DINO features and their corresponding depth predictions, as shown in Fig. 9. The visualizations show that the generated representations exhibit spatial and geometric changes consistent with the conditioned flow, with additional results provided in the appendix.

5 Conclusion

We presented DynaRep, which jointly generates future visual representations and 3D object flow within a unified framework, exploring predictive visual representations as a future-state space for embodied world modeling. Experiments show that, compared with pixel-reconstruction-oriented latent spaces, DINO-derived representation spaces provide more effective semantic and geometric constraints for object dynamics, leading to more accurate and stable 3D object flow generation across simulated and real-world manipulation scenarios. Ablation studies further demonstrate the importance of DINO-VAE, future representation prediction, modality-specific noise scheduling, and representation world model pretraining. Beyond joint generation, DynaRep also supports visual-conditioned inverse dynamics and flow-conditioned forward dynamics within the same model, revealing a bidirectional conditional relationship between future visual states and object-centric dynamics. Overall, our results suggest that future-state prediction in embodied world models need not be restricted to pixel-reconstruction spaces, and that predictive visual representations can provide an effective state space for bridging scene evolution and object-centric dynamics.

Limitations and Future Work. A key limitation of DynaRep is the difficulty of obtaining accurate 3D object flow annotations in real-world scenes, where supervision often relies on existing 3D trackers and may contain tracking noise. Future work will explore more reliable flow acquisition and extend DynaRep to longer-horizon tasks with more complex temporal dynamics.

6 AI USE STATEMENT

We used generative AI tools to assist with language polishing, grammar correction, and limited portions of experimental code development. The core methodology, experimental design, result analysis, and primary manuscript preparation were carried out by the authors, with AI serving only a limited supporting role. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final manuscript and its conclusions.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, et al. AgiBot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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.

Tianci Bi, Xiaoyi Zhang, Yan Lu, and Nanning Zheng. VFM-VAE: Vision foundation models can be good tokenizers for latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2510.18457.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Gabrijel Boduljak, Yushi Lan, Christian Rupprecht, and Andrea Vedaldi. VFMF: World modeling by forecasting vision foundation model features. *arXiv preprint arXiv:2512.11225*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Haonan Chen, Junxiao Li, Ruihai Wu, Yiwei Liu, Yiwen Hou, Zhixuan Xu, Jingxiang Guo, Chongkai Gao, Zhenyu Wei, Shensi Xu, Jiaqi Huang, and Lin Shao. MetaFold: Language-guided multi-category garment folding framework via trajectory generation and foundation model. *arXiv preprint arXiv:2503.08372*, 2025.

Egor Cherepanov, Nikita Kachaev, Alexey K. Kovalev, and Aleksandr I. Panov. Memory, benchmark & robots: A benchmark for solving complex tasks with reinforcement learning. *arXiv preprint arXiv:2502.10550*, 2025.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *International Journal of Computer Vision (IJCV)*, 2022.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-rae: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *arXiv preprint arXiv:1909.12271*, 2019.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4D reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Seungjae Lee, Yoonkyo Jung, Inkook Chun, Yao-Chih Lee, Zikui Cai, Hongjia Huang, Aayush Talreja, Tan Dat Dao, Yongyuan Liang, Jia-Bin Huang, and Furong Huang. TraceGen: World modeling in 3D trace space enables learning from cross-embodiment videos. *arXiv preprint arXiv:2511.21690*, 2025.

Seungjae Lee, Yoonkyo Jung, Jusuk Lee, Jonghun Shin, Amir Hossein Shahidzadeh, Yao-Chih Lee, H. Jin Kim, Jia-Bin Huang, and Furong Huang. μ_0: A scalable 3D interaction-trace world model. *arXiv preprint arXiv:2606.13769*, 2026.

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. EgoWAM: World action models beyond pixels with in-the-wild egocentric human data. *arXiv preprint arXiv:2607.08436*, 2026a.

Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. NovaFlow: Zero-shot manipulation via actionable flow from generated videos. *arXiv preprint arXiv:2510.08568*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Jiahao Lu, Jiayi Xu, Wenbo Hu, Ruijie Zhu, Chengfeng Zhao, Sai-Kit Yeung, Ying Shan, and Yuan Liu. Track4World: Feedforward world-centric dense 3D tracking of all pixels. *arXiv preprint arXiv:2603.02573*, 2026.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4RC: 4D reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2112.10752.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. *arXiv preprint arXiv:2410.00425*, 2024.

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

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. SpatialTracker: Tracking any 2D pixels in 3D space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. VA-VAE; arXiv:2501.01423.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

Jianing Zhang, Chenhao Zheng, Yajun Yang, Max Argus, Rustin Soraki, Winson Han, Taira Anderson, Chun-Liang Li, Shuo Liu, Jiafei Duan, Zhongzheng Ren, Jieyu Zhang, and Ranjay Krishna. MolmoMotion: Forecasting point trajectories in 3D with language instruction. *arXiv preprint arXiv:2606.18558*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.11690.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, 2025.

APPENDIX

A ADDITIONAL METHOD DETAILS

A.1 3D OBJECT FLOW ENCODING

Given the initial 3D position of the i-th point on the target object, $p_i^0 \in \mathbb{R}^3$, and its position at future timestep t, $p_{t,i}$, we define 3D object flow as the cumulative displacement relative to the first frame:

$$f_{t,i} = p_{t,i} - p_i^0. \tag{10}$$

The complete flow sequence for one sample is therefore represented as $F \in \mathbb{R}^{T_F \times P \times 3}$, where T_F denotes the prediction horizon and P is the number of tracked points. Since flow is defined directly in metric 3D space, we first normalize it using a global scale factor s_{flow}:

$$\widetilde{F} = \frac{F}{s_{\text{flow}}}. \tag{11}$$

The normalized flow is used directly as a continuous generative representation without an additional VAE or patchification. After corruption, we flatten the temporal and point dimensions and project each flow token to the Transformer hidden dimension. To preserve the initial geometric location associated with each flow token, we construct a geometry embedding from the source point p_i^0. Specifically, we encode the absolute 3D position of each point and its local position relative to the object centroid. We define

$$PE_{\text{abs}} = \phi_{\text{abs}}(p_i^0), \tag{12}$$

and

$$PE_{\text{rel}} = \phi_{\text{rel}}(p_i^0 - c), \qquad c = \frac{1}{P}\sum_i p_i^0, \tag{13}$$

where ϕ_{abs} and ϕ_{rel} denote the corresponding Fourier positional encoding and MLP modules. The two geometric representations are processed by Fourier positional encoding and MLPs, and then fused to obtain $PE_{\text{geom}}(p_i^0) \in \mathbb{R}^{4096}$. All future tokens corresponding to the same tracked point share the same geometry embedding, anchoring the entire trajectory to its initial 3D location.

In addition, flow tokens use a spatiotemporal RoPE aligned with the visual representation stream. Let (u_i^0, v_i^0) denote the source pixel of point p_i^0 in the first frame. We map it to the corresponding DINO spatial patch (h_i^0, w_i^0). The flow token corresponding to the same point at future timestep t is then assigned the RoPE coordinate (t, h_i^0, w_i^0). The spatial coordinates (h_i^0, w_i^0) remain fixed across the trajectory, while only the temporal coordinate changes over time. Thus, the spatial coordinate indicates which visual representation patch the trajectory originates from in the first frame, rather than the point's future image location. The temporal coordinate is further aligned with that of the temporally compressed DINO-VAE representation, establishing explicit spatiotemporal correspondence between the visual representation and flow modalities.

A.2 MODALITY-SPECIFIC NOISE SCHEDULING

This section provides additional details on timestep sampling, modality-specific noise scheduling, and the asymmetric denoising trajectory used in DynaRep. For both the visual representation and 3D object flow, we adopt Waver timestep sampling. The base timesteps for the two modalities are independently sampled as

$$\tau_R \sim p_{\text{Waver}}(\tau), \qquad \tau_F \sim p_{\text{Waver}}(\tau). \tag{14}$$

The sampled timesteps are then mapped to modality-specific corruption levels through shifted schedules:

$$\sigma_R = S_{s_R}(\tau_R), \qquad \sigma_F = S_{s_F}(\tau_F), \tag{15}$$

where

$$S_s(\tau) = \frac{s\tau}{1 + (s-1)\tau}. \tag{16}$$

The corresponding forward corruption process is

$$x^m_{\sigma_m} = (1 - \sigma_m)x^m_0 + \sigma_m \epsilon_m, \qquad \epsilon_m \sim \mathcal{N}(0, I), \tag{17}$$

where $m \in \{R, F\}$ denotes the visual representation and flow modalities, respectively. We use different timestep shifts for the two modalities to account for their distinct generative characteristics. For example, at an input resolution of 256p, we use $s_R = 3$ for the visual representation and $s_F = 5$ for 3D object flow. During inference, the two modalities share the same solver steps but follow their respective modality-specific schedules. As illustrated in Fig. 5, this results in asymmetric denoising trajectories: the representation stream progresses ahead of the flow stream, while flow remains at a higher scheduler-defined corruption level. Consequently, progressively refined future world states can continuously constrain and guide flow generation.

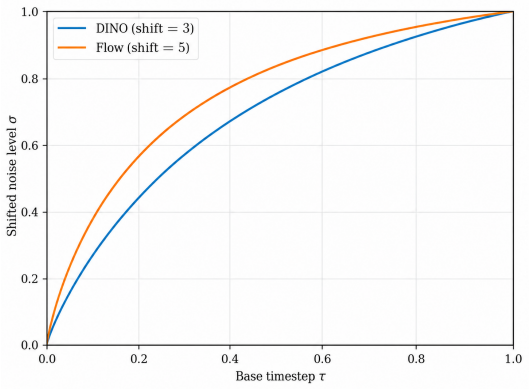

Figure 5: **Modality-specific shifted noise schedules.** The DINO-VAE latent and 3D object flow use different timestep shifts, with $s_R = 3$ and $s_F = 5$, respectively. A larger shift maps the same base timestep to a higher scheduler-defined corruption level, yielding asymmetric denoising trajectories in which the representation stream progresses ahead of the flow stream.

The visual representation stream further adopts a resolution-dependent timestep shift. In particular, as spatial resolution increases, the same nominal noise level can induce weaker effective corruption, since the representation contains more spatially redundant information and thus retains more recoverable structure under the same perturbation (Esser et al., 2024). Visual representations at different spatial resolutions exhibit different degrees of spatial redundancy and information structure, such that the same nominal corruption coefficient may correspond to different effective corruption levels across resolutions. Following the resolution-aware scheduling strategy of Cosmos3, we select the representation shift s_R according to the input resolution, such that different resolution buckets exhibit comparable effective corruption levels and denoising difficulty after schedule calibration. In this way, the resolution-dependent shift compensates for resolution-induced differences and maintains a relatively consistent generation process across representation resolutions. In contrast, the representation of 3D object flow is independent of image resolution: its temporal horizon, number of points, and feature dimensionality remain fixed across resolution buckets. We therefore keep the flow shift unchanged across all input resolutions rather than adapting it to the image resolution. This design calibrates the visual representation schedule separately for each resolution while preserving a consistent flow trajectory, thereby maintaining a stable asymmetric denoising relationship between the representation and flow streams.

A.3 Unified Inference Modes

By covering diverse representation–flow corruption configurations during training, DynaRep supports multiple dynamics inference modes within a single model by fixing one modality as a clean condition while generating the other, as shown in Table 7.

In joint generation, both modalities are initialized from noise and jointly generated along their respective modality-specific denoising trajectories. In inverse dynamics, ground-truth future visual representations are provided as a fixed clean condition, while the corresponding 3D object flow is generated from noise, allowing the model to infer object motion from future scene evolution.

Table 7: **Unified inference modes supported by DynaRep.**

Inference Mode	Visual Representation	3D object flow
Joint generation	Generated from noise	Generated from noise
DINO-conditioned inverse dynamics	Clean and fixed	Generated from noise
Flow-conditioned forward dynamics	Generated from noise	Clean and fixed

Conversely, in flow-conditioned forward dynamics, a clean object flow is fixed as the geometric dynamics condition, and the model generates the corresponding future visual representations, predicting future world states under the prescribed motion. Therefore, the same representation–flow generative model supports joint future prediction as well as bidirectional conditional dynamics inference by configuring the corruption states of the two modalities.

B Implementation Details

We next provide the training data, optimization settings, inference configuration, and model parameterization used in our experiments.

B.1 DINO-VAE Implementation

The frozen DINOv3-L encoder is excluded from all counts below. The trainable codec has 83.3M parameters in six modules (Table 8): temporal attention pooling reduces the T input frames to $n{+}1$ latent frames, a three-layer block-causal ViT encoder contextualizes them at the feature width $D{=}1024$, and a latent head emits μ and $\log\sigma^2$ at $d_z{=}128$. Decoding mirrors this, with a linear projection back to width D, a three-layer ViT decoder, and a temporal unpatchify layer that restores T frames. Both towers therefore operate at D rather than at d_z, and only two small modules touch the compressed width. Compression is $4\times$ in time and $8\times$ in channels with the spatial axis left intact, or $32\times$ overall. The N spatial tokens per frame are never reduced, so the compression ratio is independent of resolution, which allows training against a fixed token budget rather than a fixed frame size. The first frame occupies a window of its own, and attention is causal across latent frames while remaining bidirectional within each frame. Its gradient with respect to every later frame is exactly zero, a property asserted by a unit test, which makes the first latent an exact conditioning signal. Training uses AdamW at a peak learning rate of 10^{-4} under a warmup-stable-decay schedule for five epochs in `bfloat16`.

Table 8: **Trainable parameters of DINO-VAE.** The frozen DINOv3-L encoder is excluded. The two ViT towers dominate the count and are symmetric, while the two modules that change width account for 0.4M between them.

Module	Function	Params
Temporal attention pooling	$T \to n{+}1$ along time	3.15M
ViT encoder (3 layers)	context at width D	37.79M
Latent head	$D \to d_z$, emits μ and $\log\sigma^2$	0.26M
Latent entry projection	$d_z \to D$	0.13M
ViT decoder (3 layers)	context at width D	37.79M
Temporal unpatchify	$n{+}1 \to T$ along time	4.20M
Total		83.33M

B.2 Training Setup

The generative model is trained on $967{,}377$ clips from eleven sources (Table 9), grouped as the benchmark groups them: seven simulation suites, three real-world capture sources, and one real capture set with calibrated intrinsics. To alleviate data imbalance across multiple sources, sampling weights are proportional to the square root of each source's size rather than to its size. Videos are used at their native aspect ratio and resolution, with each side rounded to a multiple of 32, the long

side capped at 1280 pixels and the short side floored at 128. Frame counts follow from a fixed token budget rather than a fixed clip length, so a larger frame yields fewer frames. Each sample supplies 64 query points, sampled on the visible surface of the object mask and tracked over 48 future frames. Their displacements are divided by a single fixed constant, so that the flow branch sees a unit-scale target. Clips of the evaluation benchmark are removed before the training split is formed. Training runs on 64 NVIDIA H100 GPUs, eight nodes of eight, in `bfloat16`. Each step consumes 64 packed sequences of at most 55,296 tokens. The optimizer is AdamW at a peak learning rate of 2×10^{-4} with no weight decay, warming up over 500 steps and then decaying on a cosine schedule to half the peak. An exponential moving average of the weights is maintained throughout, and that average is what we evaluate. A step takes about 14 seconds, and all results in this paper come from the checkpoint at step 8,000, reached in roughly 34 hours.

Table 9: **Mixture dataset in training.** Clip counts per source before the hold-out sets are removed and before the train/validation split. Group names follow those used by the benchmark.

Source	Group	Clips
AgiBot World (AgiBot-World-Contributors et al., 2025)	Real3	597,272
Something-Something v2 (Goyal et al., 2017)	Real3	84,153
EPIC-KITCHENS (Damen et al., 2022)	Real3	10,519
RoboTwin (Mu et al., 2025)	Sim7	115,775
Meta-World (Yu et al., 2019)	Sim7	52,477
ManiSkill (Tao et al., 2024)	Sim7	9,033
RoboCasa (Nasiriany et al., 2024)	Sim7	6,616
MIKASA (Cherepanov et al., 2025)	Sim7	4,599
RLBench (James et al., 2019)	Sim7	3,137
MetaFold (Chen et al., 2025)	Sim7	2,013
Own capture	SpatialCog	81,783
Total		967,377

B.3 Inference Setup

Generation is conditioned on a single observed frame and the task description. The codec's temporal aggregation is causal and its first window holds one frame, so the first latent is an exact function of that frame alone (§3.2) and conditioning needs no approximation. The flow branch is anchored the same way, on the absolute frame-0 positions of the query points. We sample with a UniPC solver for 10 steps at a classifier-free guidance scale of 1.5, using the moving-average weights. As in training, the two modalities are denoised on separate timestep schedules, and the two schedules are shifted differently. The representation branch keeps the shift its resolution tier was pretrained with. The flow branch is sampled at a lower shift than it was trained at, which concentrates solver steps at the low-noise end where trajectory detail is resolved. Both branches predict the clean sample rather than a velocity. Generated latents are mapped back to features by the frozen codec decoder. Predicted displacements are rescaled by the constant used to normalize them and accumulated from the anchor, which yields absolute positions in meters.

B.4 Model Parameters

We adapt the Cosmos3-Nano architecture, a mixture-of-transformers in which an understanding expert and a generation expert share one attention operation over one token sequence, and initialize from its released weights. Of its 15.18B parameters we train 6.97B, or 46%, while keeping the understanding expert, with 8.21B parameters, frozen. Almost all of the trainable mass is the generation expert itself, at 6.95B. The modules added for this work are small by comparison: 17.8M parameters for a timestep embedding, 3.0M for the object-flow branch that carries and reads out the trajectory tokens, and 0.5M each for the two projections that move representation latents in and out of the trunk. Those two projections are single linear maps between the codec's 128 latent channels and the trunk width of 4096. They are the entire interface between the codec and the generative model, with no adapter, resampler or feature pyramid between them. The DINOv3-L encoder and the DINO-VAE codec are frozen and excluded from these counts.

Table 10: **Model configuration and hyperparameters of DynaRep.**

Category	Hyperparameter	Value
Backbone	Backbone	Cosmos3-Nano
	Hidden dimension	4096
	Total parameters	15.18B
	Trainable parameters	6.97B
Visual representation	Encoder	DINOv3-L
	DINO feature dimension	1024
	DINO-VAE latent dimension	128
	Temporal compression	$4\times$
	Channel compression	$8\times$
3D object flow	Number of points P	64
	Prediction horizon T	48
	Flow dimension	3
	Normalization scale s_{flow}	0.1131
Noise schedule	Timestep sampling	Waver
	Cross-modal timestep sampling	Independent
	Representation shift s_R	256p: 3; 480p: 5; 720p: 10
	Flow shift s_F	5
Objective	Prediction target	x-prediction
	Representation loss weight	1.0
	Flow loss weight λ_F	1.0

C Experimental Details

C.1 Datasets and Benchmarks

We evaluate flow generation on WorldFlowBench, a benchmark of $2{,}000$ manipulation clips drawn from five sources and reported as three groups: **Sim7** (400 clips) aggregates seven simulation suites, **Real3** ($1{,}200$ clips) aggregates three real-world capture sources, and **SpatialCog** (400 clips) is a separate real capture set with calibrated intrinsics. Each clip provides a single conditioning RGB frame at 288×480, a natural-language task description, 64 query points specified by their pixel location in that frame, and a ground-truth trajectory spanning 49 frames, stored as 48 previous-frame-relative displacements from which absolute positions follow by cumulative summation. The 49 frames are subsampled from source episodes of varying length. All coordinates are metric and expressed in the frame-0 camera frame, with undistorted pinhole intrinsics supplied per clip. Metric scale has different provenance across the three groups — simulator ground truth for Sim7, calibrated intrinsics for SpatialCog, and up-to-scale monocular estimation for Real3. The read-out and probing experiments use separate data described with their own protocols: held-out manipulation episodes for the depth and segmentation probes, and UCF101 under the DINOv3 protocol for the latent probe.

C.2 Evaluation Metrics

For flow generation, we report ADE, the mean displacement error over all query points and timesteps; FDE, the final-timestep error; and PWT, the average fraction of predictions within $\{0.01, 0.02, 0.05\}$ meters of the ground truth. Both errors are in meters, and metrics are computed per clip over the full horizon and macro-averaged within each group. For the depth read-out, AbsRel is the mean error relative to the ground-truth depth of each pixel, RMSE is the root-mean-square error in meters, and δ_1 is the fraction of pixels accurate to within a factor of 1.25. For the segmentation read-out, IoU is the intersection over union of a class, and mIoU is its mean over the five classes. Latent probing reports top-1 accuracy under two protocols: Single uses one deterministic view, the largest center square crop with the first frame, the last frame and uniformly spaced intermediate frames, while TTA averages the post-softmax class probabilities of six views, formed by three square crops along the longer image axis and two temporally interleaved frame sequences.

C.3 BASELINES

We compare DynaRep against two families of baselines: video-generation-and-tracking pipelines that recover 3D motion from generated future videos, and direct-flow methods that predict future 3D trajectories without explicit RGB video generation.

Video-generation-and-tracking baselines. **NovaFlow** (Li et al., 2025) first generates a future video conditioned on the initial observation and task description, and then extracts actionable 3D object flow from the generated video using off-the-shelf perception modules. **Cosmos3-Nano + Track4World** combines the Cosmos3-Nano world model (NVIDIA, 2026) for future RGB video generation with Track4World (Lu et al., 2026), a feed-forward world-centric 3D tracker, to recover the trajectories of query points from the generated video. **Cosmos3-Nano + 4RC** uses the same generated future video but replaces the tracker with 4RC (Luo et al., 2026), which reconstructs scene geometry and time-dependent motion from monocular video to obtain the corresponding 3D point trajectories. **LingBot-VA + Track4World** uses LingBot-VA (Li et al., 2026b) to autoregressively generate future visual observations and applies Track4World to recover world-centric 3D trajectories from the generated sequence. **LingBot-VA + 4RC** similarly couples LingBot-VA with 4RC, using the generated future video as input to 4RC for 3D motion reconstruction and trajectory extraction. These pipelines therefore inherit errors from both future video generation and subsequent 3D tracking or reconstruction.

Direct-flow baselines. μ_0 (Lee et al., 2026) is a 3D interaction-trace world model that directly predicts future trajectories of semantic interaction points in a normalized, anchor-relative 3D trace space. **TraceGen** (Lee et al., 2025) models future dynamics directly in a compact 3D trace space, learning transferable motion priors from cross-embodiment human and robot videos. **EgoWAM** (Li et al., 2026a) studies world-action modeling beyond pixels and includes a 3D-flow variant that predicts camera-stabilized 3D motion as the world-model target. **MolmoMotion-AR** (Zhang et al., 2026) predicts language-conditioned future 3D trajectories of queried image points autoregressively, representing coordinates as discrete tokens generated from the visual and language context. Unlike the video-generation-and-tracking pipelines, these methods directly model geometric motion without reconstructing an intermediate future RGB video.

For every method, the conditioning frames remain fixed across rollout chunks, and only the model's own predictions are fed back as history.

C.4 DINO-VAE EVALUATION PROTOCOLS

We provide the implementation details of the read-out and probing protocols used to evaluate the information preserved by DINO-VAE.

PCA visualization. We use PCA to visualize the spatial structure of DINO-VAE latent representations. The latent feature dimension is projected onto the first three principal components, which are then mapped to RGB channels to obtain a spatial visualization of the latent space. Spatial locations with similar colors correspond to more similar representations in the projected feature space. This visualization provides an intuitive view of how semantic and geometric structure is organized after temporal and channel compression.

Segmentation read-out. The segmentation probe is a 0.76M-parameter decoder similar to the depth probe. Four parallel projections read the same last-layer feature map and their outputs are resampled to four scales. These correspond to four sampling rates over one feature map rather than dilated convolutions over a single scale. The fused output is a per-pixel distribution over five classes: background, table, robot, clutter, and the object currently being manipulated. The loss is an equally weighted sum of cross-entropy and Dice, with inverse-frequency class weights computed on the training split alone. Training uses a peak learning rate of 2×10^{-3}.

Semantic probe. The semantic probes follow the DINOv3 evaluation protocol. On UCF101, the probe is linear: patch tokens are pooled over space and time into one vector per clip, which passes through a LayerNorm and a single linear classifier, resulting in 0.11M parameters with a 1024-dimensional input. The pooling operation is chosen once on a held-out selection split and then held

fixed across every row, so that rows stay comparable. The probe is trained with cross-entropy for twenty epochs at a fixed weight decay of 10^{-2}, with the learning rate selected on the held-out split from ten candidate values on UCF101.

Depth read-out. The three read-out probes take the last layer of the frozen DINOv3-L encoder as their only input, since at deployment a generated feature map is the only thing available. The depth probe is a 0.63M-parameter convolutional decoder that forms four scales from this single feature map, fuses them top-down while upsampling, and emits one strictly positive channel at full resolution, read as metric depth. Its loss is a weighted sum of four terms: a data term in disparity space, a scale-invariant logarithmic term, a gradient term, and an edge term. The last two carry the larger weights, so that depth boundaries are not smoothed away. Training runs for eight epochs at a peak learning rate of 2×10^{-3} under a one-cycle schedule.

C.5 OOD Evaluation Protocol

The main benchmark draws its three groups from the same corpora the generative model trains on, so it measures interpolation within the training distribution. To measure generalization, we instead evaluate on an out-of-distribution dataset taken from RoboMIND 2.0, a source that is never seen during training. The evaluation includes 200 clips spanning two manipulation tasks. Every field of the evaluation contract matches the main benchmark exactly: 49 video frames at 288×480, 48 flow frames, 64 query points, displacements relative to the previous frame, and coordinates in the frame-0 camera frame in meters. The protocol, the metrics, and the zero-motion reference are therefore the ones used throughout, and the only thing that changes is the corpus. The evaluation results are summarized in Table 2.

D Additional Experimental Results

We provide additional qualitative results to complement the quantitative evaluations in the main paper, including joint generation, DINO-VAE representation analysis, inverse dynamics, and flow-conditioned forward dynamics.

D.1 Additional Flow Generation Results

Figure 6 presents additional qualitative examples of jointly generated future DINO representations and 3D object flow across diverse manipulation scenarios.

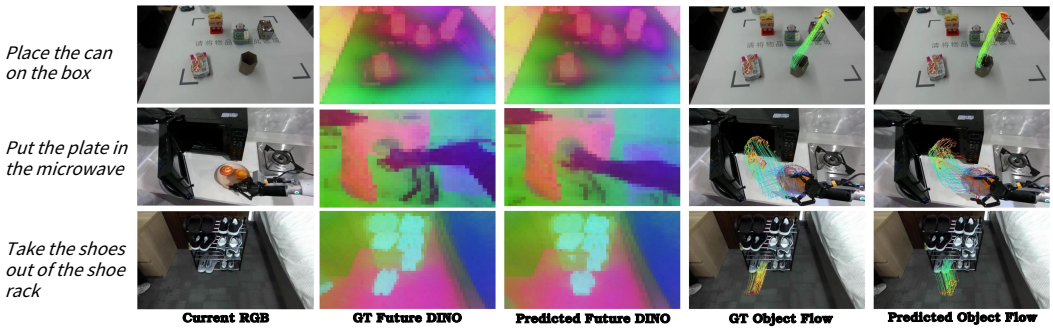

Figure 6: **Generated 3D object flow and future DINO representations.**

D.2 Additional DINO-VAE Analysis

We further provide qualitative analyses of the semantic and geometric structure preserved by DINO-VAE, together with additional segmentation and depth read-out visualizations.

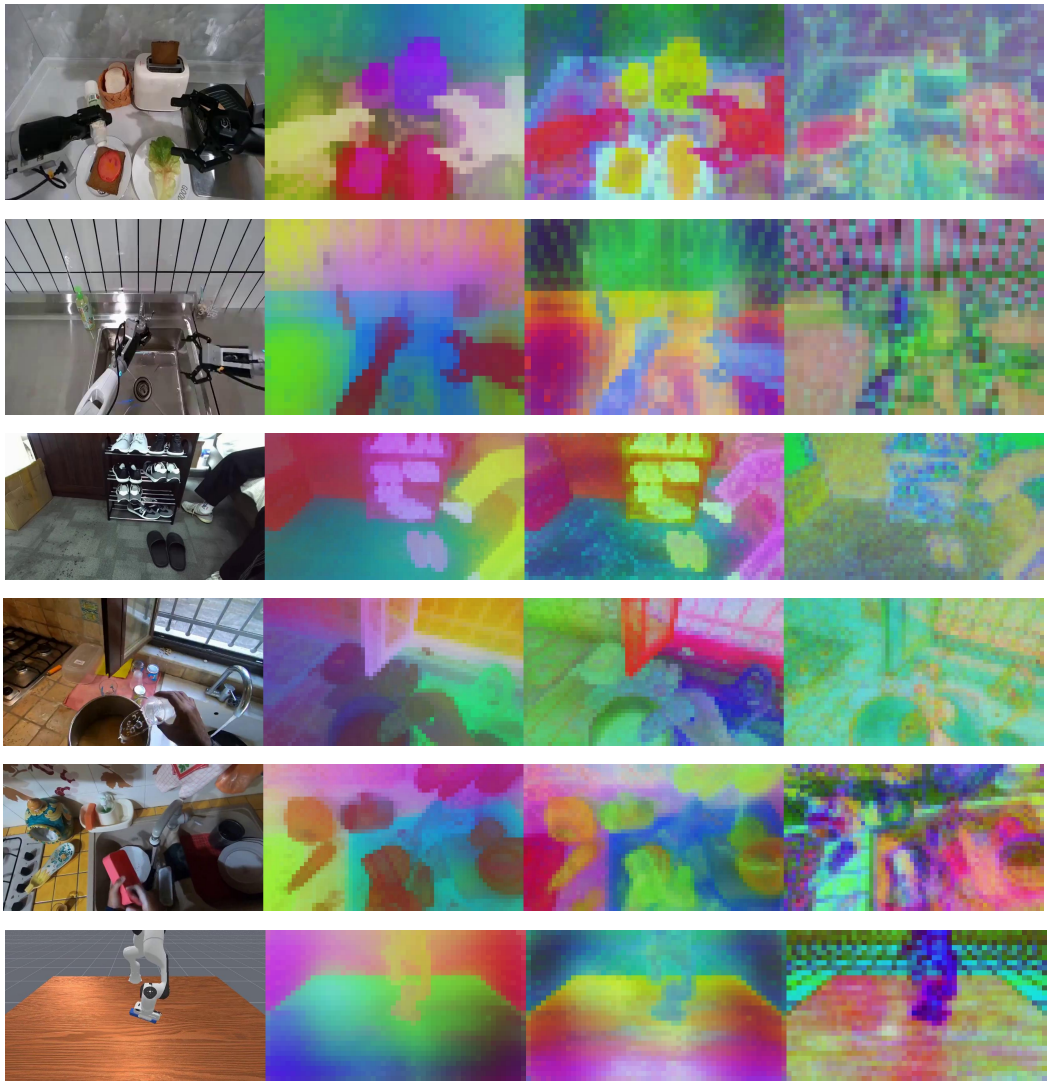

Figure 7: **PCA visualization of representation structure.** We map the first three principal components of DINOv3-L features, DINO-VAE latents, and Wan2.2 VAE latents to RGB channels. DINO-VAE largely preserves the object-level semantic and geometric organization of the original DINO features, while Wan2.2 VAE exhibits a more entangled feature structure with weaker separation between objects and background.

PCA Visualization of Representation Structure. To provide a comparison of the semantic and geometric structure encoded by different latent spaces, we visualize DINOv3-L features, DINO-VAE latents, and Wan2.2 VAE latents using PCA, mapping the first three principal components to RGB channels. In these visualizations, spatial locations with similar colors correspond to more similar representations in the projected feature space. As shown in Fig. 7, the original DINO features exhibit clear object-level structure, with distinct regions for different objects as well as foreground and background. Despite substantial temporal and channel compression, the DINO-VAE latent largely preserves this structured organization: features within the same object remain relatively coherent, while semantically and geometrically distinct regions are clearly separated. In contrast, the Wan2.2 VAE latent exhibits a more entangled and less structured PCA pattern, with weaker boundaries between objects and between foreground and background. These visualizations suggest that DINO-VAE preserves the object-centric semantic and geometric structure of DINO features

more effectively, while pixel-reconstruction-oriented VAE latents exhibit less explicit object-level organization.

D.3 Additional Inverse Dynamics Results

We provide additional qualitative results for DINO-conditioned inverse dynamics in Fig. 8. In this setting, future visual representations are provided as clean conditions, while 3D object flow is initialized from noise and inferred by the model. The predicted trajectories are consistent with the future scene evolution encoded by the visual representations, demonstrating that DynaRep can recover object-centric motion from future visual states within the same unified generative model.



Figure 8: **Visualization of DINO-conditioned inverse dynamics.**

D.4 Additional Flow-Conditioned Forward Dynamics Results

Figure 9 presents additional qualitative results for flow-conditioned forward dynamics. Here, the 3D object flow is provided as a clean geometric condition, while the future visual representations are generated from noise. The resulting representations reflect scene changes that are consistent with the prescribed object motion, showing that DynaRep can use explicit 3D dynamics to condition the prediction of future visual states.

D.5 Additional Segmentation Read-out Results

Figure 10 shows additional qualitative results of the segmentation read-out. The predicted segmentation maps preserve clear object-level organization and distinguish the manipulated objects, robot, table, clutter, and background across diverse scenes. These results further indicate that the visual representations retain semantic structure useful for dense object-centric understanding.

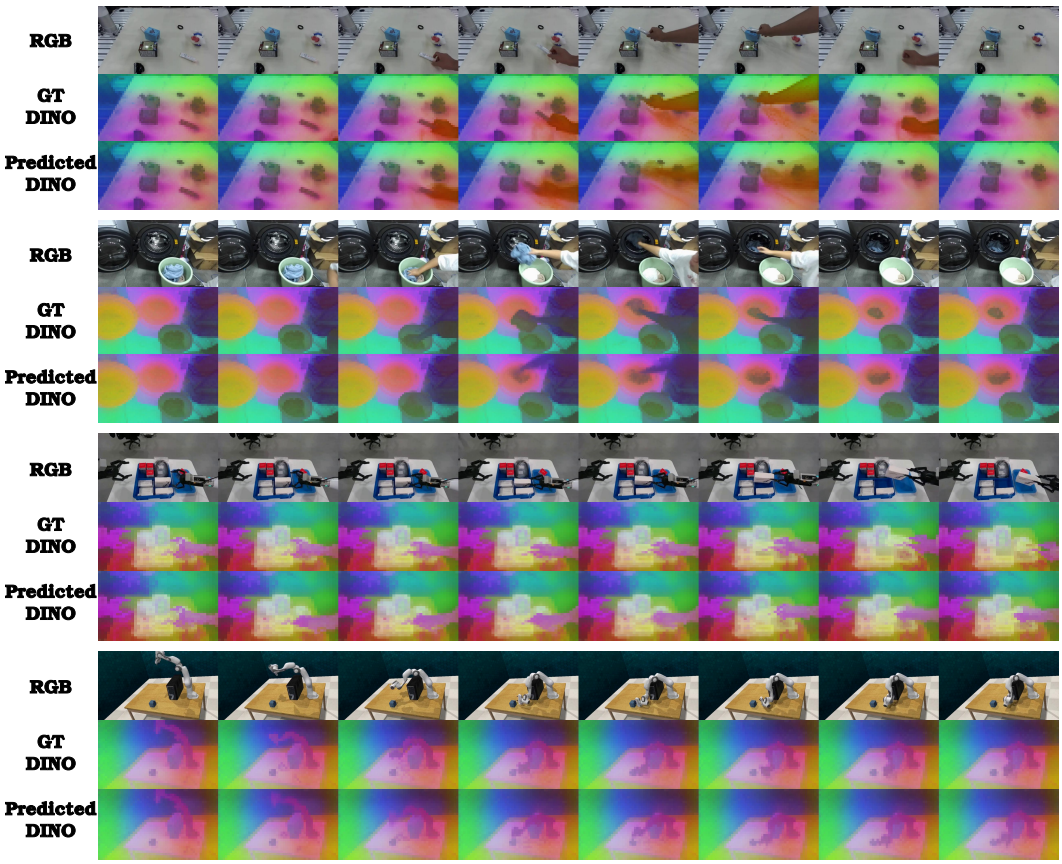

Figure 9: **Visualization of flow-conditioned forward dynamics.**

D.6 Additional Depth Read-out Results

We provide additional qualitative results of the depth read-out in Fig. 11. The predicted depth maps preserve the overall scene geometry and recover clear depth variations around objects and interaction regions. These results provide further qualitative evidence that the visual representations retain geometric information that can be effectively decoded for dense depth prediction.

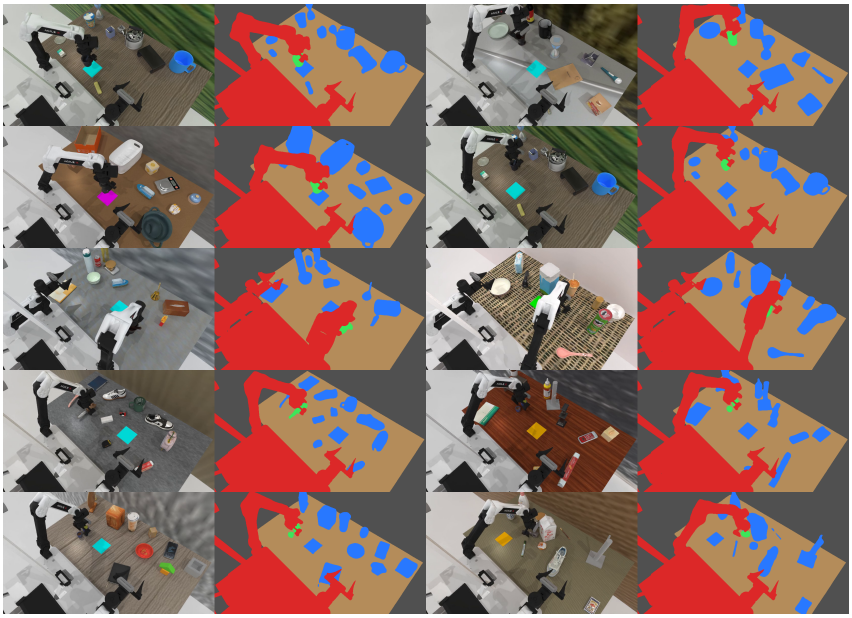

Figure 10: **Visualization of segmentation read-out.**

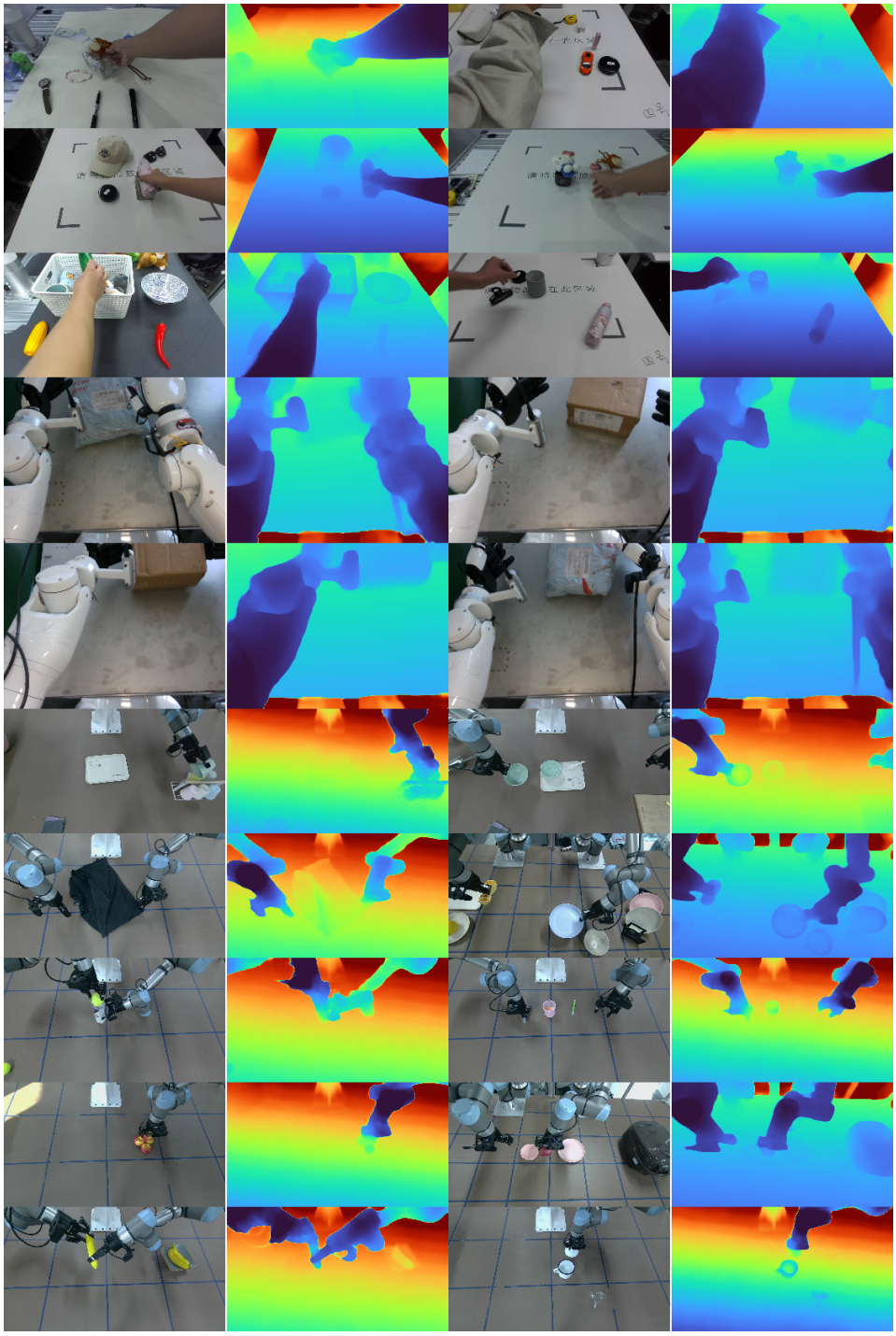

Figure 11: **Visualization of depth read-out.**