V-RAE: RETHINKING VIDEO LATENT SPACES FOR GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent video generation relies on autoencoders to define a compact space in which generative models operate. Although video autoencoders have evolved substantially, their latent spaces are still optimized primarily for pixel-level reconstruction and capture limited high-level semantics. A reconstruction-optimal latent space, however, need not be well suited to generative modeling. We propose **V-RAE**, a video representation autoencoder that builds temporally compact generative latents on top of frozen vision foundation models (VFMs). A lightweight temporal pooling module reduces temporal redundancy while preserving semantic structure, and a video decoder reconstructs videos from the compressed features. We evaluate V-RAE with four representative VFMs on video reconstruction, semantic probing, and class-conditional generation. V-RAE achieves **2.13 rFVD** on K600, outperforming all evaluated large-scale pretrained video VAEs. Its latents retain substantially more semantic information than conventional video tokenizer latents. V-RAE achieves gFVD scores of **117.86** on UCF101 and **19.16** on K600, and matches Wan2.2 VAE's converged gFVD with **up to** $6\times$ **faster convergence**. We further show that reconstruction quality alone is insufficient to characterize generative utility and introduce **tFVD**, a temporal-coherence diagnostic that correlates more strongly with downstream generation quality than rFVD. Beyond video generation, V-RAE also improves future video prediction on Cityscapes over the Wan 2.2 VAE latent space under matched prediction settings. Taken together, these results show that frozen semantic representations provide an effective basis for video reconstruction, generation, and predictive modeling. Our code is anonymously available at `https://anonymous.4open.science/r/v-rae`.

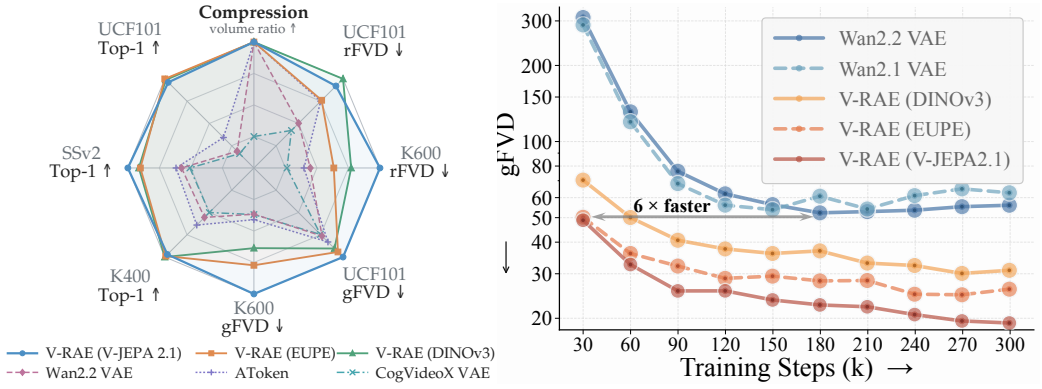


Figure 1: **V-RAE overview. Left:** Normalized reconstruction, generation, compression, and semantic performance across video tokenizers; FVD axes (↓) are reversed so outward is better. **Right:** K600 gFVD convergence; V-RAE converges up to $6\times$ faster than VAE-based latent spaces.

1 INTRODUCTION

Modern video generation (Blattmann et al., 2023) typically operates in compressed video-autoencoder latents to reduce the cost of diffusion or autoregressive modeling (Ma et al., 2025b; Kondratyuk et al., 2023). The autoencoder imposes an inductive bias: its latent structure shapes

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.

modeling difficulty and bounds attainable generation quality. Continuous VAEs (Wan et al., 2025; Kong et al., 2024; Yang et al., 2025), discrete tokenizers (Yu et al., 2023; Wang et al., 2024a), and query-based designs (Wang et al., 2025; Xiong et al., 2026) largely prioritize reconstructing textures, colors, and fine-grained appearance. However, optimizing reconstruction fidelity alone does not necessarily yield a latent space that is favorable for generative modeling (Yao et al., 2025; Skorokhodov et al., 2025; Xu et al., 2026), especially for video, which requires jointly modeling appearance, motion, and long-range spatial-temporal dependencies over latent sequences.

Recent studies have therefore begun incorporating semantics into latent generative modeling. In image generation, Ma et al. (2025a) and Wu et al. (2025b) introduce semantic alignment during autoencoder training. For video, VideoREPA (Zhang et al., 2025) extends representation alignment (Yu et al., 2024; Leng et al., 2025) to video by aligning intermediate diffusion features with pretrained representations. These methods primarily use semantics for feature alignment rather than directly defining the generative latent space. RAE (Zheng et al., 2025b), Scale-RAE (Tong et al., 2026) and RAEv2 (Singh et al., 2026) instead generate directly in a frozen image encoder's feature space. Their semantically rich, coherent latents support diffusion modeling and share representations across understanding and generation. Whether a frozen representation encoder can similarly support high-fidelity video reconstruction and effective generation remains largely unexplored.

This extension is non-trivial because video introduces three additional challenges. **First**, representation encoders produce temporally dense features. Image encoders provide no temporal compression, while the native video encoders considered here provide only $2\times$ compression, far below the rate required to keep DiT training and inference affordable. **Second**, temporal coherence must be preserved in both compressed latents and decoded motion; otherwise, videos exhibit flicker and jitter, and the latent distribution becomes difficult for diffusion models to learn. **Third**, temporal compression must preserve the semantic structure inherited from the encoder, which is the central advantage of using representation features as generative latents.

To address these challenges, we propose **V-RAE**, a video representation autoencoder that uses frozen visual features as its latent space and adds learnable temporal compression and decoding for video generation. *On the encoder side*, a frozen vision foundation model (VFM) integrates with a learnable lightweight temporal pooling module that reduces temporal redundancy while preserving semantic structure. *On the decoder side*, we design a spatiotemporal Transformer decoder equipped with 3D RoPE (Su et al., 2021) to reconstruct temporally coherent videos from the compressed latents. A multi-frame unpatchify layer maps each temporal latent step to multiple consecutive frames, allowing the decoder to accommodate the effective temporal compression ratio of each representation encoder. Since the representation encoder remains frozen, the resulting latent space stays grounded in its pretrained semantic organization.

We evaluate V-RAE instantiated with four representative frozen encoders, i.e., DINOv3 (Siméoni et al., 2025), SigLIP2 (Tschannen et al., 2025), EUPE (Zhu et al., 2026), and V-JEPA 2.1 (Mur-Labadia et al., 2026), across reconstruction fidelity, semantic preservation, and generative quality. Figure 1 provides a holistic comparison between V-RAE and representative video tokenizers. For reconstruction, V-RAE with V-JEPA 2.1 attains **2.13 rFVD on K600**, surpassing all evaluated large-scale pretrained video VAEs. On UCF101, its DINOv3-L variant reaches 6.12 rFVD, ranking second only to Wan2.1 VAE at 6.05. V-RAE latents also preserve substantially richer semantic information than conventional VAE latents across three probing benchmarks. In particular, V-RAE with SigLIP2-L reaches **90.92% top-1 accuracy on UCF101**, compared with only 30.83% for the strongest VAE baseline. The semantic richness of the learned latent space further translates into improved generation. Our best V-RAE variants achieve gFVD scores of **117.86** on UCF101 and **19.16** on K600. On K600, V-RAE matches Wan2.2 VAE's converged gFVD using only **30K** versus **180K** training updates ($6\times$ faster convergence). Together, these results show that V-RAE can preserve pretrained semantics under temporal compression while supporting effective video generation. Our experiments show that rFVD has Pearson correlations with gFVD of 0.200 on UCF101 and 0.473 on K600, suggesting that reconstruction quality alone is insufficient to characterize generative suitability. To complement this assessment, we examine temporal interpolation quality in video latent spaces. We introduce **tFVD** to assess this property by evaluating videos decoded from temporally interpolated latents. V-RAE achieves the **lowest tFVD** among the evaluated tokenizers, suggesting that its latent space supports smoother and more coherent temporal interpolation. Furthermore, tFVD correlates more strongly with gFVD (**0.621 on UCF101 and 0.919 on K600**), supporting its use as a more informative diagnostic of generative suitability. Finally, we demonstrate that V-RAE

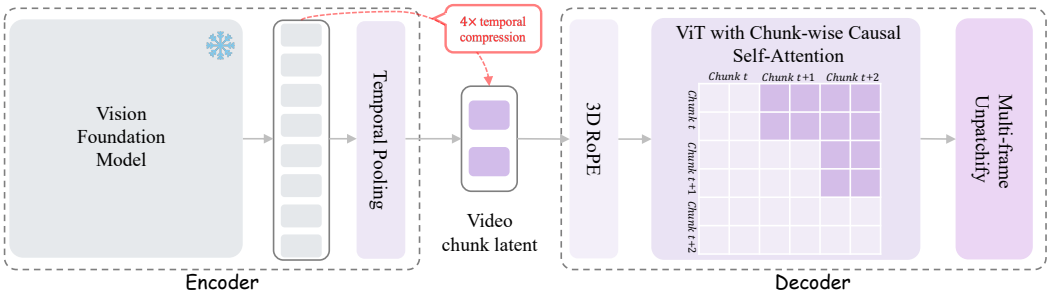


Figure 2: V-RAE architecture. A frozen visual representation encoder produces temporally dense features, which are compressed by a lightweight temporal pooling module and reconstructed by a Transformer decoder. The figure depicts the chunk-wise causal decoder used with image encoders.

provides an effective latent representation beyond video generation: on future video prediction with Cityscapes, it substantially outperforms the Wan2.2 VAE latent space under the same prediction architecture and training budget. Our contributions can be summarized as follows:

- We introduce V-RAE, combining a frozen VFM, a lightweight temporal attention pooler, and a spatiotemporal decoder to produce temporally compressed, reconstructable, and semantically rich video latents. We validate this design across four VFM backbones.
- We introduce tFVD to probe temporal interpolation decodability and show that it correlates more strongly with generation quality than reconstruction fidelity across evaluated tokenizers.
- V-RAE achieves competitive reconstruction and stronger generation performance with up to $6\times$ faster training convergence, while better future prediction supports the predictability of its semantic latents.

2 Method

2.1 V-RAE Architecture

Figure 2 outlines V-RAE, which combines a frozen representation encoder, a learnable temporal pooler, and a Transformer decoder. The same framework supports frame-wise DINOv3 (Siméoni et al., 2025), SigLIP2 (Tschannen et al., 2025), and EUPE (Zhu et al., 2026), and spatiotemporal V-JEPA 2.1 (Mur-Labadia et al., 2026). Following RAEv2 (Singh et al., 2026), we fuse features from multiple encoder blocks before temporal pooling; only the pooler and decoder are trained.

Temporal pooling. For an input video $\boldsymbol{X} \in \mathbb{R}^{T\times 3\times H\times W}$, the frozen VFM produces $\boldsymbol{F} = \mathcal{E}(\boldsymbol{X}) \in \mathbb{R}^{T_E\times C\times h\times w}$ with native temporal compression ratio $r_E = T/T_E$. The pooler $\mathcal{P}$ maps each non-overlapping window of r_{pool} features to one latent feature map:

$$\boldsymbol{Z} = \mathcal{P}(\boldsymbol{F}) \in \mathbb{R}^{T_Z\times C\times h\times w}, \qquad T_Z = T_E \,/\, r_{\text{pool}}, \qquad r_{\text{all}} = T \,/\, T_Z = r_E \cdot r_{\text{pool}}. \tag{1}$$

To balance reconstruction quality, semantic preservation, and parameter efficiency, we adopt temporal attention pooler as our default design, with detailed comparisons to alternative pooling strategies provided in Appendix B.1. Specifically, the pooler aggregates patch features across time independently at each spatial location, where t denotes the pooled time step and $p \in \{1,\dots,hw\}$ indexes a position on the flattened $h \times w$ feature grid. Let $f_{\tau,p} \in \mathbb{R}^C$ denote the patch feature in F at time τ and position p. The pooled token is

$$\boldsymbol{z}_{t,p} = \text{Norm}\Big(\text{MHA}\Big(\boldsymbol{q}, \{\boldsymbol{f}_{r_{\text{pool}}t+i,p}\}_{i=0}^{r_{\text{pool}}-1}\Big)\Big). \tag{2}$$

The 1D query $\boldsymbol{q}$ is shared across temporal windows and spatial positions. To initialize the module close to temporal mean pooling, we use identity-initialized projections and zero-initialized queries and temporal biases, allowing it to subsequently learn content-adaptive weights.

Video decoder. The decoder maps latent $\boldsymbol{Z}$ back to reconstructed video $\widehat{\boldsymbol{X}}$ using an MAE-style ViT with 3D RoPE (He et al., 2022; Dosovitskiy et al., 2021; Su et al., 2021). Its attention pattern follows the temporal context of the frozen VFM: frame-wise encoders use chunk-wise causal attention, whereas V-JEPA 2.1 uses full spatiotemporal attention. A multi-frame unpatchify head maps each latent step to r_{all} consecutive frames, matching the combined compression of the encoder and pooler. We jointly optimize the temporal pooler and video decoder using the reconstruction objective in Eq. 9. Detailed architecture and training settings are provided in Appendices B.1 and B.2.

Table 1: Reconstruction performance of video tokenizers on UCF101 and K600. **Bold** and underlined values indicate the best and second-best results across all methods, respectively.

Tokenizer Configuration				UCF101 test				K600 val			
Method	**Causal**	**Comp.**	**Dim.**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**rFVD ↓**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**	**rFVD ↓**
Large-Scale Pretrained Video Tokenizers											
Wan2.1 VAE	✓	$4 \times 8 \times 8$	16	0.068	34.09	0.937	**6.05**	0.052	35.23	0.946	3.58
Wan2.2 VAE	✓	$4 \times 16 \times 16$	48	**0.067**	34.16	0.942	12.20	**0.049**	35.73	0.951	4.76
HunyuanVideo VAE	✓	$4 \times 8 \times 8$	16	0.069	**35.14**	**0.950**	7.73	0.056	**36.31**	**0.954**	4.38
CogVideoX VAE	✓	$4 \times 8 \times 8$	16	0.081	34.22	0.943	14.53	0.069	34.82	0.946	8.11
Cosmos-0.1 (CV4x8x8)	✓	$4 \times 8 \times 8$	16	0.109	32.57	0.925	17.01	0.100	33.55	0.928	9.88
AToken	✗	$4 \times 16 \times 16$	48	0.089	33.11	0.934	8.17	0.072	34.05	0.937	5.36
Benchmark-Scale Video Tokenizers											
Open-MAGVIT2	✓	$4 \times 8 \times 8$	18	0.140	27.74	0.867	36.21	0.147	27.74	0.859	17.16
OmniTokenizer VAE	✓	$4 \times 8 \times 8$	8	0.132	29.59	0.908	28.86	0.139	27.75	0.899	15.35
LARP-L-long	✗	1024 tokens	16	0.233	25.32	0.799	142.29	0.236	26.00	0.801	125.00
V-RAE Series (Ours)											
V-RAE (DINOv3-L)	✓	$4 \times 16 \times 16$	1024	0.111	27.76	0.858	6.12	0.104	27.96	0.850	2.76
V-RAE (EUPE-B)	✓	$4 \times 16 \times 16$	768	0.122	26.76	0.829	8.05	0.114	27.30	0.829	3.36
V-RAE (SigLIP2-L)	✓	$4 \times 16 \times 16$	1024	0.142	26.41	0.815	9.83	0.134	27.05	0.816	3.38
V-RAE (V-JEPA2.1-L)	✗	$4 \times 16 \times 16$	1024	0.121	27.82	0.856	6.65	0.110	28.48	0.858	**2.13**

2.2 Generative Modeling in V-RAE Latents

With V-RAE frozen, we train a DiT on its normalized latents using rectified flow (Peebles & Xie, 2023; Liu et al., 2023), with dimension-dependent time shifting and an auxiliary prediction head for intermediate supervision, as detailed in Appendix B.3.

3 Experimental Settings

Datasets. We evaluate reconstruction and class-conditional generation on UCF101 (Soomro et al., 2012) and Kinetics-600 (K600) (Carreira et al., 2018). For reconstruction, all V-RAE variants are trained on the union of the two training sets and evaluated on the UCF101 test set and K600 validation set. For generation, we train a separate class-conditional DiT on each dataset using 20-frame clips sampled at an interval of 3; UCF101 and K600 gFVD use 2,048 and 50,000 generated clips, respectively. We probe semantic preservation on UCF101, Something-Something V2 (SSv2) (Goyal et al., 2017), and Kinetics-400 (K400) (Kay et al., 2017).

Implementation. Reconstruction metrics are LPIPS (Zhang et al., 2018), PSNR, SSIM (Wang et al., 2004), and rFVD at 256×256. V-RAE, AToken (Lu et al., 2025), and LARP-L-long (Wang et al., 2025) use 16 frames; causal VAEs with a separate first frame use 17. Unlike LARP, which evaluates on the training split, we evaluate the initial clip of each UCF101 test and K600 validation video. Sampling every third frame exposes larger motion and follows Latte (Ma et al., 2025b). For class-conditional generation, matched settings refer to a shared DiT backbone, $1{,}280$-token budget with tokenizer-specific patchification, training budget, and evaluation protocol, not identical output heads. VAE baselines use standard DiT; V-RAE uses DiT^{DH} with RAE's wide DDT head (Zheng et al., 2025b). All gFVD evaluations use 17 frames at 256×256: causal VAEs generate 17 directly; V-RAE and AToken generate 20 and retain the first 17. See Appendices B and C for details.

Baselines. Baselines include Wan2.1 (Wan et al., 2025) and Wan2.2 VAEs (Team Wan, 2025), HunyuanVideo VAE (Kong et al., 2024), CogVideoX VAE (Yang et al., 2025), Cosmos tokenizer (Agarwal et al., 2025), AToken (Lu et al., 2025), MAGVIT-v2 (Yu et al., 2023; Luo et al., 2024), OmniTokenizer (Wang et al., 2024a), and LARP (Wang et al., 2025). Generation comparisons evaluate quality and convergence, while frame-wise RAEv2 (Singh et al., 2026) and controlled pooling variants isolate temporal modeling and the reconstruction–semantics trade-off.

4 Results and Discussion

4.1 Reconstructing Videos from Semantic Latents

Overall Reconstruction Performance. Table 1 shows that V-RAE remains competitive with large-scale reconstruction-oriented video tokenizers despite keeping its semantic encoder frozen. On UCF101, the DINOv3-L variant achieves an rFVD of 6.12, close to Wan2.1 VAE at 6.05. On K600, the V-JEPA 2.1-L variant achieves an rFVD of 2.13, improving over the strongest video VAE

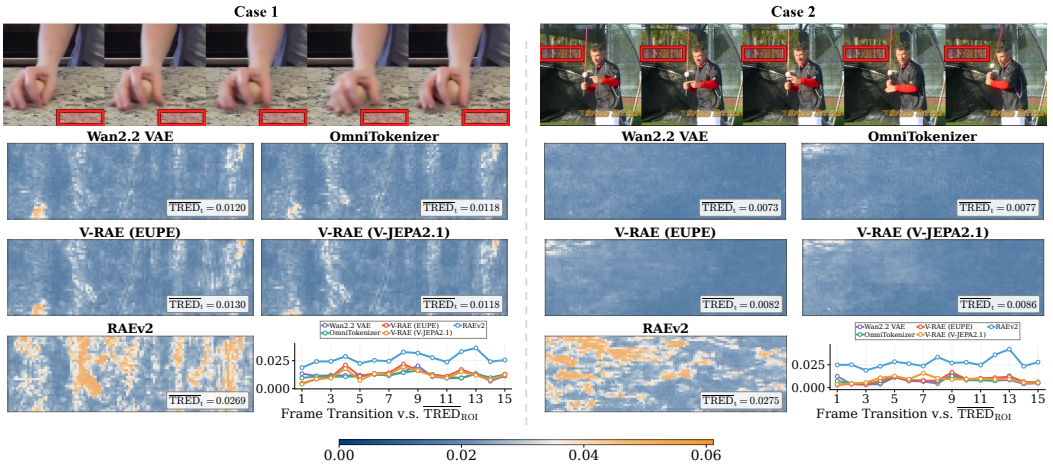


Figure 3: TRED on K600 validation examples. Heatmaps show temporal averages within high-frequency regions of interest; curves show spatial averages over consecutive frame transitions.

baseline at 3.58 by 40.5%. All four V-RAE variants also outperform the evaluated large-scale video VAE baselines on K600. Rankings vary by dataset: DINOv3-L leads among V-RAE variants on UCF101, whereas V-JEPA 2.1-L leads on K600. These results suggest that reconstruction capability extends across encoders and pretraining paradigms, while each encoder's spatiotemporal inductive bias affects its compatibility with the target distribution.

Despite these strong rFVD results, V-RAE trails some reconstruction-oriented video VAEs on LPIPS, PSNR, and SSIM. This divergence is informative. Frozen semantic features may discard low-level appearance details emphasized by frame-wise distortion metrics while preserving the spatiotemporal structure captured by rFVD. These results indicate that optimizing the encoder for pixel reconstruction is not necessary for constructing a coherent video latent space, motivating our later investigation of its generative properties. Figure 3 shows that V-RAE approaches the two VAEs in temporal stability and outperforms frame-wise RAEv2 on the presented examples, measured by Temporal Reconstruction Error Difference (TRED), defined in Appendix E.

The Impact of Temporal Modeling. To further assess the contribution of temporal modeling to reconstruction quality, we compare V-RAE with frame-wise RAEv2. Each pair in Figure 4 uses the same frozen image VFM, controlling for the underlying semantic representation. Across DINOv3-L, SigLIP2-L, and EUPE-B, V-RAE lowers rFVD by factors between 2.7 and 7.6. This controlled comparison shows that temporal pooling and cross-frame decoding provide gains beyond those inherited from the pretrained encoder.

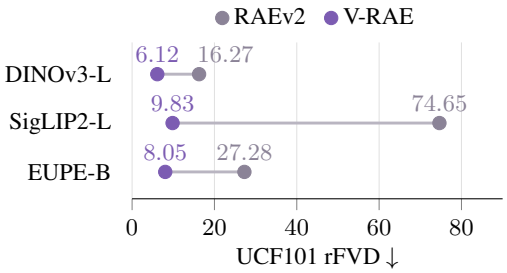


Figure 4: Effect of temporal modeling on UCF101 reconstruction; lower rFVD is better.

Semantic Preservation. Beyond reconstruction quality, we examine whether V-RAE preserves pretrained semantics after multi-layer fusion and temporal compression. Adapting the DINOv3 protocol (Siméoni et al., 2025), we conduct probing evaluations on UCF101, SSv2, and K400. Table 2 shows that the best V-RAE variants achieve 90.92%, 72.91%, and 83.12% accuracy, respectively, only 3.13, 3.67, and 3.20 percentage points below their corresponding frozen encoders. These scores substantially exceed the strongest evaluated VAE results (30.83%, 45.05%, and 53.27%), demonstrating that V-RAE preserves appearance and motion semantics.

FINDING 1 **Temporal compression need not erase semantics.**
Adaptive temporal pooling reduces redundancy while retaining pretrained semantics.

4.2 Generation from Semantic Latents

Overall Generation Performance. We train separate class-conditional DiTs from scratch on UCF101 and K600 in each tokenizer's latent space. As shown in Table 3, all four V-RAE vari-

Table 2: Top-1 probing accuracy; parentheses show changes from the corresponding frozen encoder.

Model	Temporal Compression	UCF101 ↑		SSv2 ↑		K400 ↑	
Probe Type		*Linear Probe*		*Attentive Probe*		*Attentive Probe*	
		Single	TTA	Single	TTA	Single	TTA
RAE-based Video Autoencoders							
DINOv3-L	1×	91.84	91.84	69.92	70.21	86.32	87.15
V-RAE (DINOv3-L)	4×	89.13 (-2.71)	88.94 (-2.90)	66.55 (-3.37)	67.19 (-3.02)	83.12 (-3.20)	84.36 (-2.79)
EUPE-B	1×	93.86	93.78	67.17	67.64	83.53	84.21
V-RAE (EUPE-B)	4×	90.16 (-3.70)	89.93 (-3.85)	65.67 (-1.50)	66.21 (-1.43)	82.21 (-1.32)	83.31 (-0.90)
SigLIP2-L	1×	94.05	93.55	68.54	68.80	86.00	86.93
V-RAE (SigLIP2-L)	4×	90.92 (-3.13)	90.88 (-2.67)	65.39 (-3.15)	66.08 (-2.72)	82.56 (-3.44)	83.91 (-3.02)
V-JEPA 2.1-L	2×	93.02	92.83	76.58	77.11	84.29	85.20
V-RAE (V-JEPA 2.1-L)	4×	86.65 (-6.37)	86.65 (-6.18)	72.91 (-3.67)	73.50 (-3.61)	80.70 (-3.60)	81.87 (-3.34)
VAE-based Video Autoencoders							
Wan2.2 VAE	4×	16.94	16.29	41.86	42.62	46.13	48.11
AToken	4×	30.83	30.29	45.05	45.95	53.27	55.32
CogVideoX VAE	4×	14.51	14.14	37.03	38.34	41.59	43.53

ants achieve lower gFVD than the evaluated baselines on both datasets. V-RAE with V-JEPA 2.1 performs best, achieving 117.86 gFVD on UCF101 and 19.16 on K600, improvements of 25.14 and 22.50 points over the strongest VAE baselines.

Optimization Efficiency. Figure 5 shows that the V-JEPA 2.1 and EUPE-B variants maintain lower gFVD than Wan2.2 VAE at every evaluated UCF101 checkpoint. V-JEPA 2.1 matches Wan2.2 VAE's 150K-update performance in roughly 30K updates, converging approximately 5× faster.
On K600, Figure 1 (right) shows that EUPE-B matches Wan2.2 VAE's 180K-update performance at 30K updates (6× faster), while improving gFVD by 24.1 at the same 180K-update budget. Figure 6 shows improved visual integrity and cross-frame consistency on UCF101; additional results for K600, DINOv3-L, and SigLIP2-L appear in Appendix F.

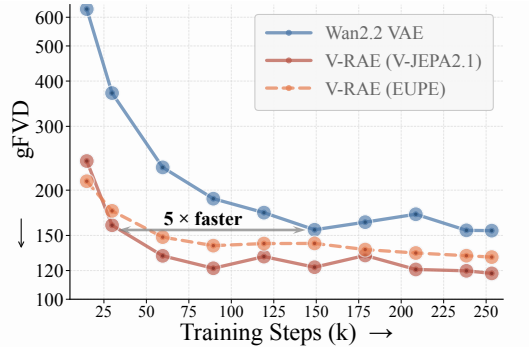


Figure 5: gFVD convergence on UCF101. V-RAE is up to 5× faster than Wan2.2 VAE.

FINDING 2 **Semantic latents make video generation easier to learn.**
Semantic latents expose objects, actions, and scene structure, letting the generator focus on visual dynamics rather than recover semantics from reconstruction-oriented latents.

4.3 Good Reconstruction Does Not Guarantee Good Generation

Reconstruction fidelity alone does not determine whether a latent space is suitable for generative modeling. Reconstruction follows the deterministic path $\boldsymbol{X} \xrightarrow{\mathcal{E}} \boldsymbol{F} \xrightarrow{\mathcal{P}} \boldsymbol{Z} \xrightarrow{\mathcal{D}} \widehat{\boldsymbol{X}}$, where the decoder receives latents encoded from real videos. A generator must instead learn the latent distribution and produce new latent samples that remain reliably decodable. This distinction is evident in the results in Table 3, HunyuanVideo VAE achieves a lower rFVD than V-RAE with EUPE-B (7.73 vs. 8.05), but a substantially worse gFVD (211.53 vs. 125.98). Similarly, on K600, Wan2.1 VAE and V-RAE with SigLIP2-L achieve comparable rFVD scores (3.58 vs. 3.38), while their gFVD scores differ markedly (53.75 vs. 34.48). These reversals show that fidelity at encoded samples does not reveal how easily the surrounding latent space can be learned and decoded.

Temporal Fréchet Video Distance (tFVD). To probe decodability between encoded temporal states without training a generator, we introduce **tFVD**. As shown in Figure 7(a), given $\boldsymbol{Z} =$

Table 3: Class-conditional video generation on UCF101 and K600 with different video latent spaces.

Method	Resolution, Frames	Generator	#Tokens	UCF101 gFVD ↓	K600 gFVD ↓
VAE-Based Video Autoencoders					
Wan2.1 VAE	256 × 256, 17	DiT XL/2	1280	148.20	53.75
Wan2.2 VAE	256 × 256, 17	DiT XL/1	1280	154.64	52.25
HunyuanVideo VAE	256 × 256, 17	DiT XL/2	1280	211.53	53.28
CogVideoX VAE	256 × 256, 17	DiT XL/2	1280	159.20	51.83
Cosmos-0.1 (CV4x8x8)	256 × 256, 17	DiT XL/2	1280	152.70	41.66
AToken	256 × 256, 20 → 17	DiT XL/1	1280	143.00	46.74
V-RAE Series (Ours)					
V-RAE (DINOv3-L)	256 × 256, 20 → 17	DiT^{DH} XL/1	1280	131.40	30.09
V-RAE (SigLIP2-L)	256 × 256, 20 → 17	DiT^{DH} XL/1	1280	142.60	34.48
V-RAE (EUPE-B)	256 × 256, 20 → 17	DiT^{DH} XL/1	1280	125.98	24.77
V-RAE (V-JEPA 2.1-L)	256 × 256, 20 → 17	DiT^{DH} XL/1	1280	**117.86**	**19.16**

* *Note:* DiT^{DH} uses RAE's wide DDT head (Zheng et al., 2025b). 20 → 17 denotes cropping 20 generated frames to the first 17 for evaluation.

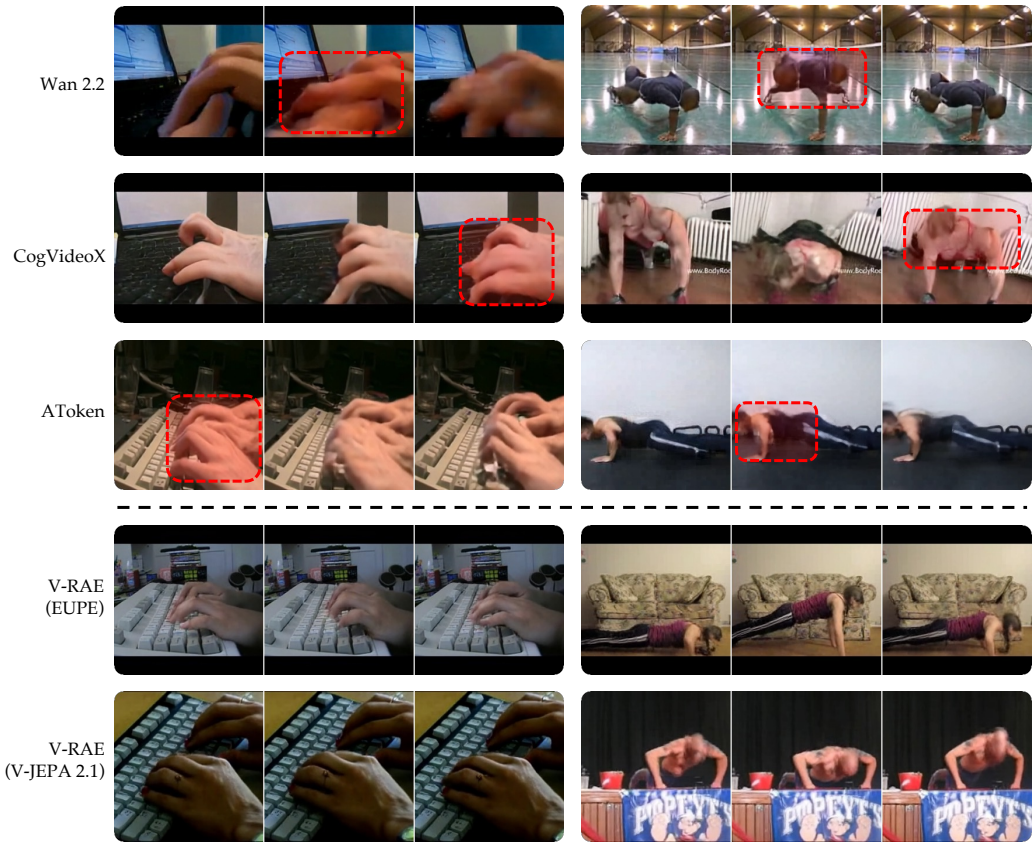


Figure 6: Qualitative comparison of class-conditional video generation on UCF101.

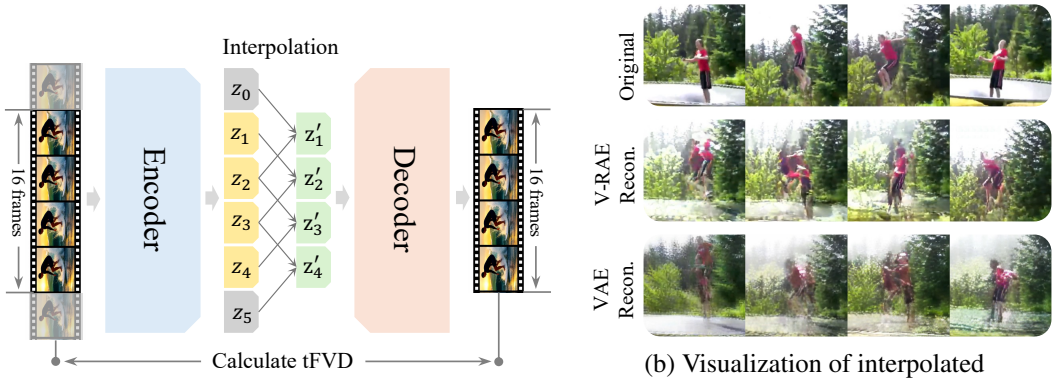
$[\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{L+1}]$ with temporal slices $\boldsymbol{z}_i \in \mathbb{R}^{C\times h\times w}$ at time step i, we interpolate its interior maps as:

$$\boldsymbol{z}'_t = 0.5\,(\boldsymbol{z}_{t-1} + \boldsymbol{z}_{t+1})\,, \qquad t = 1, \ldots, L. \tag{3}$$

With $\boldsymbol{Z}' = [\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_L]$, we compute FVD (Unterthiner et al., 2018) between the decoded videos and corresponding real clips:

$$\mathrm{tFVD} = \mathrm{FVD}\,(\{\boldsymbol{X}\}, \{\mathcal{D}(\boldsymbol{Z}')\})\,. \tag{4}$$

See Appendices D and E for motivation and protocol. Unlike rFVD, tFVD evaluates interpolated latents, which may leave decodable regions along curved or discontinuous trajectories, causing ghosting or flicker. Lower tFVD suggests better support for coherent interpolation, which may also help decode approximate generator predictions. Figure 7(b) shows ghosting around the moving subject in both methods, with more severe blurring and greater loss of fine detail in the VAE reconstructions.



(a) tFVD computation

(b) Visualization of interpolated reconstructions

Figure 7: Overview of tFVD, computed against real clips after decoding latents interpolated from temporal neighbor midpoints.

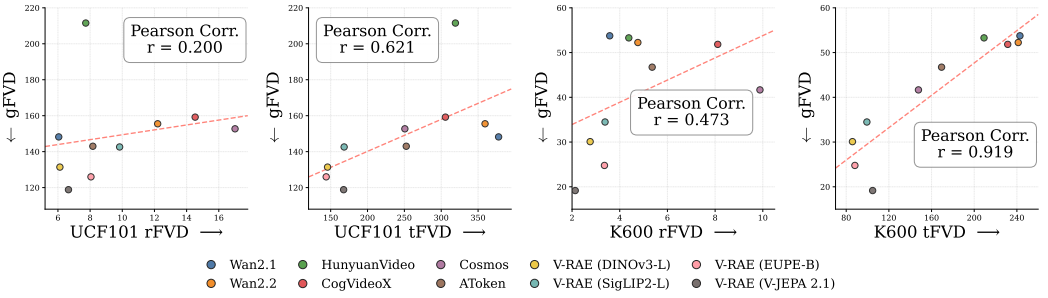


Figure 8: Metric-correlation analysis on UCF101 and K600, comparing reconstruction fidelity and temporal interpolation consistency against downstream generation quality.

Table 4: **Future video prediction results on Cityscapes at 40k training steps.** Both methods use the same conditional DiT and training budget.

Latent space	rFVD↓	tFVD↓	gFID↓	gFVD↓
Wan2.2 VAE	**7.03**	319.02	15.02	144.47
V-RAE (EUPE-B)	29.29	**224.60**	**11.52**	**111.36**

Predicting Generation Performance. Figure 8 reports stronger Pearson correlations with gFVD for tFVD than for rFVD on both UCF101 ($r = 0.621$ vs. 0.200) and K600 ($r = 0.919$ vs. 0.473). This consistent association suggests that temporal interpolation consistency is a more informative diagnostic of generative utility than reconstruction fidelity alone.

FINDING 3 **Generation-friendly latents require more than reconstruction fidelity.**
tFVD evaluates decoded temporal interpolations and exhibits stronger Pearson correlations with gFVD than rFVD across the evaluated tokenizers on both benchmarks.

4.4 Future Video Prediction in Semantic Latent Space

Beyond reconstruction and generation, we next examine future video prediction, which requires coherent scene evolution conditioned on observed frames. We evaluate on Cityscapes (Cordts et al., 2016), comparing V-RAE (EUPE-B) with Wan2.2 VAE under matched predictor and training settings. The prediction formulation and full protocol are provided in Appendix B.4.

Results and Analysis. As shown in Table 4, under matched settings, V-RAE achieves lower gFID (11.52 vs. 15.02) and gFVD (111.36 vs. 144.47), despite higher rFVD (29.29 vs. 7.03). Its lower tFVD (224.60 vs. 319.02) is consistent with the importance of latent temporal interpolation decodability beyond reconstruction fidelity (Section 4.3). Qualitatively, Figure 9 and Appendix Figure 15 show that V-RAE better preserves dynamic-object contours and spatial relationships, whereas Wan2.2 increasingly blurs and distorts them at longer prediction horizons.



Figure 9: Future video prediction on a vehicle-dominated Cityscapes scene.

FINDING 4 **Semantic latents form a directly decodable predictive state space.**
A predictor learns transitions between semantic states, while the same frozen decoder renders predicted states into pixels, unifying future-state prediction and visual reconstruction within a single latent interface.

5 Related Work

Representation Learning for Generation. Pretrained representations support generation through semantic tokenization (Ma et al., 2025a; Wu et al., 2025b), diffusion feature alignment or entanglement (Yu et al., 2024; Leng et al., 2025; Wu et al., 2025a), or direct semantic latent modeling (Zheng et al., 2025b; Singh et al., 2026; Tong et al., 2026). DeRA (Guo et al., 2025) aligns video tokenizer features; VideoREPA (Zhang et al., 2025) aligns spatiotemporal token relations. Divot (Ge et al., 2025) uses diffusion decoding; SemanticGen (Bai et al., 2025) generates semantic features to condition a second diffusion model in VAE space. V-RAE directly models the distribution of pretrained semantic latents for video generation, rather than using semantic features as alignment targets.

Video Autoencoders. Video generators (Blattmann et al., 2023) use continuous VAEs (Wan et al., 2025; Kong et al., 2024; Yang et al., 2025; Wang et al., 2024b), discrete tokenizers (Yu et al., 2023; Wang et al., 2024a; Agarwal et al., 2025), and query-based tokenizers (Wang et al., 2025; Tan et al., 2025; Xiong et al., 2026). V-RAE compresses frozen VFM representations with a temporal pooler and trains the pooler and video decoder jointly for reconstruction. This preserves pretrained semantic structure while learning coherent video decoding. Appendix A provides further discussion.

6 Conclusion

We presented V-RAE, which combines frozen visual representations with temporal pooling and spatiotemporal decoding. Across four encoders, V-RAE preserves rich semantics, achieves competitive reconstruction rFVD, and outperforms all evaluated VAE baselines in class-conditional generation with up-to 6× faster convergence. Its benefits extend to future world prediction. Our tFVD analysis further shows that temporal interpolation decodability is more indicative of generation quality than reconstruction fidelity alone. These findings motivate designing video latent spaces that preserve semantic structure and support temporally coherent decoding for both generation and prediction.

AI USE STATEMENT

We used generative AI tools to assist with writing, optimizing, and testing research code, as well as interpreting a limited subset of experimental results. We additionally used these tools to assist with drafting and revising the manuscript, improving English expression and organization, identifying relevant literature, and editing LaTeX source files, including checking cross-references and formatting.

The authors reviewed and revised the AI-assisted manuscript text, including the wording of scientific claims and the correspondence between model variants and reported results. The authors take full responsibility for the final manuscript and associated research artifacts, including the research code, experimental results, analyses, and references.

REPRODUCIBILITY STATEMENT

We provide detailed architecture, training, and evaluation specifications in the appendix and anonymized codebase to support reproducibility. These include model configurations, reconstruction objectives, optimization settings, downstream DiT architectures, training schedules, sampling procedures, and benchmark-specific evaluation protocols.

The anonymized implementation includes model definitions, experiment configurations, distributed training scripts, inference code, and evaluation pipelines, together with instructions for environment setup, dataset preparation, and pretrained-model dependencies. All datasets and pretrained representation encoders used in this work are publicly available.

To preserve submission anonymity, pretrained V-RAE checkpoints are not included in the anonymous artifact. We will publicly release the checkpoints required to reproduce the reported results after the anonymous review period.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 4, 9, 19

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. CoVLA: Comprehensive vision-language-action dataset for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1933–1943, 2025. 21

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 16

Jianhong Bai, Xiaoshi Wu, Xintao Wang, Xiao Fu, Yuanxing Zhang, Qinghe Wang, Xiaoyu Shi, Menghan Xia, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Kun Gai. SemanticGen: Video generation in semantic space. *arXiv preprint arXiv:2512.20619*, 2025. 9, 15

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 9

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 4

Bowei Chen, Sai Bi, Hao Tan, He Zhang, Tianyuan Zhang, Zhengqi Li, Yuanjun Xiong, Jianming Zhang, and Kai Zhang. Aligntok: Aligning visual foundation encoders to tokenizers for diffusion models. *arXiv preprint arXiv:2509.25162*, 2025. 15

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016. 8, 21

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

Yuying Ge, Yizhuo Li, Yixiao Ge, and Ying Shan. Divot: diffusion powers video tokenizer for comprehension and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13606–13617, 2025. 9, 15

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014. 19

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, 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 *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5842–5850, 2017. 4

Pengbo Guo, Junke Wang, Zhen Xing, Chengxu Liu, Daoguo Dong, Xueming Qian, and Zuxuan Wu. Dera: Decoupled representation alignment for video tokenization. *arXiv preprint arXiv:2512.04483*, 2025. 9, 15

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. 3

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 21

Sungwon Hwang, Hyojin Jang, Kinam Kim, Minho Park, and Jaegul Choo. Cross-frame representation alignment for fine-tuning video diffusion models. *arXiv preprint arXiv:2506.09229*, 2025. 15

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. DINO-foresight: Looking into the future with DINO. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 16

Efstathios Karypidis, Spyros Gidaris, and Nikos Komodakis. Representations before pixels: Semantics-guided hierarchical video prediction. *arXiv preprint arXiv:2604.11707*, 2026. 16

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 1

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2, 4, 9

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18262–18272, 2025. 2, 9, 15

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 4

Jiasen Lu, Liangchen Song, Mingze Xu, Byeongjoo Ahn, Yanjun Wang, Chen Chen, Afshin Dehghan, and Yinfei Yang. Atoken: A unified tokenizer for vision. *arXiv preprint arXiv:2509.14476*, 2025. 4, 19

Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-MAGVIT2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 4

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. *Advances in Neural Information Processing Systems*, 38:129274–129297, 2025a. 2, 9

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025b. 1, 4

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mahmoud Assran, Koustuv Sinha, Michael Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. 2, 3

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. 4

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 2, 3, 5

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026. 2, 3, 4, 9, 15, 21, 24

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025. 2, 16

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 2, 3

Zhentao Tan, Ben Xue, Jian Jia, Junhao Wang, Wencai Ye, Shaoyun Shi, Mingjie Sun, Wenjin Wu, Quan Chen, and Peng Jiang. Sweettok: Semantic-aware spatial-temporal tokenizer for compact video discretization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23541–23550, 2025. 9

Team Wan. Wan2.2. Official model and code release, 2025. Released July 28, 2025; accessed September 10, 2026. 4, 21

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026. 2, 9, 15

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10078–10093, 2022. 19

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 2, 3

Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing Generative Adversarial Networks Under Limited Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7921–7931, 2021. 19

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7, 21

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 4, 9, 15

Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. In *International Conference on Learning Representations*, volume 2025, pp. 60671–60697, 2025. 2, 4, 9, 15

Junke Wang, Yi Jiang, Zehuan Yuan, Binyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024a. 2, 4, 9

Yuchi Wang, Junliang Guo, Xinyi Xie, Tianyu He, Xu Sun, and Jiang Bian. Vidtwin: Video vae with decoupled structure and dynamics. *arXiv preprint arXiv:2412.17726*, 2024b. 9, 15

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 4

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *Advances in Neural Information Processing Systems*, 38:7714–7743, 2025a. 9

Shengqiong Wu, Hao Fei, Xiangtai Li, Jiayi Ji, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Towards semantic equivalence of tokenization in multimodal llm. In *International Conference on Learning Representations*, volume 2025, pp. 93334–93358, 2025b. 2, 9

Tianwei Xiong, Jun Hao Liew, Zilong Huang, Zhijie Lin, Jiashi Feng, and Xihui Liu. Evatok: Adaptive length video tokenization for efficient visual autoregressive generation. *arXiv preprint arXiv:2603.12267*, 2026. 2, 9, 15

Tongda Xu, Mingwei He, Shady Abu-Hussein, Jose Miguel Hernandez-Lobato, Chunhang Zheng, Kai Zhao, Chao Zhou, Ya-Qin Zhang, and Yan Wang. Making reconstruction fid predictive of diffusion generation fid. *arXiv preprint arXiv:2603.05630*, 2026. 2, 16, 22

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025. 2, 4, 9, 15

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 2025. 2, 15

Junqiu Yu, Pandeng Li, Yikai Wang, Jiaxing Zhao, Yujie Wei, Kaixun Jiang, Quanhao Li, Hongtao Yu, Zhihang Liu, Zhaohe Liao, et al. Affinetok: Semantic affine consistency for diffusion-friendly visual tokenizer. *arXiv preprint arXiv:2608.23864*, 2026. 16

Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2, 4, 9, 15

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. *arXiv preprint arXiv:2410.06940*, 2024. 2, 9, 15, 21

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. 4, 19

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. *Advances in Neural Information Processing Systems*, 38:122647–122676, 2025. 2, 9, 15

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable Augmentation for Data-Efficient GAN Training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7559–7570, 2020. 19

Anlin Zheng, Xin Wen, Xuanyang Zhang, Chuofan Ma, Tiancai Wang, Gang Yu, Xiangyu Zhang, and Xiaojuan Qi. Vision foundation models as effective visual tokenizers for autoregressive image generation. *arXiv preprint arXiv:2507.08441*, 2025a. 15

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025b. 2, 4, 7, 9, 15, 19, 21

Tianxiong Zhong, Xingye Tian, Boyuan Jiang, Xuebo Wang, Xin Tao, Pengfei Wan, and Zhiwei Zhang. Vfrtok: Variable frame rates video tokenizer with duration-proportional information assumption. *Advances in Neural Information Processing Systems*, 38:3325–3345, 2025. 15

Tianxiong Zhong, Xingye Tian, Xuebo Wang, Xin Tao, and Pengfei Wan. Diffusing in the right space: A systematic study of latent diffusability. *arXiv preprint arXiv:2606.03578*, 2026. 16

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 13–19 Jul 2025. 16

Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, and Vikas Chandra. Efficient universal perception encoder. *arXiv preprint arXiv:2603.22387*, 2026. 2, 3, 21

APPENDIX CONTENTS

A Extended Related Work **15**

B Implementation Details **16**

B.1 V-RAE Architecture 16

B.2 Reconstruction Training Recipe 18

B.3 Generation Training Recipe 19

B.4 Future Video Prediction Setup 21

C Effects of $\mathrm{DiT}^{\mathrm{DH}}$ and Auxiliary Supervision **21**

D Motivation and Interpretation of tFVD **22**

E Evaluation Protocols **23**

F Additional Qualitative Results **25**

F.1 Class-Conditional Generation 25

F.2 Future Video Prediction 26

G Limitations **27**

A EXTENDED RELATED WORK

Semantic Representations for Visual Generation. Pretrained visual features can guide generative training. VA-VAE and AlignTok use VFM supervision for image tokenizers (Yao et al., 2025; Chen et al., 2025), while DeRA aligns video tokenizer features with VFMs (Guo et al., 2025). REPA, VideoREPA, and CREPA use pretrained features or their relations to supervise diffusion models (Yu et al., 2024; Zhang et al., 2025; Hwang et al., 2025), while REPA-E jointly optimizes the tokenizer and generator (Leng et al., 2025).

A more direct route uses semantic features as generative representations. VFMTok quantizes frozen VFM features for autoregressive image generation (Zheng et al., 2025a), while RAE and its extensions model frozen features directly with diffusion (Zheng et al., 2025b; Tong et al., 2026; Singh et al., 2026). For video, Divot learns representations with a diffusion decoder (Ge et al., 2025), whereas SemanticGen generates semantic features to condition diffusion in VAE space (Bai et al., 2025). V-RAE temporally compresses native patch features from frozen image or video VFMs, retaining their spatial structure and semantics. It models these latents directly for video generation and future prediction, then decodes them to pixels without an intermediate VAE space.

Video Tokenization and Temporal Compression. Compared with image tokenizers, video tokenizers need temporal compression to reduce downstream generation costs. CogVideoX, Wan, and MAGVIT-v2 retain 3D latent grids with explicit spatial layout and temporal order (Yang et al., 2025; Wan et al., 2025; Yu et al., 2023). Query-based tokenizers aggregate patches into one-dimensional sequences with flexible token allocation (Wang et al., 2025; Zhong et al., 2025; Xiong et al., 2026). Query pooling can retain spatial structure: VidTwin applies a temporal Q-Former at each spatial location, with separate structure and dynamics branches (Wang et al., 2024b). V-RAE retains the spatiotemporal grid and high channel dimensionality of frozen VFM patch features. Its lightweight attention pooler reduces redundancy within local temporal windows at each spatial location, supporting accurate reconstruction and semantic preservation with few additional parameters.

Evaluating Latent Diffusability. High reconstruction fidelity does not guarantee high generation quality. This motivates evaluating latent *diffusability*, or how easily diffusion models learn the latent distribution. Direct evaluation compares generation quality and convergence under controlled conditions but requires training a generator for each tokenizer (Skorokhodov et al., 2025). Proxies include semantic separability, spatial structure, spectral statistics, and Velocity Irreducible Variance (VIV), which estimates uncertainty in flow-matching velocity targets (Zhong et al., 2026). Interpolation provides another proxy: iFID measures FID on decoded nearest-neighbor latent interpolations without training a generator (Xu et al., 2026). AffineTok compares semantic features of decoded latent interpolations with interpolated features of the endpoint images (Yu et al., 2026). V-RAE's tFVD evaluates decoded temporal interpolations within a video, complementing reconstruction metrics.

Feature Prediction and Future Video Generation. DINO-Foresight predicts VFM features for future scene understanding (Karypidis et al., 2025), while DINO-WM learns action-conditioned feature dynamics for planning (Zhou et al., 2025). V-JEPA 2 combines large-scale video pretraining with action-conditioned post-training (Assran et al., 2025). These studies emphasize understanding and control; video generation additionally requires recovering pixels from predicted features. Re2Pix conditions video diffusion in VAE space on predicted semantic features (Karypidis et al., 2026). V-RAE uses one temporally compressed semantic latent space for generation and future video prediction, with a shared decoder mapping generated and predicted latents to pixels.

B Implementation Details

We provide detailed model and training configurations for reproducibility. Section B.1 describes the V-RAE architecture and temporal pooling designs, Section B.2 specifies the reconstruction recipe and shared VideoMAE discriminator, and Section B.3 reports the generation architecture, training, sampling, and evaluation settings.

B.1 V-RAE Architecture

Table 5: VFM-dependent V-RAE architecture. Block indices are zero-based; shared pooler and decoder details are described in text.

Configuration	DINOv3-L	SigLIP2-L	EUPE-B	V-JEPA 2.1-L
Encoder settings				
Backbone (blocks / width)	ViT-L/16 (24 / 1024)	ViT-L/16 (24 / 1024)	ViT-B/16 (12 / 768)	ViT-L/16 (24 / 1024)
Selected feature blocks	11,13,15,17,19,21,23	11,13,15,17,19,21,23	5,6,7,8,9,10,11	11,13,15,17,19,21,23
Patch / encoder tubelet	16 / 1	16 / 1	16 / 1	16 / 2
Input norm.	ImageNet-style	Native processor	ImageNet-style	ImageNet-style
Encoder temporal length	T	T	T	$T/2$
Decoder settings				
Temporal group size	4	4	4	2
Latent width C	1024	1024	768	1024
Decoder initialization	RAEv2 image decoder	Scratch	RAEv2 image decoder	Scratch
Decoder attention mask	Chunk-causal	Chunk-causal	Chunk-causal	Full (non-causal)
Trainable Params		$\sim$420M		

Representation encoders. We instantiate V-RAE with DINOv3-L, SigLIP2-L, EUPE-B, and V-JEPA 2.1-L and keep the representation encoder frozen throughout reconstruction training. DINOv3-L, SigLIP2-L, and V-JEPA 2.1-L use 1024-dimensional features, whereas EUPE-B uses 768-dimensional features. The three image encoders preserve the input temporal length, while the tubelet size of two in V-JEPA 2.1-L reduces it by half. We therefore pool groups of four image-encoder features or two V-JEPA features, producing the same temporal compression ratio of four for all variants. A 16-frame, 256×256 clip is represented by a $4 \times 16 \times 16$ latent grid with channel width C given in Table 5.

Temporal attention pooler and non-affine normalization. For completeness, we give the multi-head formulation summarized by Equation 2. After flattening the spatial grid, let $p \in \{1, \ldots, hw\}$

index a spatial position. The VFM's patch tokens in the window for output time t are $\boldsymbol{f}_i = \boldsymbol{f}_{r_{\text{pool}}t+i,p} \in \mathbb{R}^C$ for $i = 0, \ldots, r_{\text{pool}} - 1$. For attention head m, the keys and values are

$$\boldsymbol{k}_i^{(m)} = \boldsymbol{W}_K^{(m)} \boldsymbol{f}_i, \qquad \boldsymbol{v}_i^{(m)} = \boldsymbol{W}_V^{(m)} \boldsymbol{f}_i. \tag{5}$$

The shared query $\boldsymbol{q}^{(m)} \in \mathbb{R}^d$ and local temporal bias $\beta_i^{(m)}$ produce

$$\alpha_i^{(m)} = \text{Softmax}_i\left(\frac{(\boldsymbol{q}^{(m)})^\top \boldsymbol{k}_i^{(m)}}{\sqrt{d}} + \beta_i^{(m)}\right), \qquad \boldsymbol{u}^{(m)} = \sum_{i=0}^{r_{\text{pool}}-1} \alpha_i^{(m)} \boldsymbol{v}_i^{(m)}, \tag{6}$$

where $d = C/M$ for M heads. The pooled token is

$$\boldsymbol{z}_{t,p} = \text{Norm}\Big(\boldsymbol{W}_O \,\text{Concat}([\boldsymbol{u}^{(m)}]_{m=1}^{M})\Big). \tag{7}$$

This operation is applied independently at each spatial position and therefore introduces no additional spatial mixing. We initialize the projections to preserve their inputs and initialize the queries and temporal biases to zero, yielding uniform weights $\alpha_i^{(m)} = 1/r_{\text{pool}}$ at the start of training.

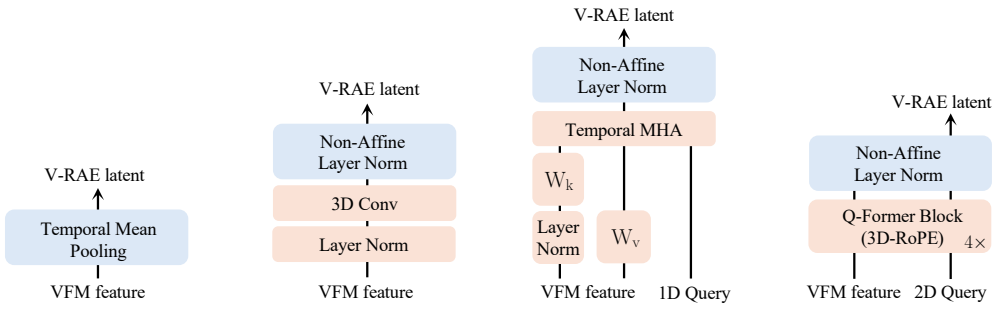
Let $\tilde{\boldsymbol{z}}$ denote the pooled feature. Before latent-noise augmentation, we apply LayerNorm without a learned scale or bias:

$$\boldsymbol{z}_{\text{clean}} = \text{LN}_{\text{na}}(\tilde{\boldsymbol{z}}), \qquad \boldsymbol{z}_{\text{train}} = \boldsymbol{z}_{\text{clean}} + \sigma\epsilon. \tag{8}$$

This choice is important for noise-augmented reconstruction training. With an affine LayerNorm, $\boldsymbol{z}_{\text{clean}} = \gamma \odot \text{LN}(\tilde{\boldsymbol{z}}) + \beta$, the optimizer can increase γ while the injected noise magnitude σ remains fixed. The effective signal-to-noise ratio then grows approximately as γ^2/σ^2, providing a shortcut that reduces the relative corruption without requiring the decoder to become robust to noise. We therefore fix $\gamma = 1$ and $\beta = 0$. This removes the rescaling shortcut, keeps the latent scale calibrated, and forces the decoder to reconstruct under the intended noise level. With identity-initialized projections and zero-initialized queries and temporal biases, the pooler is initialized close to temporal mean pooling, followed by this fixed normalization.

V-RAE decoder. All variants share a ViT-XL video decoder with width 1152, 28 transformer blocks, 16 attention heads, and a 4096-dimensional GELU feed-forward network. The decoder combines a learned 2D spatial embedding with 3D RoPE ($\theta = 10{,}000$). The DINOv3-L, SigLIP2-L, and EUPE-B reconstruction models use chunk-causal attention: tokens interact freely within a latent chunk, while each chunk can attend only to itself and earlier chunks. The V-JEPA 2.1-L model instead uses non-causal full self-attention over all spatiotemporal latent tokens, consistent with the full temporal context provided by its frozen encoder. Each output token predicts a $4 \times 16 \times 16 \times 3 = 3072$-dimensional RGB tubelet, which is rearranged into four consecutive frames. For DINOv3-L and EUPE-B, compatible transformer weights are initialized from the corresponding RAEv2 image decoder, and the image prediction head is inflated along time; the other two decoders are trained from scratch.

Temporal pooling designs. We compare four temporal pooling designs for reconstruction and semantic preservation, fixing the VFM (DINOv3-L), video decoder, training data, compression ratio, and reconstruction objective. Each maps a non-overlapping temporal group to one latent feature map on the original spatial grid. Figures 10 and 11 present their architectures and UCF101 results, respectively.



(a) Mean (b) 3D convolution (c) Temporal attention (d) Q-Former

Figure 10: Four temporal pooling designs with matched output shapes and compression ratios, differing in their within-group feature aggregation.

Mean pooling averages features without learned transformations, whereas 3D convolution learns a shared local spatiotemporal filter. These designs illustrate the reconstruction–semantics trade-off: mean pooling retains the highest probing accuracy (91.82%) but reconstructs poorly (94.25 rFVD), while convolution achieves the lowest rFVD (5.91) with substantially lower probing accuracy (78.26%). Thus, preserving the feature basis alone is insufficient for reconstruction, and improving reconstruction does not necessarily preserve semantic information.

Q-Former pooling adds capacity through a learned 2D query grid and four transformer blocks with 3D RoPE, allowing repeated spatial and temporal feature transformations. Despite its 67M parameters, it reaches only 6.87 rFVD and 84.37% probing accuracy, improving semantic retention over convolution but sacrificing reconstruction quality. Greater resampling capacity therefore does not by itself resolve the trade-off.

Temporal attention instead adapts temporal weights to the input while aggregating independently at each spatial position. It uses one shared query per head rather than a separate query grid, with the mean-pooling initialization and non-affine normalization described above.

With only 3M parameters, it achieves 6.12 rFVD and 89.13% probing accuracy: relative to convolution, it gains 10.87 percentage points in accuracy for a 0.21 increase in rFVD, and it outperforms Q-Former on both metrics. This balance supports our choice of temporal attention as the default pooler.

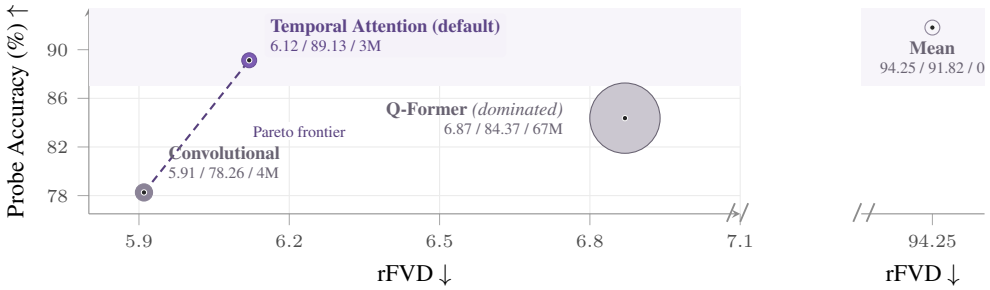


Figure 11: Temporal pooling trade-offs on the UCF101 test set (broken x-axis). Labels give rFVD / probing accuracy / pooler parameters. Bubble area is proportional to parameter count; the unfilled bubble denotes parameter-free mean pooling.

B.2 Reconstruction Training Recipe

Reconstruction training. We jointly train the temporal pooler and video decoder while keeping the representation encoder frozen in evaluation mode. Training uses the union of UCF101 train split 1 and the K600 training split. We sample 16-frame clips at an interval of 3 from a random temporal position, resize the short side to 256, and apply a random 256×256 crop and horizontal

flip. DINOv3-L and EUPE-B are trained for 40 epochs, while SigLIP2-L and V-JEPA 2.1-L are trained for 50 epochs. The remaining shared training hyperparameters are listed in Table 6.

Table 6: Shared reconstruction training hyperparameters.

Configuration	Value	Configuration	Value
Global batch size	128	Gradient clipping	Global norm 1.0
Optimizer	AdamW	Precision	BF16
Learning rate	2×10^{-4}	EMA decay	0.9995
Betas	$(0.9, 0.95)$	L1 loss weight	1
Epsilon	1×10^{-8}	LPIPS loss weight	1
Weight decay	0	GAN generator loss weight	0.3
Learning-rate schedule	Constant	Gram loss weight	100

Reconstruction objective. For each clip, the frozen encoder and temporal pooler produce a clean latent $\boldsymbol{z}_{\text{clean}}$. We perturb it during reconstruction training as $\boldsymbol{z}_{\text{train}} = \boldsymbol{z}_{\text{clean}} + \sigma\epsilon$, where $\sigma \sim \mathcal{U}(0, 0.8)$ and $\epsilon \sim \mathcal{N}(0, I)$. The video decoder is optimized with an L_1 term, LPIPS (Zhang et al., 2018), an adversarial objective (Goodfellow et al., 2014), and a Gram-matrix loss following recent video autoencoders (Lu et al., 2025; Agarwal et al., 2025):

$$\mathcal{L}_{\text{recon}} = \lambda_1 \mathcal{L}_1 \; + \; \lambda_{\text{lpips}} \mathcal{L}_{\text{LPIPS}} \; + \; \lambda_{\text{gan}} \mathcal{L}_{\text{GAN}} \; + \; \lambda_{\text{gram}} \mathcal{L}_{\text{Gram}}. \tag{9}$$

The loss weights are listed in Table 6. LPIPS is evaluated at the native 256×256 resolution, whereas the Gram branch uses 224×224 inputs; both perceptual branches cover all 16 frames. The latent perturbation is used only as a reconstruction-training augmentation; evaluation, latent-statistics estimation, and latent video generation all use clean latents.

Table 7: Shared VideoMAE discriminator configuration.

Configuration	Value
Backbone	Frozen VideoMAE-B
Feature taps	Embedding output and blocks 2, 5, 8, 11
Temporal head	Conv1d, kernel size 9
Normalization	Local batch normalization + spectral normalization
Adversarial start	Epoch 30
Update interval	One discriminator step per five generator steps
Optimizer	AdamW
Learning rate	2×10^{-5}
Learning-rate schedule	Constant
Betas	$(0.9, 0.95)$
Weight decay	10^{-4}

Adversarial objective. Beginning at epoch 30, we add the same adversarial objective based on VideoMAE (Tong et al., 2022) for all four variants. The VideoMAE-B backbone remains frozen, and only the temporal Conv1d heads are optimized. We use a non-saturating generator loss and a smoothed non-saturating discriminator loss, both weighted by 0.3, and update the discriminator once every five generator steps. Translation, color, and cutout DiffAug (Zhao et al., 2020) are applied with probability 1.0 and cutout ratio 0.2. We additionally use LeCam regularization (Tseng et al., 2021) with weight 0.001 and logit-EMA decay 0.9995.

B.3 Generation Training Recipe

DiT architecture. We use a high-capacity DiT to reduce the risk that limited generator expressiveness becomes the dominant bottleneck. Our output-head choice follows Zheng et al. (2025b), who report that the wide DDT head improves DINOv2-based RAE generation but worsens SD-VAE FID

under their evaluated settings. Table 8 summarizes V-RAE's generator architecture and task-specific configurations for UCF101, K600, and Cityscapes.

For UCF101 and K600, all four representation encoders use the same class-conditional DiT backbone: a 28-block, 1536-dimensional transformer encoder and a two-block, 2048-dimensional output decoder. Only the input channel count and corresponding time shift change for EUPE-B. Cityscapes retains the 28-block depth but uses a 1152-dimensional encoder with 16 attention heads of dimension 72 and EUPE-B latents with 768 channels. It uses a clean-context/noisy-future token sequence; the output decoder predicts only the future tokens. All variants use full self-attention, per-head Q/K RMSNorm, 3D RoPE, AdaLN-Zero conditioning, RMSNorm, and SwiGLU.

Table 8: Latent video DiT configurations for UCF101, K600, and Cityscapes.

Configuration	UCF101 generation	K600 generation	Cityscapes prediction
Input and token geometry			
RGB input	20×256^2, interval 3	20×256^2, interval 3	Frames 4–15 $\rightarrow$ 16–27, 432×768
Clean latent grid	$5 \times 16 \times 16 \times C$	$5 \times 16 \times 16 \times C$	$3 \times 27 \times 48 \times 768$ each
Encoder / output tokens	1280 / 1280	1280 / 1280	7776 / 3888
Latent patch size	1×1	1×1	1×1
DiT encoder and output decoder			
Main encoder width / blocks	1536 / 28	1536 / 28	1152 / 28
Main encoder heads / head dim	24 / 64	24 / 64	16 / 72
Output decoder width / blocks	2048 / 2	2048 / 2	2048 / 2
Output decoder heads / head dim	16 / 128	16 / 128	16 / 128
Self-attention	Full spatiotemporal	Full spatiotemporal	Full spatiotemporal
Positional encoding	3D RoPE ($\theta = 10{,}000$)	3D RoPE, ($\theta = 10{,}000$)	3D RoPE, ($\theta = 10{,}000$)
Block conditioning	AdaLN-Zero	AdaLN-Zero	AdaLN-Zero
Normalization / epsilon	RMSNorm / 10^{-6}	RMSNorm / 10^{-6}	RMSNorm / 10^{-6}
FFN / MLP ratio	SwiGLU / 4.0	SwiGLU / 4.0	SwiGLU / 4.0
Time embedding	256-D Gaussian Fourier	256-D Gaussian Fourier	256-D Gaussian Fourier
Auxiliary base branch	Encoder block 8	Encoder block 8	Encoder block 8
Trainable parameters	1.368B	1.369B	0.843B
Conditioning			
Condition	Class label	Class label	Clean context latent
Number of classes	101	600	–
Condition dropout	0.1	0.1	0.1
Context embedding	–	–	Dropout $\rightarrow$ mean pool $\rightarrow$ 2-layer MLP
Optimization			
Global batch size	64	128	8
Optimizer / learning rate	AdamW / 1×10^{-4}	AdamW / 1×10^{-4}	AdamW / 3×10^{-4}
Weight decay	0	0	0
LR schedule	Linear to 5×10^{-5}	Linear to 5×10^{-5}	Cosine to 10^{-4}
Warmup	10 epochs from zero	2 epochs from zero	1,490 updates from zero
Gradient clipping	Global norm 1.0	Global norm 1.0	Global norm 1.0
Precision	BF16	BF16	BF16
Training length / EMA	1700 epochs / 0.9995	100 epochs / 0.9995	40k steps / 0.9995
Flow matching			
Prediction parameterization	x-prediction	x-prediction	x-prediction
Time distribution	LogitNormal$(0, 1)$	LogitNormal$(0, 1)$	LogitNormal$(0, 1)$
Velocity denominator clamp t_ϵ	0.05	0.05	0.05
Full / base loss weights	1.0 / 1.0	1.0 / 1.0	1.0 / 1.0
Online sampling and evaluation			
Online sampler / steps	Euler / 100	Euler / 100	Euler / 100
Online CFG / internal guidance	1.0 / 1.3	1.0 / 1.2	1.0 / 1.2
Evaluation population	2048 clips	50,000 clips	500 clips
Video metric	17-frame gFVD	17-frame gFVD	12-frame gFID + gFVD

Training details. We freeze V-RAE during DiT training and compute latent statistics separately for each dataset and encoder. For Cityscapes, statistics use only future latents from the training split. Optimization settings are listed in Table 8.

For flow matching, given a normalized clean latent $\boldsymbol{x}_0$ and Gaussian noise $\boldsymbol{\epsilon}$, we form the interpolated latent

$$\boldsymbol{x}_t = (1-t)\cdot \boldsymbol{x}_0 + t\cdot \epsilon \ , \qquad t = \frac{s\cdot t_0}{1+(s-1)\cdot t_0} \ , \qquad t_0 \sim \mathrm{LogitNormal}(0,1)\ . \tag{10}$$

Following RAE (Zheng et al., 2025b), we set the time shift according to the video latent dimensionality:

$$s = \sqrt{\frac{T_Z \cdot h \cdot w \cdot C}{n}}, \qquad n = 4096. \tag{11}$$

The temporal factor T_Z extends the image-space shift to video latents. The DiT uses x-prediction, predicting the clean latent $\widehat{\boldsymbol{x}}_{0,\theta}$ and converting it to the velocity estimate

$$\widehat{\boldsymbol{v}}_\theta = \frac{\boldsymbol{x}_t - \widehat{\boldsymbol{x}}_{0,\theta}}{\max(t, t_\epsilon)}, \qquad t_\epsilon = 0.05. \tag{12}$$

The target is $\boldsymbol{v} = (\boldsymbol{x}_t - \boldsymbol{x}_0)/\max(t, t_\epsilon)$. Inspired by RAEv2 (Singh et al., 2026), we supervise the final head (full) and an auxiliary head after DiT block 8 (base), both using x-prediction. Their velocity-space losses have unit weights (constant normalization omitted):

$$\begin{gathered}\mathcal{L}^h_{\mathrm{RF}} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon},t}\left[\left\|\widehat{\boldsymbol{v}}^h_\theta - \boldsymbol{v}\right\|_2^2\right], \qquad h \in \{\mathrm{full}, \mathrm{base}\}, \\ \mathcal{L}_{\mathrm{DiT}} = \mathcal{L}^{\mathrm{full}}_{\mathrm{RF}} + \mathcal{L}^{\mathrm{base}}_{\mathrm{RF}}.\end{gathered} \tag{13}$$

No REPA-style feature-alignment loss (Yu et al., 2024) is used.

B.4 Future Video Prediction Setup

Let $\mathbf{X}^c$ and $\mathbf{X}^f$ denote the context and future clips, respectively. The prediction pipeline is

$$\mathbf{Z}^c = \mathcal{P}(\mathcal{E}(\mathbf{X}^c))\,, \qquad \mathbf{Z}^f = \mathcal{P}\big(\mathcal{E}\big(\mathbf{X}^f\big)\big)\,, \qquad \widehat{\mathbf{Z}}^f \sim p_\theta\big(\mathbf{Z}^f \mid \mathbf{Z}^c\big)\,, \qquad \widehat{\mathbf{X}}^f = \mathcal{D}\Big(\widehat{\mathbf{Z}}^f\Big)\,. \tag{14}$$

Here, $\mathbf{Z}^c$ and $\mathbf{Z}^f$ are the context and target future latents produced by the frozen VFM $\mathcal{E}$ and temporal pooler $\mathcal{P}$. The conditional latent DiT $p_\theta(\cdot \mid \cdot)$ predicts $\widehat{\mathbf{Z}}^f$ from $\mathbf{Z}^c$, and the frozen decoder $\mathcal{D}$ renders the predicted future clip $\widehat{\mathbf{X}}^f$. This setup evaluates whether a latent space supports predictable future dynamics and whether approximate predicted states remain reliably decodable.

We evaluate on Cityscapes (Cordts et al., 2016), conditioning on frames 4–15 and predicting frames 16–27 at 432×768. V-RAE uses EUPE-B (Zhu et al., 2026) as its representation encoder and is fine-tuned on CoVLA (Arai et al., 2025) for high-resolution video reconstruction. We use Wan2.2 VAE (Team Wan, 2025) as the conventional latent baseline. Both latent spaces share the same Transformer backbone, latent-token budget, training schedule, and sampling configuration, with only the latent-channel-dependent input and output projections adjusted. The complete architecture and optimization settings are reported in Table 8. We compute gFID (Heusel et al., 2017) over 6,000 future frames from 500 validation sequences and gFVD (Unterthiner et al., 2018) over the corresponding 500 12-frame videos; further evaluation details are provided in Section E.

C Effects of DiT^{DH} and Auxiliary Supervision

Despite prior evidence that DiT^{DH} (DiT with a wide DDT head) benefits RAE but not SD-VAE in image generation (Zheng et al., 2025b), and that intermediate supervision improves RAE generation (Singh et al., 2026), their effects on video generation require further examination. We therefore study how these choices affect the generation performance of V-RAE and Wan2.2 VAE on UCF101.

For both Wan2.2 VAE and V-RAE with V-JEPA2.1-L, we evaluate all combinations of standard DiT versus DiT^{DH} and training with versus without auxiliary supervision. Figure 12 shows this complete 2×2 comparison for each latent space. All runs share the DiT backbone, token budget, training budget, and evaluation protocol. Auxiliary supervision denotes the intermediate prediction loss in Eq. 13.

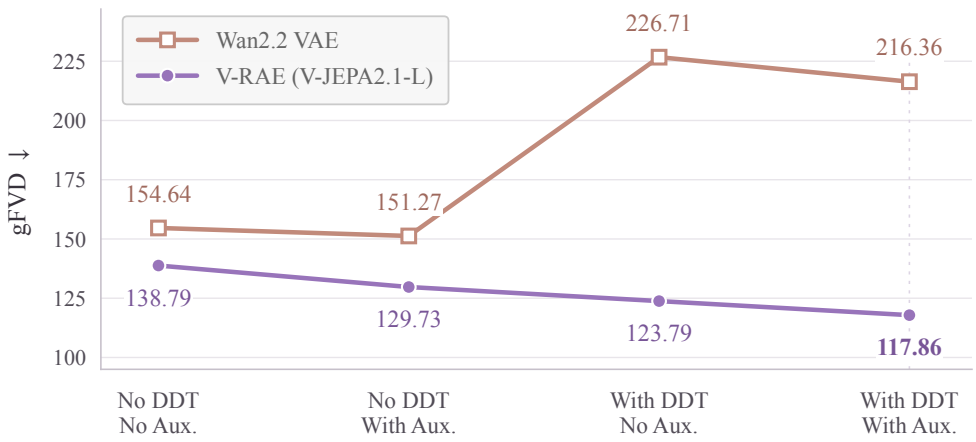


Figure 12: Effects of the wide DDT head and auxiliary supervision (Aux.) on UCF101 generation. V-RAE uses V-JEPA2.1-L. "No DDT" denotes standard DiT; "With DDT" denotes DiT^{DH}. Lines connect the four discrete configurations.

V-RAE benefits from both components. V-RAE's curve decreases across the four configurations, from 138.79 with neither component to 117.86 with both, a 15.1% reduction in gFVD. Comparisons that hold the other component fixed show that the wide DDT head reduces gFVD by 11.87–15.00, while auxiliary supervision yields smaller but consistent reductions of 5.93–9.06. Thus, either measure improves V-RAE generation on its own, and their combination gives the best result.

For VAE, auxiliary supervision helps but the wide head degrades generation. Wan2.2 VAE shows only a small initial improvement with auxiliary supervision, from 154.64 to 151.27 (2.2%), followed by much higher gFVD in the two configurations with the wide DDT head (226.71 and 216.36). At either auxiliary-supervision setting, the wide head worsens gFVD by 65.09–72.07. Auxiliary supervision still helps with both standard DiT and the wide head, reducing gFVD by 3.37 and 10.35, respectively, but these gains cannot compensate for the degradation caused by the wide head. Consequently, using both components leaves gFVD 39.9% higher than using neither, and standard DiT with auxiliary supervision remains the best VAE configuration. Together, these contrasting trends show that auxiliary supervision benefits both latent types, whereas the wide head benefits V-RAE but harms Wan2.2 VAE under the evaluated settings.

D Motivation and Interpretation of tFVD

This section explains the design and interpretation of tFVD. Inspired by iFID (Xu et al., 2026), we construct latent interpolations using temporal neighbors within a video. Computing tFVD requires no generator; its association with downstream generation quality is assessed empirically in Section 4.3.

Inspiration from iFID: evaluating interpolation quality. iFID connects diffusion generalization to local interpolation and composition of training samples. Under this view, these combinations can produce valid new samples, but they can also fall outside the data manifold. iFID evaluates this property by interpolating neighboring latent codes, decoding them, and computing FID against real images. This motivates evaluating more than reconstruction at encoded data points. We should also examine whether combinations of nearby codes can produce valid data. We extend this idea to video representations.

Challenges in video: local similarity does not ensure temporal correspondence. Interpolating between full video latents mixes changes in subjects, scenes, and motion. This makes it difficult to isolate the behavior of local temporal structure. Searching for neighbors at individual time steps raises another problem. A close match at one step may follow a different motion afterward. Independent matches may also come from different videos and fail to form a coherent sequence. Restricting

the search to the same time index does not ensure that the videos show the same stage of an action. We therefore seek interpolation endpoints with a clear temporal relationship.

Our approach: interpolate between temporal neighbors within a video. To address these challenges, we take a different approach and use the video's own temporal continuity to identify interpolation neighbors. Within a locally continuous segment, nearby time steps often share the same subject, scene, and similar semantic content. We therefore expect their latent codes to lie near one another. This provides a natural local neighborhood without searching across different videos. We use the two temporal neighbors of each target position as interpolation endpoints.

Given a latent sequence $\boldsymbol{Z} = [\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{L+1}]$, we replace each interior code with the midpoint of its two neighbors:

$$\widetilde{\boldsymbol{z}}_t = \frac{\boldsymbol{z}_{t-1} + \boldsymbol{z}_{t+1}}{2}, \qquad t = 1, \ldots, L. \tag{15}$$

Here, $\boldsymbol{z}_t$ denotes the latent features at one temporal position. It may represent a single frame or a temporally compressed group of frames. Using neighbors on both sides gives the interpolation a clear intermediate position. We can then examine whether replacing the original state with this combination still produces a plausible video.

We form the interpolated sequence

$$\widetilde{\boldsymbol{Z}} = [\widetilde{\boldsymbol{z}}_1, \ldots, \widetilde{\boldsymbol{z}}_L], \tag{16}$$

decode it as a whole, and compute FVD against the temporally aligned real clips:

$$\mathrm{tFVD} = \mathrm{FVD}\left(\{\mathcal{D}(\widetilde{\boldsymbol{Z}}_i)\}_{i=1}^{N}, \{\boldsymbol{X}_i^{\mathrm{aligned}}\}_{i=1}^{N}\right). \tag{17}$$

Here, $\mathcal{D}$ is the decoder, N is the number of evaluation clips, and $\boldsymbol{X}_i^{\mathrm{aligned}}$ is the real clip aligned with the decoded output. The metric compares the feature distributions of the two video sets. It does not directly measure distances between interpolated and original latent codes.

Interpretation: the quality of decoded temporal interpolations. tFVD tests whether combinations of neighboring temporal states can still decode into plausible videos. It does not assume that motion or latent trajectories are linear. Nor does it require exact recovery of each original intermediate state. A lower tFVD means that the decoded interpolations are closer to real videos in FVD feature statistics. This extends the interpolation-based evaluation of iFID to neighborhoods defined by a video's temporal structure. We assess its connection to generation quality through experiments.

Experimental evidence: the relationship with generation quality. The Pearson correlations between tFVD and gFVD are 0.621 on UCF101 and 0.919 on K600 (Figure 8). Both exceed the corresponding correlations for rFVD, which are 0.200 and 0.473. Across the evaluated tokenizers and settings, temporal interpolation quality provides information beyond reconstruction quality and shows a stronger relationship with downstream generation performance. These results support tFVD as a complementary diagnostic for evaluating video representations for generation.

E Evaluation Protocols

Unless stated otherwise, we evaluate EMA checkpoints. The rFVD, tFVD, and gFVD metrics use clip features from an I3D network pretrained on Kinetics-400, whereas gFID uses Inception-v3 pool3 features. All Fréchet means and covariances are accumulated in float64.

Reconstruction. We evaluate on the UCF101 test set and the K600 validation set. Each clip is taken from the beginning of a video with a temporal interval of 3; we resize the short side and apply a center crop to 256×256. V-RAE and AToken use 16-frame clips, while causal VAEs that encode the first frame separately use 17-frame clips, following Table 1. K600 contributes 27,874 clips after removing 36 videos that are too short; no temporal padding is used. LPIPS, PSNR, and SSIM are evaluated over all valid frames, and rFVD is computed between the reconstructed and real clip distributions.

Temporal Reconstruction Error Difference. Given ground-truth frames $\{\mathbf{x}_t\}_{t=1}^{T}$ and their reconstructions $\{\widehat{\mathbf{x}}_t\}_{t=1}^{T}$, let Ω_{ROI} denote a high-frequency region of interest (ROI). For each spatial location $\mathbf{p} \in \Omega_{\mathrm{ROI}}$, the channel-averaged reconstruction error is

$$e_t(\mathbf{p}) = \frac{1}{C} \sum_{c=1}^{C} \left| x_{t,c}(\mathbf{p}) - \widehat{x}_{t,c}(\mathbf{p}) \right|. \tag{18}$$

We summarize its temporal variation using a time-averaged spatial heatmap and a spatially averaged temporal curve:

$$\begin{aligned} \overline{\mathrm{TRED}}_{\mathrm{t}}(\mathbf{p}) &= \frac{1}{T-1} \sum_{t=1}^{T-1} \left| e_{t+1}(\mathbf{p}) - e_t(\mathbf{p}) \right|, \qquad \mathbf{p} \in \Omega_{\mathrm{ROI}}, \\ \overline{\mathrm{TRED}}_{\mathrm{ROI}}(t) &= \frac{1}{|\Omega_{\mathrm{ROI}}|} \sum_{\mathbf{p} \in \Omega_{\mathrm{ROI}}} \left| e_{t+1}(\mathbf{p}) - e_t(\mathbf{p}) \right|, \quad t = 1, \ldots, T-1. \end{aligned} \tag{19}$$

Lower values indicate more stable reconstruction errors across adjacent frames. As shown in Figure 3, both V-RAE variants exhibit markedly lower TRED than the frame-wise RAEv2 baseline on the presented examples and approach the stability of two VAEs. This localized diagnostic complements rFVD and indicates that V-RAE suppresses abrupt reconstruction-error fluctuations that appear as visible flicker.

Temporal interpolation. For tFVD, we construct a temporally ordered sequence of six clean latent codes $[\boldsymbol{z}_0, \ldots, \boldsymbol{z}_5]$ using each tokenizer's native temporal grouping. Frame indices below are zero-based and refer to the sampled input clip. VAEs that encode the first frame separately use 21 input frames: $\boldsymbol{z}_0$ encodes frame 0, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_4$ cover frames 1–16, and $\boldsymbol{z}_5$ covers frames 17–20. V-RAE uses 24 input frames, with each latent code corresponding to a four-frame group. We compute all four interior midpoints from the original codes:

$$\widetilde{\boldsymbol{Z}} = \left[\frac{\boldsymbol{z}_0 + \boldsymbol{z}_2}{2}, \frac{\boldsymbol{z}_1 + \boldsymbol{z}_3}{2}, \frac{\boldsymbol{z}_2 + \boldsymbol{z}_4}{2}, \frac{\boldsymbol{z}_3 + \boldsymbol{z}_5}{2} \right]. \tag{20}$$

For VAEs with separate first-frame encoding, we decode $[\boldsymbol{z}_0, \widetilde{\boldsymbol{z}}_1, \ldots, \widetilde{\boldsymbol{z}}_4]$ into 17 frames, discard the first decoded frame, and compare the remaining 16 frames with ground-truth frames 1–16. For V-RAE, we decode $\widetilde{\boldsymbol{Z}}$ directly into 16 frames and compare them with ground-truth frames 4–19. Thus, both protocols evaluate 16-frame outputs against their corresponding real temporal windows, using the same I3D implementation as rFVD.

Class-conditional generation. All tokenizers are evaluated on 17-frame 256×256 videos. V-RAE and AToken retain the first 17 of 20 generated frames; causal VAEs with separate first-frame encoding generate 17 directly. For gFVD, all tokenizers share the evaluation population and 100-step Euler sampler within each benchmark, using 2,048 generated clips on UCF101 and 50,000 on K600. V-RAE uses EMA weights, CFG 1.0, and internal guidance following RAEv2 (Singh et al., 2026), with a fixed scale per comparison.

Future prediction. For Cityscapes, the model conditions on frames 4-15 and predicts frames 16-27 at 432×768 and 16 fps. We evaluate all 500 validation sequences. The same 12-frame predictions are used for both metrics: gFID is computed over the resulting 6,000 frames, and gFVD over the 500 video clips. Sampling uses the EMA model, 100 Euler steps, CFG 1.0, and internal guidance 1.2 over $t \in [0.10, 1.0]$.

F Additional Qualitative Results

F.1 Class-Conditional Generation

Figure 13 provides the Kinetics-600 comparison corresponding to the UCF101 result in Figure 6. Figure 14 provides additional samples from the DINOv3-L and SigLIP2-L variants of V-RAE. Together, these results cover all four V-RAE representation encoders evaluated in our generation experiments.

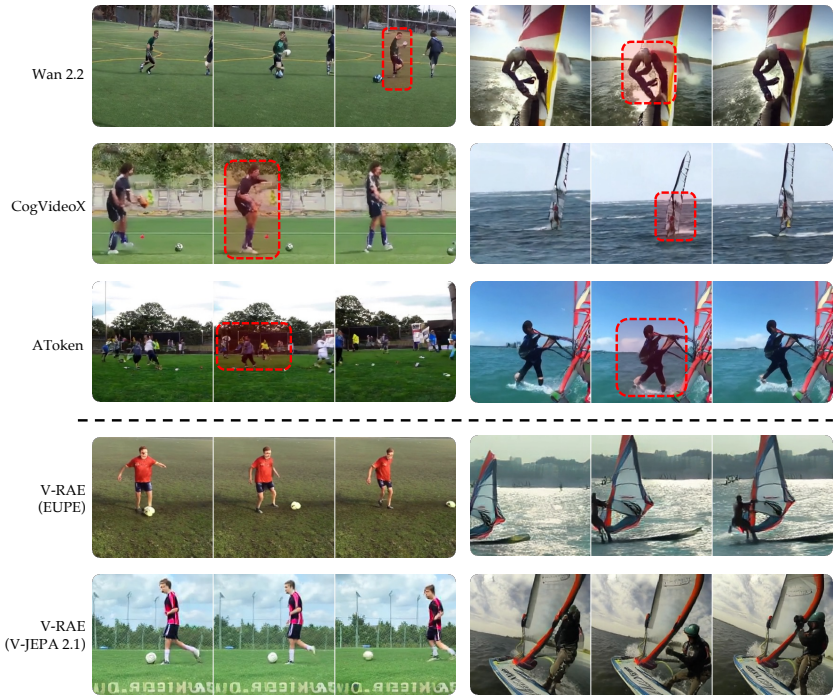


Figure 13: Qualitative class-conditional generation on Kinetics-600.



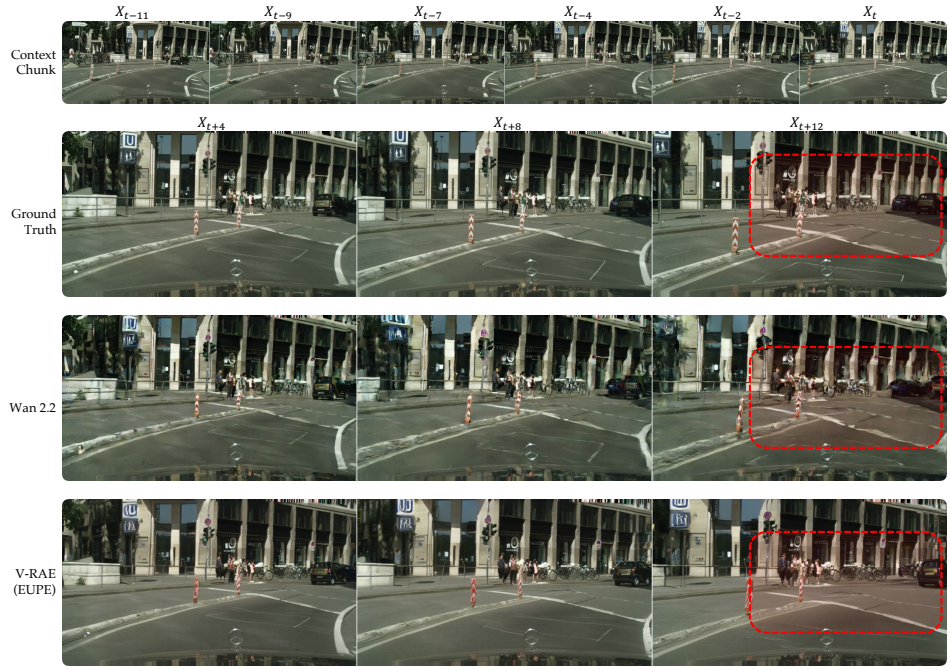
(a) UCF101

(b) Kinetics-600

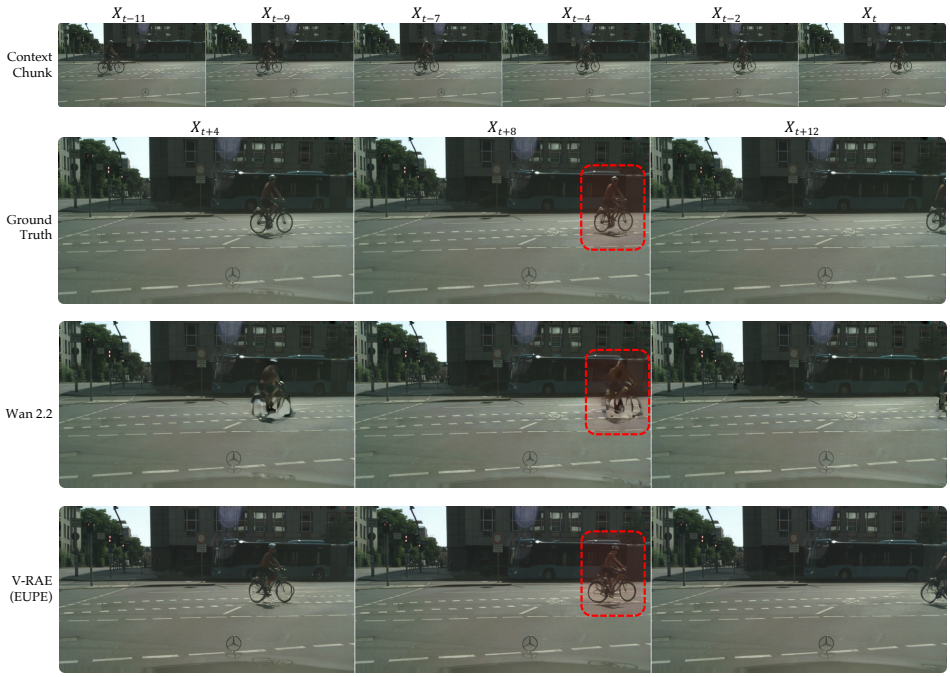
Figure 14: Class-conditional generation results for V-RAE with DINOv3-L and SigLIP2-L

F.2 Future Video Prediction

Figure 15 shows two additional Cityscapes scenes under the prediction setting in Appendix B.4.



(a) Pedestrian-dominated intersection.



(b) Crossing cyclist.

Figure 15: **Additional future video prediction examples on Cityscapes.** Rows show context, ground truth, Wan2.2 VAE predictions, and V-RAE predictions.

G LIMITATIONS

We finally discuss several limitations of the current work that motivate future research. First, our experiments primarily focus on relatively controlled video benchmarks and moderate-scale generative models. While these settings allow us to isolate the effect of the latent representation, it remains unclear whether the advantages of V-RAE extend to large-scale open-domain text-to-video generation with larger datasets, higher resolutions, longer videos, and substantially larger DiT backbones. Second, although we evaluate several representative image- and video-based encoders, we do not systematically disentangle how different pretraining objectives, architectures, and spatiotemporal inductive biases affect reconstruction and generation. Finally, while our experiments reveal a strong association between semantic preservation and downstream generation quality, the precise relationship between semantic structure in the latent space and its generative utility remains to be better understood.