EmbodiedVAE: Disentangled Video VAE for Efficient and Controllable Embodied Manipulation

Jiayi Luo[1,2], Hanxin Zhu[3,*], Chen Gao[1,4],
Jiankun Wang[1], Cong Wang[5,2], Tianyu He[6],
Jianxin Li[1,†], and Zhibo Chen[3,2,†]

[1] Beihang University
[2] Zhongguancun Academy
[3] University of Science and Technology of China
[4] National University of Singapore
[5] CASIA, Institute of Automation, Chinese Academy of Sciences
[6] Microsoft Research Asia
[*]Project Lead [†]Corresponding Authors

Abstract. Latent diffusion models (LDMs) have recently significantly advanced embodied learning in constructing powerful embodied manipulation world models. However, despite the remarkable performance, existing LDMs predominantly rely on Variational Autoencoders (VAEs) optimized for natural scenes while failing to account for the unique characteristics of embodied manipulation scenarios, yielding latent representations that are neither compact nor controllable, thereby hindering efficient training of LDMs and precise robotic control. To solve this problem, we present **EmbodiedVAE**, a novel video VAE that provides **compact yet controllable** latent representations tailored for the robotic manipulation world models. Specifically, EmbodiedVAE adopts a dual-encoder, single-decoder architecture with an asymmetric spatio-temporal compression module, which automatically disentangles the robot arm's motion from background environment, resulting in overall compactness while providing explicit embodied latent to support fine-grained action control. To further preserve the temporal consistency of learned robotic motion latent, we introduce an optimal-transport-based consistency module that explicitly enforces motion fidelity and inter-frame coherence. Extensive experiments demonstrate that our proposed EmbodiedVAE achieves superior reconstruction quality with high compression rate, while enabling more precise action control in robotic manipulation scenarios with an average of 2dB PSNR improvement over state-of-the-art video VAEs. Code is available at: `https://github.com/Mutual-Luo/EmbodiedVAE`

Keywords: Video VAE, Embodied Manipulation

1 Introduction

Recent advances in latent diffusion models (LDMs) have demonstrated remarkable potential for embodied learning, particularly in constructing expressive em-

bodied manipulation world models [15, 21, 59], which enable embodied agents to imagine future interactions with environments at a low cost. At the heart of LDMs lies a variational autoencoder (VAE) that encodes high-dimensional data into latent representations through an encoder-decoder architecture [35, 41, 42]. Within this latent space, the diffusion process operates to progressively refine these representations, enabling high-fidelity visual generation [3, 4, 44, 54].

While the remarkable performance of LDMs, directly applying them to build embodied robotic manipulation world models still faces some challenges. In such scenarios, the core objective is to accurately predict the outcomes of arm-environment interactions under given future action conditions [6, 9, 23, 28, 29]. However, existing LDMs predominantly rely on image or video VAEs designed for natural scenes. As a result, their latents remain insufficiently compact and controllable, thereby *hindering efficient downstream training and precise robotic control*. While frame-by-frame compression with 2D image VAEs preserves temporal continuity, it fails to exploit inter-frame redundancy, producing large latents that hinder efficient LDM training [27,38, 56]. Conversely, although existing video VAEs achieve relatively compact latents via spatial-temporal compression, their temporal compression, which lacks explicit modeling of the robotic actions, often causes the motion semantics to be lost. [21,47,59]. As shown in Figure 1, given history frames and corresponding future actions, the world model trained upon the widely used Wan-VAE [44] yields high-quality visuals but fails to enable accurate action control. In summary, existing video VAEs in embodied scenarios still struggle to balance compression and motion fidelity, limiting their capacity for accurate motion control in downstream video generation.

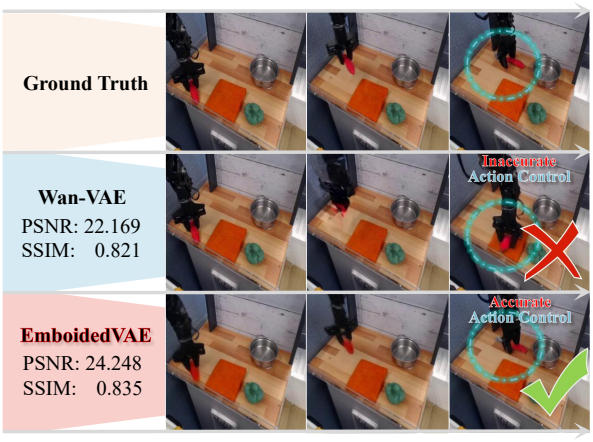

Fig. 1: Example comparison on the Bridge dataset [43] illustrating latent controllability of the video VAE in robotic manipulation scenarios, given history frames and future arm actions. The LDM backbone is fixed as IRASim-L (461 M) [59], while *the only difference lies in the video VAE used for training and inference.*

To address the aforementioned problems, in this work, we propose our novel **EmbodiedVAE**, a video VAE tailored for embodied robotic manipulation world models that produces compact yet controllable latent representations to enable efficient and effective downstream embodied learning. The core idea of our pro-

posed EmbodiedVAE is to automatically disentangle an input video into two compact latent spaces: one capturing the foreground fine-grained, robot-related motion information, and the other modeling the contextual background. This explicit disentangle allows the robotic latent to directly incorporate action conditions on the manipulator itself while remaining unaffected by irrelevant environmental noise, leading to more accurate and controllable action generation.

Specifically, EmbodiedVAE disentangles the robotic arm and environmental context with a dual-encoder, single-decoder architecture. The environment encoder captures scene context, while the robotic arm encoder models the manipulator's motion dynamics. *(1) To learn to attend to different types of information*, we employ a two-stage training scheme where robotic arm masks are utilized solely during training to facilitate the disentanglement of motion and context. After training, EmbodiedVAE operates without any mask input. In the first stage, two VAEs are trained for robotics and environment, and in the second, their frozen encoders guide a unified decoder to reconstruct complete videos from disentangled latents. *(2) To obtain highly compact latent representations*, we adopt an asymmetric compression strategy: the robotic arm encoder applies stronger spatial compression to focus on fine-grained motion, while the background encoder applies stronger temporal compression to exploit redundancy, yielding compact yet informative latent representations. *(3) To capture motion consistency in the robotic arm latent*, we introduce an optimal-transport-based loss that minimizes information transport cost across frames, encouraging stable motion dynamics and temporal coherence in the learned representations.

Our main contributions can be summarized as follows:

- We propose EmbodiedVAE, a novel video VAE tailored for embodied robotic manipulation that provides compact yet controllable latent representations.
- We achieve disentangled and compact latents using a dual-encoder, single-decoder architecture trained in two stages with asymmetric compression rates. An optimal-transport-based loss further enforces temporal consistency of robotic arm motion in the latent space.
- Extensive experiments on embodied manipulation tasks show that EmbodiedVAE achieves high-quality reconstruction under strong compression, enabling more accurate and consistent action control in the downstream tasks.

2 Related Work

2.1 Visual Autoencoders

Recent advances in visual autoencoders have greatly improved image and video generation by learning compact, expressive latent representations [50]. They typically encode visual data into a low-dimensional latent space, as in the 2D VAE of Stable Diffusion [3, 35], while early video extensions inflated 2D convolutions to 3D [3]. Subsequent works developed fully spatio-temporal autoencoders with explicit temporal compression [32, 44, 54]. These methods can be broadly categorized into discrete and continuous autoencoders: discrete variants quantize features into a finite codebook of tokens via vector quantization [7, 42, 52, 55], while

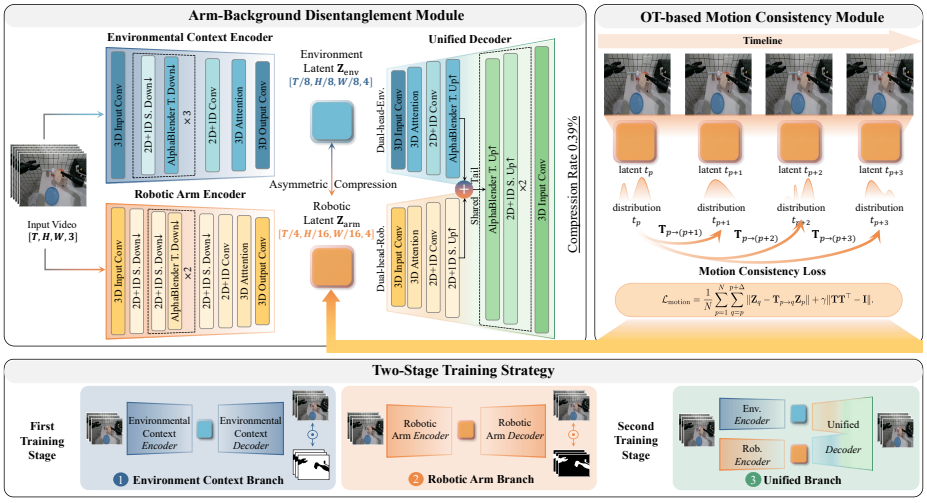

Fig. 2: The architecture of our EmbodiedVAE, a novel video VAE tailed for embodied manipulation. The Arm–Background Disentanglement Module adopts a dual-encoder, single-decoder architecture with asymmetric spatio-temporal compression, producing compact yet controllable latent representations. To ensure temporal motion coherence, the OT-based Motion Consistency Module further regularizes the robotic arm's latent. The entire EmbodiedVAE is optimized through a two-stage training strategy: the first stage learns the two encoders, while the second stage trains the unified decoder.

continuous variants learn a Gaussian latent space through differentiable sampling for smooth, end-to-end visual reconstruction and generation [1, 31, 38, 44, 54, 54].

2.2 High-Compression Video VAEs

Operating in a compact latent space allows LDMs to train and sample more efficiently, which requires a video autoencoder capable of strong compression while preserving essential semantics. Several recent works have explored this direction [8, 25, 39]. LTX-Video [17] adopts a video VAE with an aggressive downsampling ratio, whose decoder also performs the final diffusion denoising step. VidTwin [47] achieves a 0.20% compression rate by decoupling videos into structural and dynamic latent vectors, and Hi-VAE [25] further decomposes video dynamics into global motion and fine-grained spatial details. H3AE [51] investigates efficient architectures and introduces a latent loss that eliminates the need for complex discriminators. *However, existing video VAEs still struggle to balance compactness and controllability, limiting their ability to support precise action control in downstream robotic manipulation video generation.*

2.3 Robotic Manipulation Video Generation

Video generation models used as world simulators have advanced embodied learning by predicting future states in videos that capture realistic robot-object

interactions conditioned on past observations and actions. iVideoGPT [49] performs autoregressive, action-conditioned video prediction for embodied tasks, while VLP [12] and UniSim [53] integrate language and action cues to enhance semantic controllability in video generation. EVAC [21] introduces a multi-level action-conditioning mechanism with ray-map encoding to improve generalization across environments. IRASim [59] further incorporates a frame-level action-conditioning module within each transformer block to explicitly align actions with generated frames and strengthen temporal consistency. *However, most existing methods rely on image VAE latents, which limit the training efficiency, while video VAEs struggle to provide controllable latents.*

3 EmbodiedVAE

In this section, we elaborate on EmbodiedVAE, a high-compression video VAE that offers controllable latent representations tailored for embodied manipulation. As illustrated in Figure 2, our EmbodiedVAE is built upon a dual-encoder, single-decoder architecture with asymmetric compression rates. One encoder extracts foreground robotic arm features, while the other captures environmental context. A unified decoder then reconstructs the entire scene from the fused latent representations, optimized under our designed two-stage training strategy. To further enhance motion stability, we introduce an optimal-transport-based consistency loss that enforces temporal coherence in the latent space.

3.1 Arm-Background Disentanglement Module

Accurate action control is essential for robotic manipulation video generation, such as in robotic world models, where the system must precisely simulate and predict complex interactions among the robot, manipulated objects, and the surrounding environment [16,21,59]. However, most existing world models based on LDMs still rely on 2D image VAEs for latent representations [3, 59], since temporal compression in video VAEs often leads to motion information loss, making precise action control difficult. Although the 2D image VAEs effectively capture spatial structures, they fails to fully exploit temporal redundancy. As a result, the latent space remains insufficiently compact, leading to inefficient training and limited scalability for downstream LDMs.

To address the aforementioned challenges, our proposed EmbodiedVAE aims to learn a dedicated latent space for robotic arm motion, enabling the precise integration of downstream action-control information into the robotic arm representation while suppressing irrelevant backgrounds. To achieve this, EmbodiedVAE employs a dual-encoder, single-decoder architecture that automatically disentangles the robotic arm from its environmental surroundings without requiring explicit masks during inference. One encoder focuses on capturing fine-grained robotic arm dynamics, while the other encoder models the environmental context. This dual-encoder design produces two complementary latent representations, the robotic arm and the environment latent. The explicit robotic latent enables precise action injection while remaining unaffected by unrelated noise.

Foreground Robotic Arm Encoder The detailed network architecture is visualized in Figure 2. Following VidTok [38], we avoid using fully 3D architectures to reduce the computational overhead, while we only apply causal 3D convolutions in both the input and output layers to integrate spatio-temporal information and preserve temporal causality. In the penultimate layer, a 3D attention block is introduced to capture fine-grained spatio-temporal dependencies and enhance feature representation. Between the input and output stages, we stack multiple hybrid blocks composed of 2D spatial and causal 1D temporal downsampling modules, fused by the AlphaBlender operator [24]:

$$\mathbf{x}_{\text{down}} = \alpha \cdot \mathbf{x}_1 + (1 - \alpha) \cdot \mathbf{x}_2, \tag{1}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the intermediate downsampled features produced from the interpolation operation and the convolution operation branch, respectively. The blending weight α is a fixed hypermeter in practice, and we set $\alpha = \text{Sigmoid}(0.2)$ to balance the information flow from two branches here, and the $\mathbf{x}_{\text{down}}$ serves as the final downsampled representation as defined.

Since the robotic arm in embodied manipulation scenarios typically occupies only a small, albeit crucial, fraction of the visual field, our foreground robotic-arm encoder therefore applies a higher spatial compression rate to its feature representation to better match its limited spatial footprint, resulting in a more compact latent representation. Specifically, we stack four spatial downsampling blocks together with two temporal downsampling blocks, achieving an overall compression ratio of $4\times(16\times16)$ (temporal $\times$ spatial). We set the number of latent channels to 4, which further reduces spatial redundancy while preserving the most informative arm-centric details, yielding a compact and efficient robotic-arm latent representation for downstream modeling and control.

Background Environmental Context Encoder For the background environmental context encoder, we adopt an architecture similar to the foreground robotic arm encoder. However, in embodied manipulation scenarios, the first-person perspective introduces substantial temporal redundancy in the background, which we leverage by applying a more aggressive temporal compression strategy. Specifically, we stack three spatial with three temporal downsampling blocks to acheive a $8\times(8\times8)$ (temporal $\times$ spatial) compression and we also set the latent channels to 4, effectively reducing the temporal redundancy and yielding a temporal compact environment latent representation.

Unified Reconstruction Decoder As illustrated in Figure 2, the decoder consists of two components: a dual-head module and a shared tail module. In the dual-head module, the robotic arm and environment latent representations are upsampled independently. The arm head performs stronger spatial upsampling, whereas the background head emphasizes temporal upsampling. When both latent representations reach the same resolution, we concatenate them along the channel dimension and feed the merged latent into the shared tail decoder to

reconstruct the complete video. The whole workflow can be formulated as:

$$\hat{\mathbf{X}} = D(\mathbf{Z}_{\text{arm}}, \mathbf{Z}_{\text{env}}), \tag{2}$$

$$\mathbf{Z}_{\text{arm}} = R(E_{\text{arm}}(\mathbf{X})), \quad \mathbf{Z}_{\text{env}} = R(E_{\text{env}}(\mathbf{X})). \tag{3}$$

Here, $\hat{\mathbf{X}}$ and $\mathbf{X}$ denote the reconstructed and input videos, respectively, and $\mathbf{Z}_{arm}, \mathbf{Z}_{env}$ denotes the two latent representations. $D(\cdot), E_{\text{arm}}(\cdot)$ and $E_{\text{env}}(\cdot)$ represent the decoder and the two encoders, while $R(\cdot)$ denotes the latent regularizer. To mirror the architectures of the robotic-arm encoder and the environmental-context encoder, the unified decoder stacks multiple 2D and 1D upsampling blocks integrated with the AlphaBlender operator defined in Eq. (1); the input and output layers use 3D convolutions, and the second layer incorporates a 3D attention block to ensure architectural completeness and consistency of VAE.

3.2 OT-based Motion Consistency Module

The motion of the robotic arm carries the essential information in embodied manipulation scenarios, reflecting the interactions between the embodied agent and its environment in a fine-grained, time-varying manner. However, the temporal compression used in existing video VAEs often causes motion details to be lost during the latent encoding and reconstruction processes. This loss of motion information in the latent space impairs downstream robotic arm action control that relies on these latents for accurate, consistent decision making.

To maximize the preservation of motion information in the resulting compact latents, we introduce an Optimal-Transport-based Motion Consistency Module. Our design is inspired by the observation that stable motion patterns in videos maintain visual invariance [45], meaning that visual correspondences between patches remain coherent across consecutive frames. Such temporal visual invariance reflects the intrinsic stability of motion dynamics. Our module leverages this property to encourage the latent representations to remain motion consistent over time by minimizing the information transport cost across latent frames. This constraint is applied to the latent space of the robotic arm encoder, guiding it to stabilize capture temporally coherent motion dynamics, since its corresponding decoder is discarded after training as introduced in Section 3.3.

Optimal Transport (OT) is a mathematical framework to measure how much effort it takes to transform one distribution into another [36], and has proven effective in vision tasks such as semantic correspondence [26] and visual place recognition [19] by facilitating coherent alignment of feature distributions. Specifically, let the latent distributions at timestamp t_p and t_q be μ_p, μ_q over the latent space $\mathcal{Z}$. The Kantorovich optimal transport plan [36] between distribution μ_p and distribution μ_q is a coupling $\pi_{p\to q} \in \Pi(\mu_p, \mu_q)$ that minimizes the following expected transport cost under the standard OT formulation:

$$\pi^\star = \arg\min_{\pi \in \Pi(\mu_p, \mu_q)} \int c(z_p, z_q) d\pi(z_p, z_q), \tag{4}$$

where $c(\cdot,\cdot)$ denotes the transport cost and $\Pi(\mu_p, \mu_q)$ is the joint distribution. Due to the discrete nature of video pixels, here we model the latent distribution as discrete probability distributions: $\mu_p = \sum_{i=1}^{N} a_i \delta_{x_i}$ and $\mu_q = \sum_{j=1}^{N} b_j \delta_{y_j}$, where δ_{x_i} denotes the Dirac measure [13] at x_i, satisfying $\int f(x)\delta_{x_i}(x)dx = f(x_i)$, and $\mathcal{A} = \{a_i\}_i^N, \mathcal{B} = \{b_j\}_j^N$ represent the corresponding probability masses. As the similarity between two latent representations roughly encodes the likelihood of content correspondence across time, consequently, we design the cost function $c(\cdot,\cdot)$ as the negative similarity between these two latent slices:

$$c(i,j) = -\mathbf{S}_{i,j} = \frac{\langle \mathbf{Z}_p[i], \mathbf{Z}_q[j] \rangle}{\sqrt{d}} \tag{5}$$

where $\mathbf{Z}_p[i]$ denotes the latent feature at position i and d is the channel dimension. Building upon this, and following [10], we can reformulate the optimal transport problem in Eq. (4) into an entropy-regularized formulation:

$$\begin{aligned} \mathbf{T}^\star = \arg\min_{\mathbf{T}} \sum_{ij} \mathbf{T}_{ij} c(i,j) + \epsilon \mathcal{H}(\mathbf{T}), \\ \text{s.t.} \quad \mathbf{T}\mathbf{1} = \mu_p, \mathbf{T}^\top \mathbf{1} = \mu_q \end{aligned} \tag{6}$$

where $\mathcal{H}(\mathbf{T}) = -\sum_{ij} \mathbf{T}_{ij}(\log \mathbf{T}_{ij} - 1)$ denotes the entropy regularizer, and ε controls the strength of regularization during optimization. The matrix $\mathbf{T}$ represents the discrete optimal transport plan that minimizes the information transport cost between two latent distributions, serving as the discrete counterpart of $\pi^\star$ in this setting. We compute $\mathbf{T}$ using the Sinkhorn-Knopp algorithm [10], a differentiable and efficient approximation to the Hungarian algorithm [30] in practice. After obtaining $\mathbf{T}$, we define the motion consistency loss to enforce temporal stability and visual invariance under smooth motion as follows:

$$\mathcal{L}_{\text{motion}} = \frac{1}{N} \sum_{p=1}^{N} \sum_{q=p}^{p+\Delta} \|\mathbf{Z}_q - \mathbf{T}_{p\to q}\mathbf{Z}_p\| + \gamma \|\mathbf{T}\mathbf{T}^\top - \mathbf{I}\|. \tag{7}$$

Here, Δ controls the temporal window for alignment, and γ balances the permutation regularizer $\|\mathbf{T}\mathbf{T}^\top - \mathbf{I}\|$, which regularize $\mathbf{T}$ to approximate a one-to-one correspondence as closely as possible. $\mathcal{L}_{\text{motion}}$ promotes temporal consistency by minimizing feature misalignment guided by the optimal transport plan computed above. To further analyze the behavior of $\mathcal{L}_{\text{motion}}$, we present the following proposition, with details provided in Appendix A for completeness.

Proposition 1. *Let $\hat{\mathbf{T}}^\star$ denote the OT plan obtained by the Sinkhorn–Knopp algorithm after row normalization. Denote $j_i^\star = \arg\max_j \mathbf{S}_{ij}$ and define the margin $\gamma_i = \mathbf{S}_{ij_i^\star} - \max_{j \neq j_i^\star} \mathbf{S}_{ij}$, we have $\hat{\mathbf{T}}^\star_{ij^\star} \geq \frac{1}{1+ae^{-\gamma_i/\epsilon}}$ with a constant a.*

Proposition 1 implies that when latent features exhibit clear separability across frames and the motion remains temporally smooth, the transport matrix $\mathbf{T}$ will converge toward a nearly permutation-like structure, reflecting a strong motion temporal consistency. In contrast to a similarity matrix that encodes only *local*

pairwise affinities, the optimal transport plan $\mathbf{T}$ establishes a *global* alignment between distributions, capturing coherent many-to-many correspondences that account for all pairwise relations across time [10, 19] more effectively.

3.3 Two-stage Training Strategy

The specific training strategy of our proposed novel EmbodiedVAE consists of two training stages. In the first training stage, the two encoders are obtained, while in the second training stage, we train the unified decoder.

First training stage. In the first stage, our objective is to train the robotic arm encoder to capture fine-grained motion of the manipulator, while the environmental encoder preserves scene context. To achieve this, we first generate robotic arm masks using SAM2 [33] guided by Grounding DINO [34] with the text prompt "robotic arm". It is worth noting that these masks are used only as auxiliary supervision during training; once trained, EmbodiedVAE can automatically disentangle motion and context information without relying on masks. The training target loss for the robotic arm is defined as:

$$\mathcal{L}_{\text{arm}} = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{motion}} + \beta\mathcal{L}_{\text{KL}} + \lambda\mathcal{L}_{\text{aux}}, \tag{8}$$

where the reconstruction loss $\mathcal{L}_{\text{rec}} = \|(\hat{\mathbf{X}} - \mathbf{X}) \odot \mathbf{M})\|$, and $\mathbf{M} \in \{0,1\}^{T\times H\times W}$ is the pre-built binary robotic arm mask. $\mathcal{L}_{\text{KL}}$ denotes the KL divergence loss term that defined as $\mathcal{L}_{\text{KL}} = \text{KL}(\mathcal{N}(\mathbf{u}_{\mathbf{z}}, \boldsymbol{\sigma}_{\mathbf{z}})\|\mathcal{N}(\mathbf{0}, \mathbf{I}))$, and $\mathcal{L}_{\text{aux}}$ represents the auxiliary loss term which we further combine from the standard adversarial loss and the feature-level perceptual loss [38]. α, β, λ is the hyperparameters. As for the training of the environmental context encoder, the overall loss is defined as:

$$\mathcal{L}_{\text{env}} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}} + \lambda\mathcal{L}_{\text{aux}}, \tag{9}$$

where the reconstruction loss $\mathcal{L}_{\text{rec}} = \|(\hat{\mathbf{X}} - \mathbf{X}) \odot (1 - \mathbf{M})\|$ and the other loss terms are the same as those used for training the robotic arm encoder in Eq. (8).

Second training stage After the first training stage, we freeze the two pre-trained encoders from the robotic-arm and environment-context branches with a very low learning rate while discarding their decoders. We then train the unified decoder, which is introduced in Section 3.1, with the following loss:

$$\mathcal{L}_{\text{unified}} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}} + \lambda\mathcal{L}_{\text{aux}}, \tag{10}$$

where the reconstruction loss term$\mathcal{L}_{\text{rec}} = \|\hat{\mathbf{X}} - \mathbf{X}\|$, and the remaining terms are identical to the first training stage introduced in Section 3.3.

3.4 Action-controlled Video Generation

Our EmbodiedVAE can be seamlessly integrated with most existing LDMs, enabling the efficient and controllable downstream robotic action-controlled video generation. To demonstrate this integration, we adopt IRASim [59], a DiT-style LDM that functions as the robotic manipulation world model, as an example in our experiments. In this setup, the latent representations of the robotic arm and the environmental context, $\mathbf{Z}_{\text{arm}}$ and $\mathbf{Z}_{\text{env}}$, are first converted into two sequences of tokens with the same hidden dimension. Both latent representations are processed through the same diffusion procedure using shared DiT parameters, *ensuring that no additional computational overhead is introduced.* To facilitate effective interaction between the two complementary latent streams, we insert a cross-attention layer after every k transformer blocks, where $k = 2$. This design enables adaptive information exchange between the arm and background representations. For example, the arm motion can be adjusted based on nearby objects, while the environment representation accounts for motion-induced changes such as lighting and shadows. This interaction results in globally consistent scene dynamics and locally accurate motion generation.

4 Experiments

4.1 Experiments Setup

Datasets. We train our proposed EmbodiedVAE on a self-collected dataset of approximately one million robotic manipulation videos, aggregated from RobNet [11], RobSet [2], BC-Z [20], RH20T [14], and DROID [22], covering diverse manipulation behaviors across multiple environments. Robotic arm masks are automatically generated for each video using SAM2 [33] and Grounding DINO [34] with the text prompt "robotic arm". For evaluation, we use Agibot-2025 [21] and Bridge [43] for video VAE reconstruction tasks, and RT-1 [5] and Bridge [43] for action-conditioned robotic manipulation video generation.
Baselines. (1) *For reconstruction task evaluation*, we compare our EmbodiedVAE with several state-of-the-art high-compression video VAEs that also achieve

Table 1: Quality comparison with eight baselines on the **video VAE reconstruction task**, evaluated on two datasets using four metrics. The best results are highlighted in **bold**, and the second-best results are underlined.

Datasets		Agibot-2025 [21]				Bridge [43]			
Method	Com. Rate↓	PSNR↑	LPIPS↓	SSIM↑	FVD↓	PSNR↑	LPIPS↓	SSIM↑	FVD↓
OpenSoraPlan [24]	1.04%	30.7553	<u>0.0834</u>	<u>0.9257</u>	**290.2550**	30.3755	<u>0.1098</u>	0.8948	<u>340.5284</u>
Cosmos [1]	2.08%	28.2329	0.1520	0.8778	889.7757	27.8812	0.1916	0.8386	715.6345
iVideoGPT [49]	1.50%	29.2503	0.1060	0.9096	760.6489	28.3325	0.1447	0.8738	522.1818
MAGVIT-v2 [55]	0.65%	27.1014	0.2203	0.7197	849.5063	26.8635	0.2396	0.6986	549.9939
Vidtwin [47]	**0.20%**	30.8263	0.1216	0.9202	443.1976	29.7386	0.1679	0.8531	585.8065
EMU-3 [46]	0.53%	28.9081	0.1064	0.8926	599.8921	27.5814	0.1517	0.8317	736.5338
CV-VAE [58]	0.53%	30.2767	0.1018	0.9128	392.6846	29.8419	0.1330	0.8789	418.8394
CMD [56]	6.85%	<u>31.4030</u>	0.1086	0.8995	439.9010	**31.3764**	0.1115	<u>0.8996</u>	358.6915
EmbodiedVAE	<u>0.39%</u>	**31.6745**	**0.0723**	**0.9345**	<u>368.5511</u>	<u>30.6957</u>	**0.1060**	**0.9017**	**309.7604**

good reconstruction quality, including OpenSoraPlan [24], MAGVIT-v2 [55], and CV-VAE [58] with continuous latents; EMU-3 [46] and Cosmos [1] as discrete token-based models; and approaches that decouple videos into multiple latent spaces, such as CMD [56], iVideoGPT [49], and VidTwin [47]. (2) *For downstream video generation in action-controlled robotic manipulation*, we compare EmbodiedVAE with several representative VAEs. Specifically, we include high-fidelity video VAEs such as Wan-VAE [44] and Cog-VAE [54]; high-compression VAEs including OpenSoraPlan [24], EMU-3 [46], and CV-VAE [58]; and models that decouple videos into multiple latent components, such as VidTwin [47] and CMD [56]. In addition, we evaluate the 2D image VAE SDXL [31] from Stable Diffusion, a strong baseline that encodes videos frame by frame, retaining full temporal information without temporal compression by a non-compact latent.

Metrics. Following [38,59], we evaluate all methods on video reconstruction and action-conditioned video generation for robotic manipulation using PSNR [18], SSIM [48], LPIPS [57], and FVD [40]. Efficiency is assessed by the VAE compression rate [35,38], defined as the ratio of latent to original video dimensions.

Implements. We train EmbodiedVAE on 8 fps, 16-frame video clips at a resolution of 256×256, and evaluate it on 16 fps, 16-frame clips of the same resolution for reconstruction tasks. The model is trained using 8 NVIDIA A800 GPUs.

4.2 VAE Reconstruction Quality

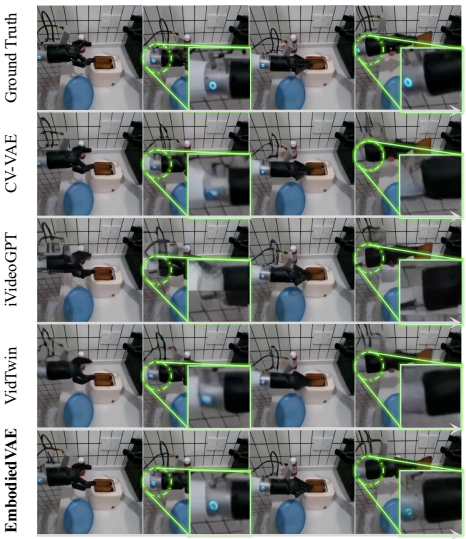

Fig. 3: Comparison of the **video VAE reconstruction quality** between the proposed EmbodiedVAE and existing representative baseline methods. Results are shown from the Agibot-2025 [21] dataset.

In the video VAE reconstruction experiments, we comprehensively compare the proposed EmbodiedVAE with all baseline methods in terms of both reconstruction quality and compression rate, and the results are summarized in the Table 1. As we can seen, the proposed EmbodiedVAE achieves the best overall reconstruction performance across both datasets (except for PSNR on the Bridge dataset) while maintaining a competitive compression rate of 0.39%. In particular, EmbodiedVAE achieves the highest PSNR of 31.6745 in Agibot-2025, while its compression rate is nearly 1/20 of that of the runner-up baseline CMD, demonstrating superior reconstruction fidelity under high compression rate. We further present a qualitative example in Figure 3, where EmbodiedVAE produces substantially clearer reconstructions of robotic arm details with a high compression rate compared with baselines. In

Table 2: Quality comparison with seven baselines on the **downstream action-controlled video generation in robotic manipulation scenarios** using four metrics. The best results are highlighted in **bold**, the second-best results are underlined, and value* denotes results obtained with the image VAE baseline used during both training and inference, which is *without any temporal compression*.

Datasets		RT-1 [5]				Bridge [43]			
Method	Com. Rate↓	PSNR↑	LPIPS↓	SSIM↑	FVD↓	PSNR↑	LPIPS↓	SSIM↑	FVD↓
SDXL [31]	8.32%	23.1988*	0.1901	0.8074	761.7837*	22.3624*	0.1753	0.7915	**455.7345***
OpenSoraPlan [24]	1.04%	21.6494	0.2486	0.7432	958.0784	21.0405	0.2376	0.7577	785.5559
Emu3 [46]	0.53%	21.6749	0.2456	0.7286	1049.2935	20.8573	0.2506	0.7053	986.3582
CV-VAE [58]	0.53%	20.3296	0.3047	0.6859	2529.3193	19.3068	0.3544	0.6568	1829.7292
CMD [56]	6.85%	20.9454	0.2560	0.7671	2644.7208	20.1799	0.2521	0.7419	1464.6843
Vidtwin [47]	**0.20%**	21.1037	0.3710	0.6697	1824.3383	20.4466	0.3438	0.6889	1690.1818
Cog-VAE [54]	2.08%	21.2231	0.2174	0.7907	1135.6917	21.6951	0.2095	0.7807	1302.4434
Wan-VAE [44]	2.45%	22.7334	0.1848	0.8138	802.2071	22.1690	0.1601	0.8211	548.9621
EmbodiedVAE	0.39%	**24.7082**	**0.1707**	**0.8263**	**716.3273**	**24.2484**	**0.1409**	**0.8366**	631.6450

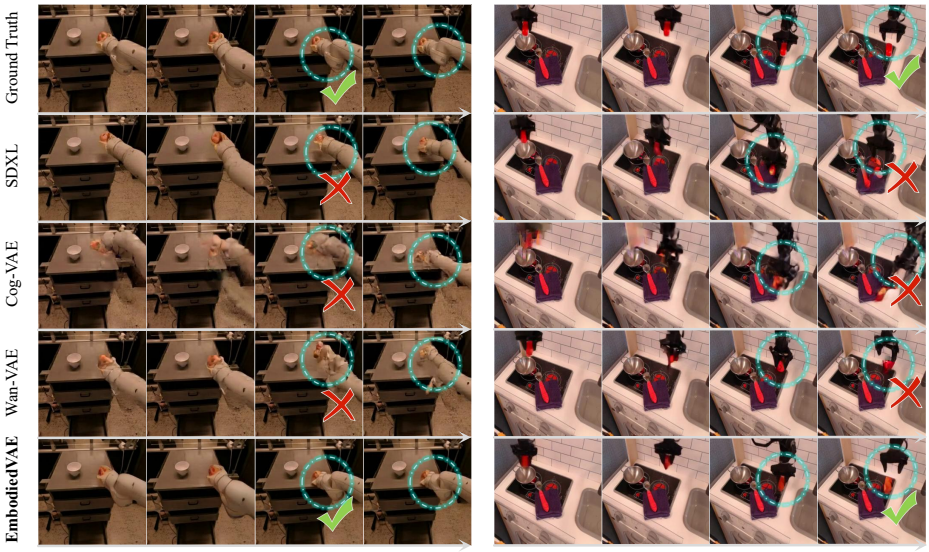

Fig. 4: Qualitative comparisons of **action-conditioned world models for robotic manipulation** trained under the same setup, where the only difference lies in the VAE used for latent representations: our EmbodiedVAE versus representative video VAEs. Results on the left are from RT-1 [5], and results on the right are from Bridge [43].

particular, baselines such as VidTwin fail to preserve fine-grained robotic arm visual cues like electrical components, whereas our EmbodiedVAE successfully reconstructs these critical details.

4.3 *Controllable Studies*: Robotic Manipulation World Model

We evaluate EmbodiedVAE on action-conditioned video prediction for robotic manipulation using IRASim [59], with the LDM backbone fixed to IRASim-L

(461M) and only the VAE component varied during training and inference (Details are in Appendix C). As shown in Table 2, our EmbodiedVAE significantly outperforms the widely recognized Wan-VAE and Cog-VAE on this task, achieving a notable improvement of 2.02 PSNR over the runner-up video VAE (Wan-VAE). It is worth noting that EmbodiedVAE even surpasses the SDXL [31], a powerful image-based VAE without temporal information compression. Example analysis in Figure 4 further show that EmbodiedVAE can accurately predict robot-environment interactions, while competing video VAEs (e.g., Wan-VAE) often produce visually plausible yet physically inconsistent motions. These advantages are attributed to our disentangled design, which enables precise action injection without interference from irrelevant factors, and the consistency loss further preserves the motion dynamics despite time compression.

4.4 Ablation Studies

We conduct an ablation study to evaluate the effectiveness of the proposed OT-based motion consistency loss $\mathcal{L}_{\text{motion}}$ in both the video VAE reconstruction task and the downstream action-controlled video prediction task. As shown in Table 3, this module consistently improves both tasks, with larger gains in the downstream manipulation setting.

Table 3: Ablation studies on both the video VAE reconstruction ("Recon.") task and the downstream action-controlled video prediction for robotic manipulation task ("Manip.").

Task	Methods	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
Recon.	Datasets	Agibot-2025 [21]		Bridge [43]	
	EmbodiedVAE	31.6745	0.0723	30.6957	0.1060
	w/o Mask	*31.1812*	*0.0728*	*29.6680*	*0.1211*
	w/o $\mathcal{L}_{\text{motion}}$	*31.0213*	*0.0835*	*29.9769*	*0.1087*
Manip.	Datasets	RT-1 [5]		Bridge [43]	
	EmbodiedVAE	24.7082	0.1707	24.2484	0.1409
	w/o Mask	*23.8044*	*0.1904*	*23.7306*	*0.1705*
	w/o $\mathcal{L}_{\text{motion}}$	*23.8187*	*0.1863*	*23.9726*	*0.1643*

4.5 Disentangle Studies

We visualize the decoded outputs of each branch in Figure 5. As observed, the arm branch captures the robotic motion, while the environment branch models the global scene structure, resulting in disentangled latent representations that facilitate efficient and controllable downstream tasks.

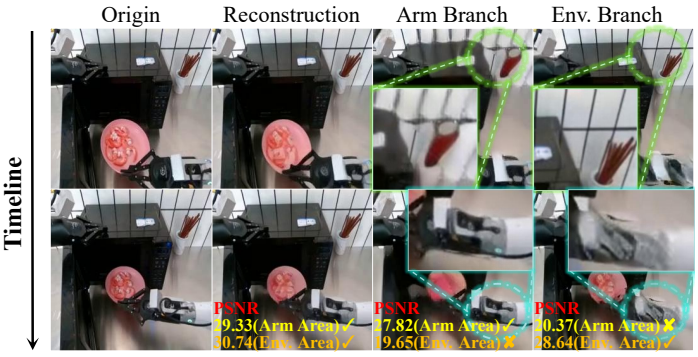

Fig. 5: Disentanglement analysis on the Agibot-2025 [21] dataset.

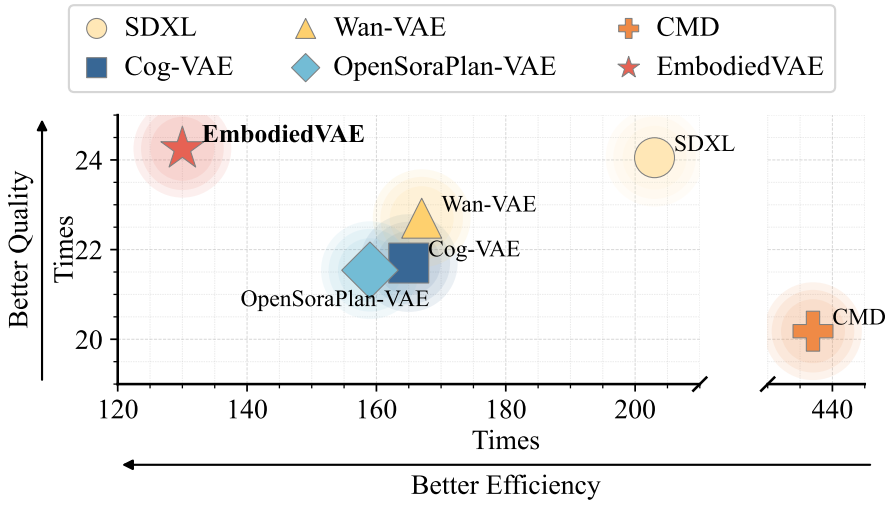

Fig. 6: Comparison of our proposed EmbodiedVAE and baselines on downstream training efficiency using the Bridge [43] dataset. The reported time corresponds to 1000 training steps with pre-encoded latents [54] from 256 × 256, 16-frame videos(17 for Wan-VAE due to its characteristics).

4.6 *Efficiency Studies*: Downstream LDM Training Time Analysis

To investigate the efficiency benefits of the compact latents introduced by EmbodiedVAE in downstream LDM training, we measure the training time and memory consumption of our model in the downstream robotic manipulation world model, also using IRASim-L [59] as an example. The results are reported in Figure 6. The reported values are measured after the pre-encoding operation, where the latents of the training data are precomputed following previous works [54, 59]. As observed, the compact latents produced by our EmbodiedVAE effectively reduce resource consumption during downstream training.

5 Conclusion

We present EmbodiedVAE, a high-compression and controllable video VAE tailored for embodied manipulation. To enable efficient and high-quality downstream action control, EmbodiedVAE disentangles videos into two complementary latents via a dual-encoder, single-decoder architecture with asymmetric compression. An OT-based motion consistency module further encourages coherent robotic motion preservation. The resulting arm and environment latents enable precise condition injection. Extensive experiments on video reconstruction and action-conditioned manipulation show the effectiveness of EmbodiedVAE.

Acknowledgements

The corresponding authors are Jianxin Li and Zhibo Chen. This work was supported by the National Natural Science Foundation of China under Grant No. 62225202, and the Zhongguancun Academy Project under Grant No. C20250302.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv (2025)
2. Bharadhwaj, H., Vakil, J., Sharma, M., Gupta, A., Tulsiani, S., Kumar, V.: Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In: ICRA. pp. 4788–4795. IEEE (2024)
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv (2023)
4. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR. pp. 22563–22575 (2023)
5. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. RSS (2022)
6. Bu, Q., Cai, J., Chen, L., Cui, X., Ding, Y., Feng, S., Gao, S., He, X., Hu, X., Huang, X., et al.: Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. arXiv (2025)
7. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: CVPR. pp. 11315–11325 (2022)
8. Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Li, M., Lu, Y., Han, S.: Deep compression autoencoder for efficient high-resolution diffusion models. ICLR (2025)
9. Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al.: Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. arXiv (2025)
10. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Neurips **26** (2013)
11. Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., Finn, C.: Robonet: Large-scale multi-robot learning. CORL (2019)
12. Du, Y., Yang, M., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J.B., Kaelbling, L., Zeng, A., Tompson, J.: Video language planning. arXiv (2023)
13. Erbar, M., Rumpf, M., Schmitzer, B., Simon, S.: Computation of optimal transport on discrete metric measure spaces. Numerische Mathematik **144**(1), 157–200 (2020)
14. Fang, H.S., Fang, H., Tang, Z., Liu, J., Wang, C., Wang, J., Zhu, H., Lu, C.: Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. RSS (2023)
15. Gao, C., Jin, L., Peng, X., Zhang, J., Deng, Y., Li, A., Wang, H., Liu, S.: Octonav: Towards generalist embodied navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 40074–40084 (2026)

16. Gao, C., Liu, S., Chen, J., Wang, L., Wu, Q., Li, B., Tian, Q.: Room-object entity prompting and reasoning for embodied referring expression. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(2), 994–1010 (2023)
17. HaCohen, Y., Chiprut, N., Brazowski, B., Shalem, D., Moshe, D., Richardson, E., Levin, E., Shiran, G., Zabari, N., Gordon, O., et al.: Ltx-video: Realtime video latent diffusion. arXiv (2024)
18. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: ICPR. pp. 2366–2369. IEEE (2010)
19. Izquierdo, S., Civera, J.: Optimal transport aggregation for visual place recognition. In: CVPR. pp. 17658–17668 (2024)
20. Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., Finn, C.: Bc-z: Zero-shot task generalization with robotic imitation learning. In: CoRL. pp. 991–1002. PMLR (2022)
21. Jiang, Y., Chen, S., Huang, S., Chen, L., Zhou, P., Liao, Y., He, X., Liu, C., Li, H., Yao, M., et al.: Enerverse-ac: Envisioning embodied environments with action condition. arXiv (2025)
22. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., et al.: Droid: A large-scale in-the-wild robot manipulation dataset. RSS (2024)
23. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. CoRL (2024)
24. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. arXiv (2024)
25. Liu, H., Sun, W., Zhang, Q., Di, D., Gong, B., Li, H., Wei, C., Zou, C.: Hi-vae: Efficient video autoencoding with global and detailed motion. arXiv (2025)
26. Liu, Y., Zhu, L., Yamada, M., Yang, Y.: Semantic correspondence as an optimal transport problem. In: CVPR. pp. 4463–4472 (2020)
27. Luo, J., Chen, J., Wang, J., Wang, C., Zhu, H., Sun, Q., Gao, C., Chen, Z., Li, J.: Attention sparsity is input-stable: Training-free sparse attention for video generation via offline sparsity profiling and online qk co-clustering. ICML (2026)
28. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. Neurips (2021)
29. Mu, Y., Chen, T., Chen, Z., Peng, S., Lan, Z., Gao, Z., Liang, Z., Yu, Q., Zou, Y., Xu, M., et al.: Robotwin: Dual-arm robot benchmark with generative digital twins. In: CVPR. pp. 27649–27660 (2025)
30. Munkres, J.: Algorithms for the assignment and transportation problems. SIAM **5**(1), 32–38 (1957)
31. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. ICLR (2024)
32. Qin, C., Xia, C., Ramakrishnan, K., Ryoo, M., Tu, L., Feng, Y., Shu, M., Zhou, H., Awadalla, A., Wang, J., et al.: xgen-videosyn-1: High-fidelity text-to-video synthesis with compressed representations. In: ECCV. pp. 249–265. Springer (2024)
33. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. ICLR (2025)

34. Ren, T., Jiang, Q., Liu, S., Zeng, Z., Liu, W., Gao, H., Huang, H., Ma, Z., Jiang, X., Chen, Y., et al.: Grounding dino 1.5: Advance the" edge" of open-set object detection. arXiv (2024)
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
36. Santambrogio, F.: Optimal transport for applied mathematicians: calculus of variations, pdes, and modeling. PNDA **87** (2015)
37. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
38. Tang, A., He, T., Guo, J., Cheng, X., Song, L., Bian, J.: Vidtok: A versatile and open-source video tokenizer. arXiv (2024)
39. Tian, R., Dai, Q., Bao, J., Qiu, K., Yang, Y., Luo, C., Wu, Z., Jiang, Y.G.: Reducio! generating 1k video within 16 seconds using extremely compressed motion latents. ICCV (2024)
40. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv (2018)
41. Vahdat, A., Kautz, J.: Nvae: A deep hierarchical variational autoencoder. Neurips **33**, 19667–19679 (2020)
42. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Neurips **30** (2017)
43. Walke, H.R., Black, K., Zhao, T.Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A.W., Myers, V., Kim, M.J., Du, M., et al.: Bridgedata v2: A dataset for robot learning at scale. In: CoRL. pp. 1723–1736. PMLR (2023)
44. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv (2025)
45. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: CVPR. pp. 2566–2576 (2019)
46. Wang, X., Zhang, X., Luo, Z., Sun, Q., Cui, Y., Wang, J., Zhang, F., Wang, Y., Li, Z., Yu, Q., et al.: Emu3: Next-token prediction is all you need. arXiv (2024)
47. Wang, Y., Guo, J., Xie, X., He, T., Sun, X., Bian, J.: Vidtwin: Video vae with decoupled structure and dynamics. In: CVPR. pp. 22922–22932 (2025)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004)
49. Wu, J., Yin, S., Feng, N., He, X., Li, D., Hao, J., Long, M.: ivideogpt: Interactive videogpts are scalable world models. Neurips **37**, 68082–68119 (2024)
50. Wu, P., Zhu, K., Liu, Y., Zhao, L., Zhai, W., Cao, Y., Zha, Z.J.: Improved video vae for latent video diffusion model. In: CVPR. pp. 18124–18133 (2025)
51. Wu, Y., Li, Y., Skorokhodov, I., Kag, A., Menapace, W., Girish, S., Siarohin, A., Wang, Y., Tulyakov, S.: H3ae: High compression, high speed, and high quality autoencoder for video diffusion models. arXiv (2025)
52. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv (2021)
53. Yang, M., Du, Y., Ghasemipour, K., Tompson, J., Schuurmans, D., Abbeel, P.: Learning interactive real-world simulators. Neurips (2023)
54. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. ICLR (2025)

55. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Birodkar, V., Gupta, A., Gu, X., et al.: Language model beats diffusion–tokenizer is key to visual generation. ICLR (2023)
56. Yu, S., Nie, W., Huang, D.A., Li, B., Shin, J., Anandkumar, A.: Efficient video diffusion models via content-frame motion-latent decomposition. In: ICLR (2024)
57. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
58. Zhao, S., Zhang, Y., Cun, X., Yang, S., Niu, M., Li, X., Hu, W., Shan, Y.: Cv-vae: A compatible video vae for latent generative video models. Neurips **37**, 12847–12871 (2024)
59. Zhu, F., Wu, H., Guo, S., Liu, Y., Cheang, C., Kong, T.: Irasim: A fine-grained world model for robot manipulation. In: ICCV. pp. 9834–9844 (2025)

A Proof

To demonstrate the proposed Proposition 1, we first introduce the Sinkhorn–Knopp algorithm.

A.1 Preliminary: Sinkhorn-Knopp Algorithm

The Sinkhorn-Knopp algorithm is a classical method for efficiently solving discrete optimization problems, and is particularly well-suited for computing the optimal transport (OT) distance between two discrete probability distributions. In the context of OT, [10] introduced an entropy-regularized formulation that enables fast and stable computation. Specifically, entropy regularization smooths the feasible domain of the original OT problem, transforming it into a differentiable matrix-scaling (or permutation) problem. The resulting objective can then be efficiently optimized via the Sinkhorn iterative normalization procedure, which yields an approximate yet accurate solution to the original OT objective. Formally, the entropy-regularized OT problem reformulates as:

$$\begin{aligned} \mathbf{T}^\star = \arg\min_{\mathbf{T}} \sum_{ij} \mathbf{T}_{ij} c(i,j) + \epsilon \mathcal{H}(\mathbf{T}), \\ \text{s.t.} \quad \mathbf{T}\mathbf{1} = \boldsymbol{\mu}_p, \quad \mathbf{T}^\top \mathbf{1} = \boldsymbol{\mu}_q, \end{aligned} \tag{1}$$

where $\mathbf{T}$ denotes the transport matrix between the source distribution $\boldsymbol{\mu}_p$ and the target distribution $\boldsymbol{\mu}_q$, $c(i,j)$ represents the transport cost between elements i and j, ϵ is the entropy regularization coefficient, and $\mathcal{H}(\mathbf{T})$ denotes entropy term encouraging smoothness in the transport plan. From the perspective of probability and information theory, a uniform distribution has the maximum entropy. Therefore, entropy regularization will make the optimal transfer matrix $\mathbf{T}^\star$ closer to a uniform distribution. Then Eq. (1) can be solved by adopting the Sinkhorn-Knopp Algorithm.

Sinkhorn iterative solution. Suppose we have two marginal distributions represented by vectors $\boldsymbol{a}_u \in \mathbb{R}^{d_1}$ and $\boldsymbol{a}_v \in \mathbb{R}^{d_2}$. We initialize them as:

$$\boldsymbol{a}_u^{(0)} = \frac{\mathbf{1}_{d_1}}{d_1}, \boldsymbol{a}_v^{(0)} = \frac{\mathbf{1}_{d_2}}{d_2}, \tag{2}$$

where $\mathbf{1}_d$ denotes a d-dimensional all-ones vector. The Sinkhorn iteration alternately updates the scaling vectors as:

$$\boldsymbol{a}_u^{k+1} \leftarrow \frac{\boldsymbol{\mu}_u}{\mathbf{K} \cdot \boldsymbol{a}_u^k}, \boldsymbol{a}_v^{k+1} \leftarrow \frac{\boldsymbol{\mu}_v}{\mathbf{K}^\top \cdot \boldsymbol{a}_u^k}, \tag{3}$$

where the kernel matrix $\mathbf{K}$ is defined as $\mathbf{K}_{ij} = e^{-\lambda \mathbf{C}_{ij}}$, and $\mathbf{C}$ is the transport cost matrix with entries $\mathbf{C}_{ij}$ measuring the pairwise cost between the i-th element of the source distribution and the j-th element of the target distribution. After convergence, the optimal transport plan is obtained as:

$$\mathbf{T}^\star = \mathrm{diag}(\boldsymbol{a}_u)\mathbf{K}\mathrm{diag}(\mathbf{a}_v), \tag{4}$$

and the corresponding minimal transport distance (i.e., the optimal transport cost) is:

$$\mathbf{D}^\star = \langle \mathbf{T}^\star, \mathbf{C} \rangle, \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product, i.e., $\langle \mathbf{T}^\star, \mathbf{C} \rangle = \sum_{ij} \mathbf{T}^\star_{ij} \mathbf{C}_{ij}$.

A.2 Proof of Proposition 1

Here, we first restate the proposition as follows:

Proposition 1. *Let $\hat{\mathbf{T}}^\star$ denote the OT plan obtained by the Sinkhorn–Knopp algorithm after row normalization. Denote $j_i^\star = \arg\max_j \mathbf{S}_{ij}$ and define the margin $\gamma_i = \mathbf{S}_{ij_i^\star} - \max_{j \neq j_i^\star} \mathbf{S}_{ij}$, we have $\hat{\mathbf{T}}^\star_{ij^\star} \geq \frac{1}{1+ae^{-\gamma_i/\epsilon}}$ with a constant a.*

Proof. For the row-normalized OT plan $\hat{\mathbf{T}}^\star$, we have $\sum_j \hat{\mathbf{T}}^\star_{ij} = 1$. According to Eq. (4), the following holds:

$$\begin{aligned}
\hat{\mathbf{T}}^\star_{ij_i^\star} &= \frac{\hat{\mathbf{T}}^\star_{ij_i^\star}}{\sum_j \hat{\mathbf{T}}^\star_{ij}} && (6)\\
&= \frac{(\boldsymbol{\mu}_v)_{j_i^\star} e^{\mathbf{S}_{ij_i^\star}/\varepsilon}}{\sum_j (\boldsymbol{\mu}_v)_j e^{\mathbf{S}_{ij}/\varepsilon}} && (7)\\
&= \frac{1}{1 + \sum_{j \neq j_i^\star} \frac{(\boldsymbol{\mu}_v)_j}{(\boldsymbol{\mu}_v)_{j_i^\star}} e^{-(\mathbf{S}_{ij_i^\star} - \mathbf{S}_{ij})/\varepsilon}} && (8)\\
&\geq \frac{1}{1 + a_i e^{-\gamma_i/\varepsilon}}, && (9)
\end{aligned}$$

where the inequality in Eq. (9) follows from the fact that for any $j \neq j_i^\star$, we have $\mathbf{S}_{ij_i^\star} - \mathbf{S}_{ij} \geq \gamma_i$, and therefore $e^{-(\mathbf{S}_{ij_i^\star} - \mathbf{S}_{ij})/\varepsilon} \leq e^{-\gamma_i/\varepsilon}$. Here, $a_i = \sum_{j \neq j_i^\star} \frac{(\boldsymbol{\mu}_v)_j}{(\boldsymbol{\mu}_v)_{j_i^\star}}$.

This completes the proof.

B Compression Rate Computation

We define the compression rate of visual autoencoders in the same manner as in previous work [25, 47, 51]. The formal definition is given as follows:

Definition 1 (Compression Rate of VAEs). *Let $\mathbf{x} \in \mathbb{R}^{H \times W \times T \times C}$ denote the input video with spatial resolution (H, W), temporal length T, and channel dimension C. Let $\mathbf{z} \in \mathbb{R}^{H' \times W' \times T' \times C'}$ denote the corresponding latent representation produced by the VAE encoder. The* compression rate $\mathcal{R}$ *is defined as the ratio between the total number of latent variables and that of the input video, i.e.,*

$$r = \frac{H'W'T'C'}{HWTC}. \tag{10}$$

A smaller r indicates a higher compression level, implying that fewer latent tokens are used to represent the same visual information, thereby making the training of downstream video generative models more efficient. The compression rate of the proposed EmbodiedVAE and all baseline models is computed as follows:

- **Compression rate of EmbodiedVAE (ours)**: The proposed EmbodiedVAE employs a dual-encoder architecture that produces two complementary latent representations. Specifically, the *robotic-arm encoder* applies a temporal downsampling factor of 4 and a spatial downsampling factor of 16, while the *environment-context encoder* uses a temporal downsampling factor of 8 and a spatial downsampling factor of 8. Both latent spaces have a channel dimension of 4, leading to an overall compression rate of:

$$\begin{aligned} & r_{\text{EmbodiedVAE}} \\ =& \frac{4}{3 \times (4 \times 16 \times 16)} + \frac{4}{3 \times (8 \times 8 \times 8)} \\ \approx& 0.39\%. \end{aligned} \tag{11}$$

- **Compression rate of OpenSoraPlan-VAE** [24]: OpenSoraPlan-VAE applies a temporal downsampling factor of 4 and a spatial downsampling factor of 8, with the latent channel dimension of 8 resulting in a compression rate of:

$$r_{\text{OpenSoraPlan-VAE}} = \frac{8}{3 \times (4 \times 8 \times 8)} \approx 1.04\%. \tag{12}$$

- **Compression rate of Cosmos** [1]: We adopt the *Cosmos-Tokenizer*, specifically with "CV4x8x8", which applies a temporal downsampling factor of 4 and a spatial downsampling factor of 8, with a latent channel dimension of 16, resulting in a compression rate of:

$$r_{\text{Cosmos}} = \frac{16}{3 \times (4 \times 8 \times 8)} \approx 2.08\%. \tag{13}$$

- **Compression rate of iVideoGPT** [49]: iVideoGPT adopts a distinct compression strategy, where the first t_0 context frames are encoded using n_0 tokens, while the subsequent frames are represented with fewer tokens $n_1 < n_0$. Based on the configuration reported in the original paper, the overall compression rate is computed as:

$$\begin{aligned} r_{\text{iVideoGPT}} &= \frac{2 \times 16^2 \times 64 + 14 \times 4^2 \times 64}{3 \times 16 \times 256^2} \\ &\approx 1.50\%. \end{aligned} \tag{14}$$

- **Compression rate of MAGVIT-v2** [55]: MAGVIT-v2 applies a temporal downsampling factor of 4 and a spatial downsampling factor of 8. The latent channel dimension is set to 5, as reported in the original paper, resulting in a compression rate of:

$$r_{\text{MAGVIT-v2}} = \frac{5}{3 \times (4 \times 8 \times 8)} \approx 0.65\%. \tag{15}$$

– **Compression rate of VidTwin** [47]: Vidtwin decouples the input video into two latent representations of sizes $7 \times 7 \times 16 \times 4$ and $16 \times 14 \times 8$, for an input video clip of size $16 \times 3 \times 224 \times 224$. Accordingly, the overall compression rate is computed as:

$$\begin{aligned} r_{\text{VidTwin}} &= \frac{7 \times 7 \times 16 \times 4 + 16 \times 14 \times 8}{3 \times 16 \times 224 \times 224} \\ &\approx 0.20\%. \end{aligned} \tag{16}$$

– **Compression rate of EMU-3** [46]: Similar to MAGVIT-v2, EMU-3 achieves a $4\times$ temporal compression and an 8×8 spatial compression. With a latent channel dimension of 4, its compression rate is computed in the same manner as:

$$r_{\text{EMU-3}} = \frac{4}{3 \times (4 \times 8 \times 8)} \approx 0.53\%. \tag{17}$$

– **Compression rate of CV-VAE** [58]: In terms of compression rate, CV-VAE matches EMU-3, achieving a $4\times$ temporal compression and an 8×8 spatial compression. Accordingly, its compression rate is identical to EMU-3:

$$r_{\text{CV-VAE}} = \frac{4}{3 \times (4 \times 8 \times 8)} \approx 0.53\%. \tag{18}$$

– **Compression rate of CMD** [56]: CMD decouples video representations into *content frames* and *motion latents*. For a video of size (C, F, H, W), the content frame has dimensions (C, H, W), while the motion latent has dimensions $(D, H + W, F)$, where D denotes the dimension of the motion vector. Based on the configuration reported in the paper, the compression rate is computed as:

$$\begin{aligned} r_{\text{CMD}} &= \frac{1}{F} + \frac{D \times (H + W)}{C \times H \times W} \\ &= \frac{1}{16} + \frac{2 \times 224 \times 32}{3 \times 224 \times 224} \approx 6.85\%. \end{aligned} \tag{19}$$

– **Compression rate of SDXL** [31]: SDXL is an image-based VAE that encodes videos frame by frame without any temporal compression. Each frame is encoded with an 8×8 spatial downsampling factor and a latent channel dimension of 16. Thus, the overall compression rate is computed as:

$$r_{\text{SDXL}} = \frac{16}{3 \times 8 \times 8} \approx 8.32\%. \tag{20}$$

– **Compression rate of Cog-VAE** [54]: Cog-VAE applies a $4\times$ temporal downsampling and an 8×8 spatial downsampling, resulting in a latent representation with a channel dimension of 16. Accordingly, the overall compression rate is computed as:

$$r_{\text{Cog-VAE}} = \frac{16}{3 \times (4 \times 8 \times 8)} \approx 2.08\%. \tag{21}$$

– **Compression rate of Wan-VAE** [44]: Wan-VAE, similar to Cog-VAE, applies a $4\times$ temporal downsampling and an 8×8 spatial downsampling, resulting in a latent representation with a channel dimension of 16. However, it encodes and decodes the first frame independently. For a video of $4T + 1$ frames, the model produces $T + 1$ latent representations. Taking a $3 \times 17 \times 256 \times 256$ video clip as an example, the compression rate is computed as:

$$r_{\text{Wan-VAE}} = \frac{16 \times 5 \times 32 \times 32}{3 \times 17 \times 256 \times 256} \approx 2.45\%. \tag{22}$$

C Experiment Details

C.1 Details about the VAE Reconstruction

Implement Details We adopt a self-curated dataset containing approximately one million videos, each accompanied by pre-generated robotic arm masks obtained using SAM2 [33] and Grounding DINO [34] with the text prompt "Robotic Arm". Our model is trained on 3 fps, 16-frame, 256×256 video clips and evaluated on 16 fps, 16-frame, 256×256 video clips. Training is conducted on 16 NVIDIA A800 GPUs (40 GB). The detailed training configuration is provided in Table 1, where $\lambda_{\text{perception}}$ denotes the weight of the perception loss [38], and λ_{GAN} denotes the weight of the standard GAN loss, while β represents the weight of the KL divergence term and α corresponds to the weight of the OT-based consistency loss.

Baseline Details For the baselines including OpenSora [24], Cosmos [1], iVideoGPT [49], VidTwin [47], EMU-3 [46], CV-VAE [58], Cog-VAE [54], and Wan-VAE [44], we adopt the official checkpoints and implementations provided in their repositories. For MAGVIT-v2 [55] and CMD [56], since official code and pretrained checkpoints are not publicly available, we re-implemented these methods based on the descriptions in their respective papers.

C.2 Details about Action-Controlled Video Prediction for Robotic Manipulation

Task Description Video generative models have recently demonstrated strong potential as world models. When training such models for robotic manipulation, it is crucial to accurately simulate the complex interactions among the robot, manipulated objects, and the surrounding environment. In this task, we evaluate whether the models can generate realistic videos that depict fine-grained robot–object interaction dynamics, given a sequence of historical observations and a corresponding action trajectory.

Table 1: Training Configuration.

Parameter	Value
Input Video Resolution	256×256
Input Video Frames	16
Input Video FPS	3
Adam Optimizer	$\beta_1 = 0.9$, $\beta_2 = 0.99$
First Training Stage	
Learning Rate	1.0×10^{-5}
wegiht decay	1.0×10^{-4}
$\lambda_{\text{perception}}$	1.0×10^{0}
λ_{GAN}	5.0×10^{-3}
β	1.0×10^{-6}
α	1.0×10^{4}
Second Training Stage	
Learning Rate (Decoder)	1.0×10^{-5}
Learning Rate (Encoder)	1.0×10^{-8}
$\lambda_{\text{perception}}$	1.0×10^{0}
λ_{GAN}	5.0×10^{-3}
β	1.0×10^{-6}

LDM Backbone To ensure fairness, we adopt IRASim-L (461M) [59] as the downstream LDM backbone for all robotic manipulation video prediction experiments, while the only difference lies in the choice of VAEs (including our proposed EmbodiedVAE and the baseline models) used during training and inference. IRASim is a DiT-style model that incorporates a novel frame-level action-conditioning module within each transformer block, explicitly modeling and strengthening the alignment between every action and its corresponding video frame. IRASim provides four model scales: (1) *IRASim-S* with 12 layers and a hidden size of 768, (2) *IRASim-B* with 12 layers and a hidden size of 768, (3) *IRASim-L* with 24 layers and a hidden size of 1024, and (4) *IRASim-XL* with 28 layers and a hidden size of 1152. To balance performance and computational resource consumption, we adopt IRASim-L, which contains 461M parameters, for all robotic manipulation experiments.

Implement Details Action Condition Injection As different VAEs exhibit varying temporal compression rates, we detail the action-injection strategies applied to each VAE's latent representations for IRASim as follows:

– For our proposed *EmbodiedVAE*, given a sequence of $T-1$ action vectors $\mathbf{A} \in \mathbb{R}^{(T-1)\times d}$, where d denotes the action dimension, each action represents the relative motion with respect to the first frame. We first pad a zero vector to align the sequence with the T video frames, obtaining $\hat{\mathbf{A}} \in \mathbb{R}^{T\times d}$. A learnable 1D convolution is then applied to temporally downsample $\hat{\mathbf{A}}$ into $\hat{\mathbf{A}}_{\text{arm}} \in \mathbb{R}^{\frac{T}{4}\times d}$, which is injected into the latent representation of the robotic

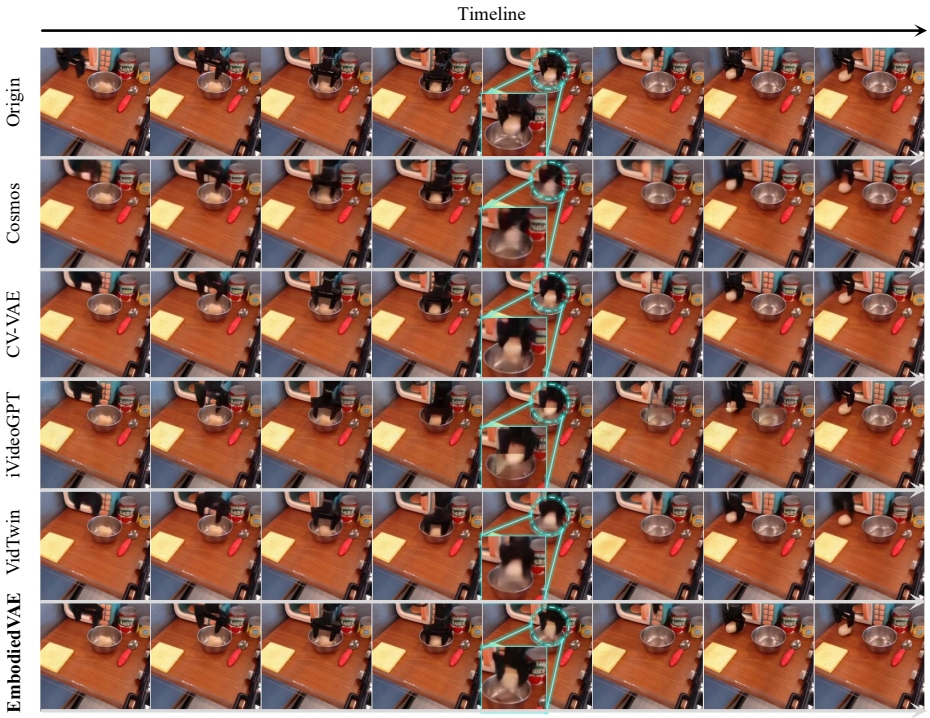

Fig. 1: Additional examples of VAE reconstruction tasks on the Bridge [43] dataset.

arm, $\mathbf{Z}_{\text{arm}}$. Subsequently, another 1D convolution is used to further downsample $\hat{\mathbf{A}}_{\text{arm}}$ into $\hat{\mathbf{A}}_{\text{env}} \in \mathbb{R}^{\frac{T}{8} \times d}$, representing the environmental actions that are injected into the latent representation $\mathbf{Z}_{\text{env}}$. Both latent representations share the same diffusion process and LDM parameters.

- For the *OpenSoraPlan-VAE* [24], *Emu3* [46], *CV-VAE* [58], and *Cog-VAE* [54] which resulting the latent with 4 temporal downsamlping, the same as our EmbodiedVAE we adopting the 1D convolution to downsampleing the $\hat{\mathbf{A}}$ into $\hat{\mathbf{A}}_{\text{align}} \in \mathbb{R}^{\frac{T}{4} \times d}$ to provide aligned actions.
- As for *Wan-VAE* [44], since it encodes the first frame independently, we follow its encoding strategy when injecting action information. Given $T+1$ frames, we apply a 1D convolution to the actions of the last T frames, downsampling them into $\frac{T}{4}$ action conditions. These are then concatenated with the first-frame action, yielding the aligned action sequence $\hat{\mathbf{A}}_{\text{align}} \in \mathbb{R}^{(1+\frac{T}{4}) \times d}$.
- For *CMD* [56], since it decouples the input into a content latent without a temporal dimension and a motion latent without temporal compression, we follow the original paper and employ two independent diffusion processes to denoise the motion and content latents, each modeled by an IRASim DiT architecture of the same design. We apply the complete action sequence

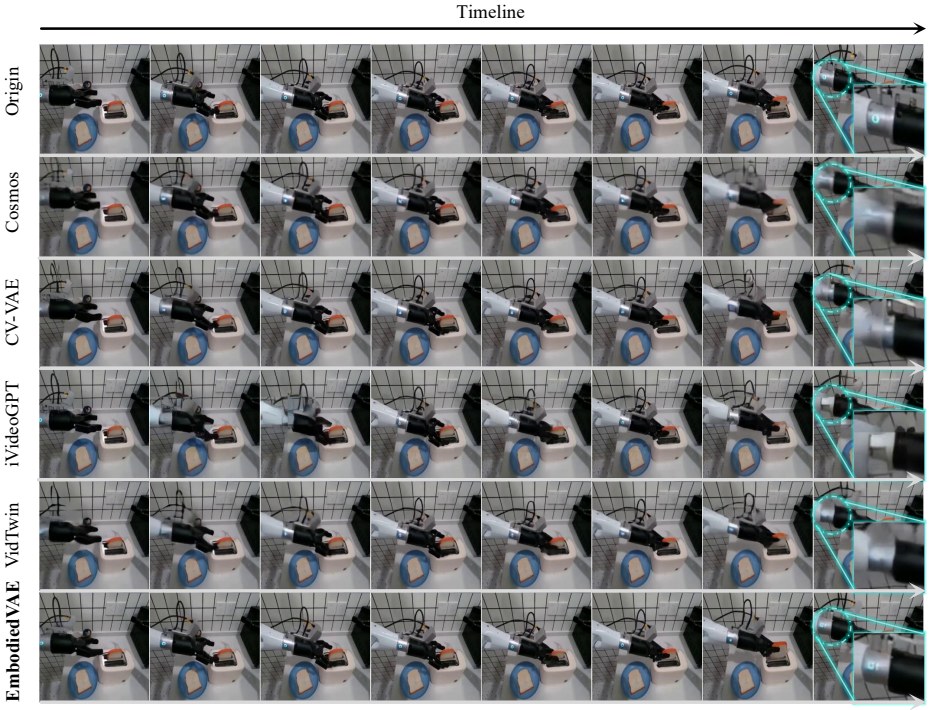

Fig. 2: Additional examples of VAE reconstruction tasks on the Agibot-2025 [6] dataset.

$\hat{\mathbf{A}} \in \mathbb{R}^{T \times d}$ to the motion diffusion branch, as the motion latent preserves the full temporal resolution. For the content diffusion branch, we use the content latent representations of the historical frames as conditional inputs to guide the generation.

- For *VidTwin* [47], it produces three latent representations: a structure latent $\mathbf{Z}_S \in \mathbb{R}^{d_S \times T \times h_S \times w_S}$ and two dynamics latents, $\mathbf{Z}_D^h \in \mathbb{R}^{d_D \times T \times h_D}$ and $\mathbf{Z}_D^w \in \mathbb{R}^{d_D \times T \times w_D}$. Following the procedure described in the original paper, we apply two independent patchification modules to convert both dynamics latents into sequences of tokens. These token sequences are normalized and aligned to a comparable scale, then concatenated along the temporal dimension. Finally, the complete action sequence $\hat{\mathbf{A}}$ is injected into the latent representations.
- For *SDXL* [31], the official VAE used in IRASim [59], we follow its default configuration and represent actions as relative motion with respect to the first frame.

In this way, we ensure fairness by injecting the action conditions consistently across all models. For CMD, we adopt two independent diffusion processes to denoise its motion and content latents separately, following the official implementation [56]. In contrast, a single diffusion process is employed in all other

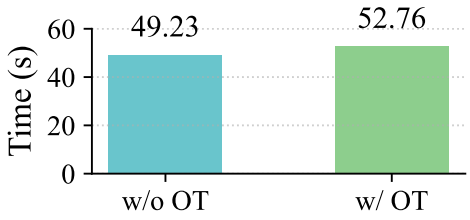

Fig. 3: Computational Time Analysis of the OT-based Module.

models Each diffusion process consists of 1000 steps, with DDIM [37] used as the sampling strategy and 50 steps applied during inference.

D Computational Time Analysis of the OT-based Module

Although the complexity of this module is $\mathcal{O}(\Delta T N^2 d)$, in practice, the OT-based loss is applied to latents with the input resolution 256×256, and our robotic arm encoder applies strong spatial compression with 16×16, so the actual computational overhead is modest. We report the training time per 100 samples with and without motion loss on A100 GPUs in Figure 3, showing that *the cost introduced by the OT module is acceptable in practice.*

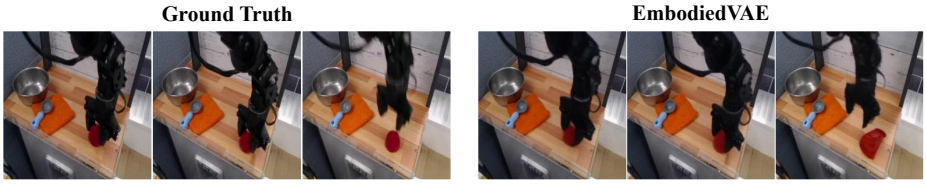

Fig. 4: Scenarios Where Robotic Arms Occupy Large Spatial Regions.

E Scenarios Where Robotic Arms Occupy Large Spatial Regions

We provide examples in Figure 4 where the robotic arms occupy relative large portion of the frame. As shown, *even when the robotic arms violate the assumption* of occupying only a small spatial region, our proposed model still achieves high-quality reconstruction and manipulation performance.

F Additional Results

Here, we present additional experimental results for both tasks. The additional experimental results for the video VAE reconstruction tasks are shown in Fig-

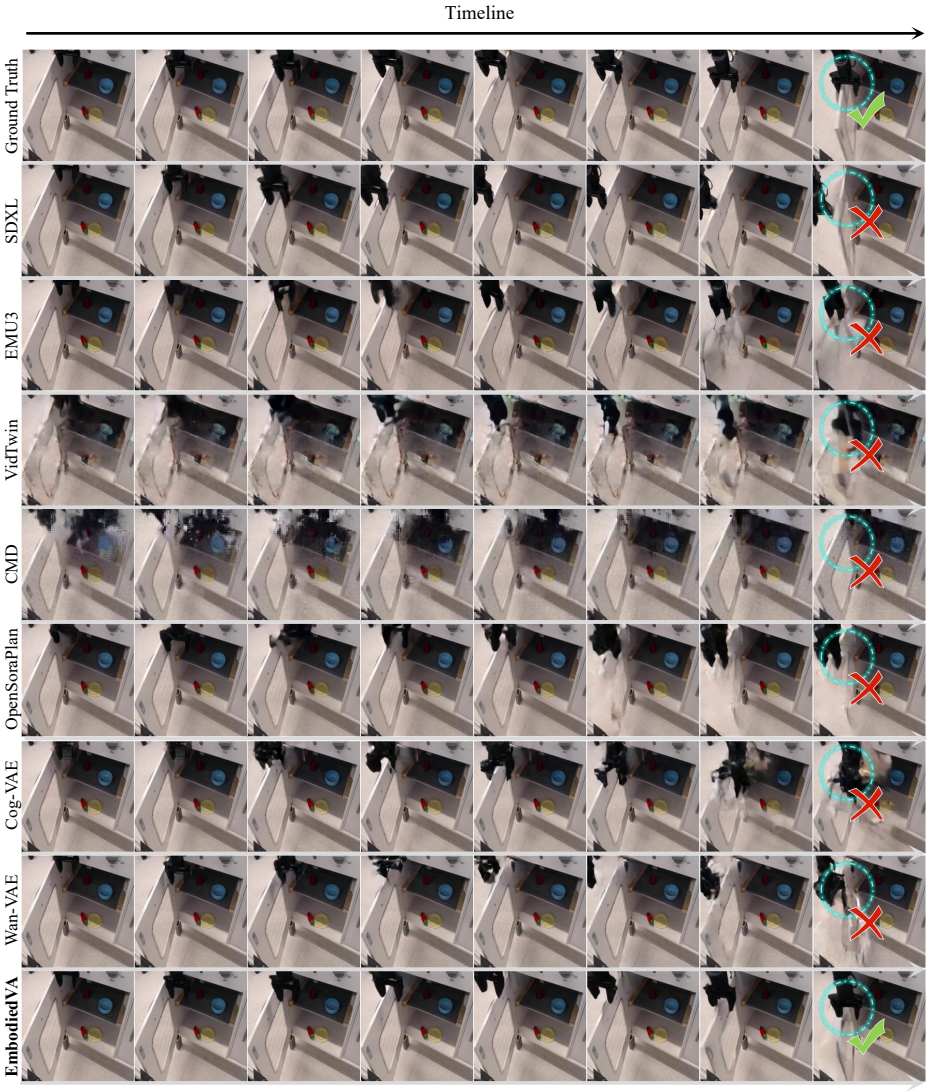

Fig. 5: Additional examples of video prediction tasks in robotic manipulation scenarios on the Bridge [43] dataset.

ure 1 and Figure 2. As can be observed, our proposed EmbodiedVAE demonstrates superior reconstruction capability, particularly in accurately capturing the robotic arms and electronic components during manipulation. Furthermore, the additional results for downstream video prediction in robotic manipulation scenarios are presented in Figure 5 and Figure 6. As shown, EmbodiedVAE provides a more controllable and compact latent representation that enables more accurate action control. This improvement can be attributed to our disentan-

Fig. 6: Additional examples of video prediction tasks in robotic manipulation scenarios on the RT-1 [5] dataset.

gling modules coupled with the OT-based motion consistency module, which together produce explicit, temporally consistent motion latent representations that are free from the irrelevant background information.