Latent-Compressed Variational Autoencoder for Video Diffusion Models

Supplementary Material

Contents

A. Training Details

The training hyperparameters are listed in Tab. 5. A reference implementation and pretrained checkpoints can be found at: https://github.com/1mather/LC-VAE-code.

Table 5. Training hyperparameters used in all experiments.

Hyperparameter	Value
Training steps	200k
Learning rate	1×10^{-5}
Total batch size	32
Perceptual (LPIPS) weight	1.0
Adversarial loss weight (λ_{adv})	dynamic
KL weight (λ_{KL})	1×10^{-6}
Resolution	256×256
Number of frames	32
EMA decay	0.999

Following [27], we adopt a dynamic adversarial-loss weighting scheme that balances the gradient magnitudes of the adversarial and reconstruction objectives:

$$\lambda_{\text{adv}} = \frac{1}{2} \frac{\|\nabla_{G_L} \mathcal{L}_{\text{recon}}\|}{\|\nabla_{G_L} \mathcal{L}_{\text{adv}}\| + \delta}, \tag{9}$$

where $\nabla_{G_L}(\cdot)$ denotes the gradient with respect to the decoder's final layer and $\delta = 10^{-6}$ ensures numerical stability.

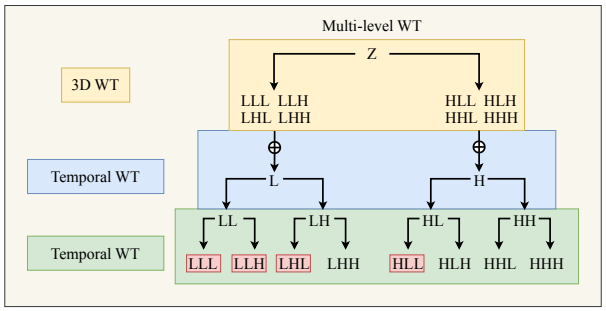


Figure 9. **Illustration of the proposed Multi-WT.** A 3D WT is first applied to the latent $\mathbf{z}$ to obtain eight subbands; two successive Temporal WT stages then further decompose them. In the Multi-WT representation the three letters (e.g., LHL) index temporal decomposition stages rather than spatial axes as in Eq. (5). We retain only the low-frequency–dominant subbands (LLL , LLH , LHL , HLL) and zero out the rest; $\oplus$ denotes channel-wise concatenation.

B. Multi-Level Wavelet Transform of Latent

The Multi-WT proceeds in three stages. **(1) 3D WT:** starting from $\mathbf{z}$, a 3D WT along the temporal, height, and width dimensions yields eight subbands LLL, LLH, LHL, LHH, HLL, HLH, HHL, HHH, where each character (L/H) denotes a low-/high-frequency component along one axis. **(2) Second-level Temporal WT:** the eight subbands are grouped by their temporal component and a Temporal WT is applied to each group, producing four subbands LL, LH, HL, HH. **(3) Third-level Temporal WT:** a further Temporal WT on each of the four subbands yields eight subbands again denoted LLL–HHH.

This Multi-WT notation differs from the basic 3D WT in Eq. (2): there, the three letters index the temporal, height, and width axes; here they index the first-, second-, and third-stage decompositions. We empirically select LLL, LLH, LHL, HLL as the low-frequency–dominant subbands to retain, guided by the energy and autocorrelation analyses in Sec. C.

C. Latent Frequency Analysis

We provide a comprehensive characterisation of the latent frequency spectrum through visualisation (Sec. C.1), energy analysis (Sec. C.2), and temporal autocorrelation (Sec. C.3).

C.1. Frequency Subband Visualisation

Given a latent tensor of shape $(T \times C \times H \times W)$ produced by WF-VAE [27], we apply a 3D WT along the temporal and spatial dimensions to obtain eight subbands. As shown

in Figs. 13 and 14, the low-frequency subbands contain rich, semantically diverse content, whereas high-frequency subbands are nearly identical across channels—indicating that high-frequency components are amenable to further compression.

C.2. Subband Energy Distribution

We compute the average energy (mean squared magnitude) of each subband over WebVid-10M. Results in Figs. 10 and 11 show that the low-frequency subbands (LLL, LLH, LHL, LHH) collectively account for roughly 85% of the total latent energy, while the LLL subband additionally exhibits the greatest channel-wise variation. The remaining subbands show nearly uniform per-channel energy, confirming that they contribute little semantic information.

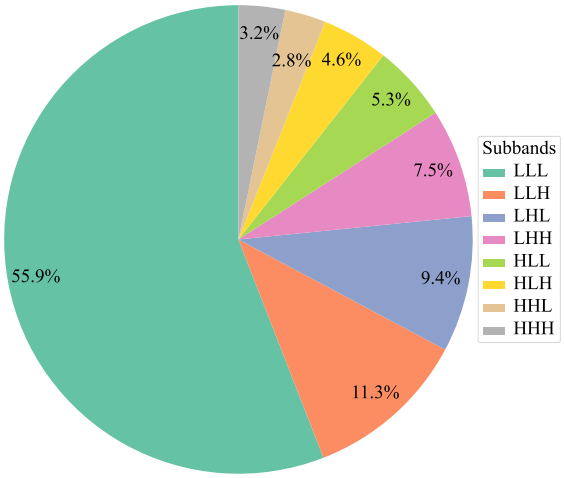


Figure 10. **Overall energy distribution across wavelet subbands (WebVid-10M).** Low-frequency subbands dominate, accounting for ∼85% of total energy.

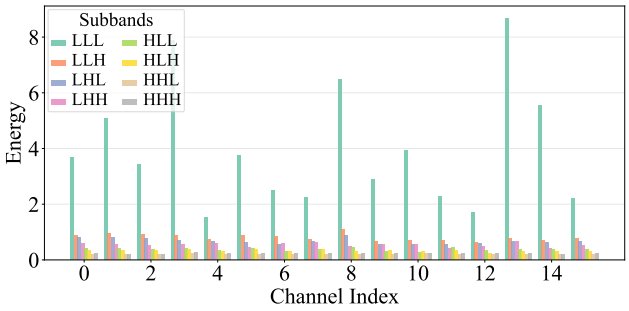


Figure 11. **Per-channel energy distribution across wavelet subbands.** The LLL subband shows substantially larger channel-wise variation than all high-frequency subbands, which exhibit nearly uniform energy across channels.

C.3. Lag-1 Temporal Autocorrelation

To quantify temporal smoothness, we compute the lag-1 temporal autocorrelation [40] for each subband. Given a temporal sequence $\{x_t\}_{t=1}^{T}$:

$$\rho(1) = \frac{\mathbb{E}[(x_t - \mu)(x_{t+1} - \mu)]}{\sigma^2}, \tag{10}$$

where $\mu = \mathbb{E}[x_t]$ and $\sigma^2 = \mathbb{E}[(x_t - \mu)^2]$. For multi-dimensional tensors of shape $T \times C \times H \times W$:

$$\rho_c(1) = \frac{\mathbb{E}_{t,h,w}[(x_{t,c,h,w} - \mu_c)(x_{t+1,c,h,w} - \mu_c)]}{\sigma_c^2}, \tag{11}$$

where c indexes channels. Results (visualised in Fig. 3) show that low-frequency subbands (LLL, LLH) consistently exhibit higher autocorrelation than high-frequency subbands (HHL, HHH), confirming that low-frequency components encode temporally stable structures whereas high-frequency components capture rapidly varying, noise-like details.

This observation also explains the improved zero-shot generalisation of LC-VAE (see Tab. 2): low-frequency content such as global layout and coarse motion is consistent across datasets, whereas high-frequency details are domain-specific and difficult to transfer. By discarding these high-frequency variations, LC-VAE concentrates the latent space on the most stable and semantically relevant components.

D. Additional Ablation Studies

D.1. Adaptive vs. Fixed Frequency Selection

A natural alternative to our fixed zero-out design is to learn which channels to retain adaptively. We implement an *adaptive Top-50%* baseline that, at each training step, preserves the 50% of channels with the highest energy. As shown in Fig. 12, after 20k steps this scheme converges to the same subbands (LLL, LLH, LHL, HLL) as our fixed design, with a 98% channel overlap—empirically validating that our fixed mask closely approximates the data-driven optimum.

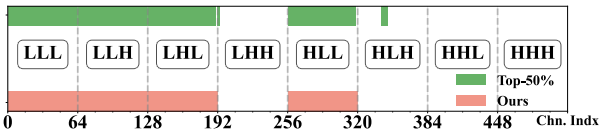


Figure 12. **Selected channels under adaptive Top-50% selection.** After 20k training steps the scheme converges to 98% channel overlap with our fixed design.

Despite this convergence, the adaptive baseline performs worse in reconstruction (Tab. 6): learning the energy distribution itself introduces additional training overhead that reduces effective capacity under a fixed compute budget. Our fixed design avoids this cost, consistent with classical approaches such as JPEG that zero out high-frequency coefficients directly via quantization.

Table 6. **Reconstruction: adaptive Top-50% vs. fixed design.**

Method	Chn.	WebVid-10M				OpenVid-1M			
		PSNR↑	SSIM↑	LPIPS↓	rFVD↓	PSNR↑	SSIM↑	LPIPS↓	rFVD↓
Top-50%	8	30.37	0.8456	0.0536	379.46	32.13	0.8329	0.0495	257.81
LC-VAE (Ours)	8	**30.51**	**0.8637**	**0.0485**	**352.73**	**32.24**	**0.8746**	**0.0414**	**234.10**
Top-50%	16	31.49	0.8597	0.0425	337.85	33.18	0.8354	0.0384	181.61
LC-VAE (Ours)	16	**31.81**	**0.8839**	**0.0413**	**319.24**	**33.26**	**0.8973**	**0.0329**	**168.46**

D.2. Robustness on Challenging Video Content

We construct *texture-heavy* and *fast-motion* subsets from OpenVid-1M and UCF-101 to stress-test the method. The texture-heavy subset contains clips with rich fine-grained details (e.g., foliage, fabrics); the fast-motion subset contains clips with large inter-frame displacements. As shown in Tab. 7, LC-VAE consistently outperforms WF-VAE across all metrics on both subsets, demonstrating robustness to challenging spatial textures and rapid temporal dynamics.

Table 7. **Reconstruction on texture-heavy and fast-motion subsets (16-channel VAEs).**

Method	Chn.	Texture-Heavy				Fast-Motion			
		PSNR↑	SSIM↑	LPIPS↓	rFVD↓	PSNR↑	SSIM↑	LPIPS↓	rFVD↓
WF-VAE	16	36.14	0.9587	0.0153	124.95	33.64	0.9431	0.0206	195.78
LC-VAE (Ours)	16	**36.78**	**0.9622**	**0.0142**	**112.78**	**34.07**	**0.9435**	**0.0194**	**177.62**

D.3. Effect of Data Quality on rFVD

In the main paper, rFVD for LC-VAE occasionally regresses on lower-quality datasets (WebVid-10M, UCF-101) despite consistent PSNR gains. We hypothesize that LC-VAE acts as an implicit artifact filter: suppressing high-frequency latent components discourages the model from fitting compression noise present in degraded data.

To verify this, we construct *OpenVid-1M (Compressed)* by re-encoding the original clips with H.264 to simulate real-world transmission quality loss. We then evaluate reconstruction against both the compressed inputs and the original clean reference frames. As shown in Tab. 8, LC-VAE yields *worse* rFVD relative to the compressed inputs (it does not overfit to artifacts) but *better* rFVD relative to the clean originals, confirming that our method implicitly filters artifacts while preserving perceptual fidelity on clean data.

Table 8. **rFVD evaluated against compressed vs. clean references (16-channel, OpenVid-1M).**

Method	Chn.	OpenVid-1M (Compressed)				OpenVid-1M (Original)			
		PSNR↑	SSIM↑	LPIPS↓	rFVD↓	PSNR↑	SSIM↑	LPIPS↓	rFVD↓
WF-VAE	16	35.97	0.9297	0.0176	**47.91**	35.63	0.9308	0.0181	53.33
LC-VAE (Ours)	16	**36.95**	**0.9375**	**0.0163**	58.79	**36.92**	**0.9384**	**0.0177**	**40.34**

E. Extended Experiments

E.1. Generation with Longer Training

The generation experiments in the main paper use 100k diffusion training steps. Here we extend to 300k steps with Latte-L [31] trained on OpenVid-1M and WebVid-10M (4× NVIDIA H200 GPUs, ≈4 days). As shown in Tab. 9, LC-VAE consistently outperforms WF-VAE across all channel counts, and the advantage grows with more channels, confirming that larger-channel LC-VAEs benefit from additional diffusion training.

Table 9. **Video generation FVD_{16} (↓) with Latte-L, 300k training steps.**

VAE Method	Chn.	WebVid-10M	OpenVid-1M
WF-VAE	4	492.87	279.65
LC-VAE (Ours)	4	**473.46**	**229.03**
WF-VAE	8	465.91	268.28
LC-VAE (Ours)	8	**431.44**	**226.45**
WF-VAE	16	457.22	236.42
LC-VAE (Ours)	16	**414.38**	**217.51**

E.2. Scalability to Larger Architectures

WanVAE-2.1. We apply our latent compression to WanVAE-2.1 [44] to verify generality beyond WF-VAE. As shown in Tab. 10, LC-VAE consistently improves PSNR and rFVD over the vanilla baseline on both datasets, demonstrating that fixed high-frequency zero-out is architecture-agnostic.

Table 10. **Reconstruction with LC-VAE applied to WanVAE-2.1 (16 channels).**

Method	Chn.	WebVid-10M				OpenVid-1M			
		PSNR↑	SSIM↑	LPIPS↓	rFVD↓	PSNR↑	SSIM↑	LPIPS↓	rFVD↓
WanVAE-2.1 (Vanilla)	16	29.26	0.8376	**0.0621**	381.19	30.99	0.8508	0.0551	274.05
WanVAE-2.1 (Ours)	16	**29.91**	**0.8437**	0.0641	**342.68**	**31.28**	**0.8591**	**0.0532**	**249.92**

Wan-2.1 (2.1B) Diffusion Model. We further train the Wan-2.1 diffusion model scaled to 2.1B parameters for 160k steps on WebVid-10M (8× H200 GPUs, ≈3 days). Tab. 11 shows that LC-VAE substantially outperforms WF-VAE, confirming that the benefits of latent compression persist at larger model scales.

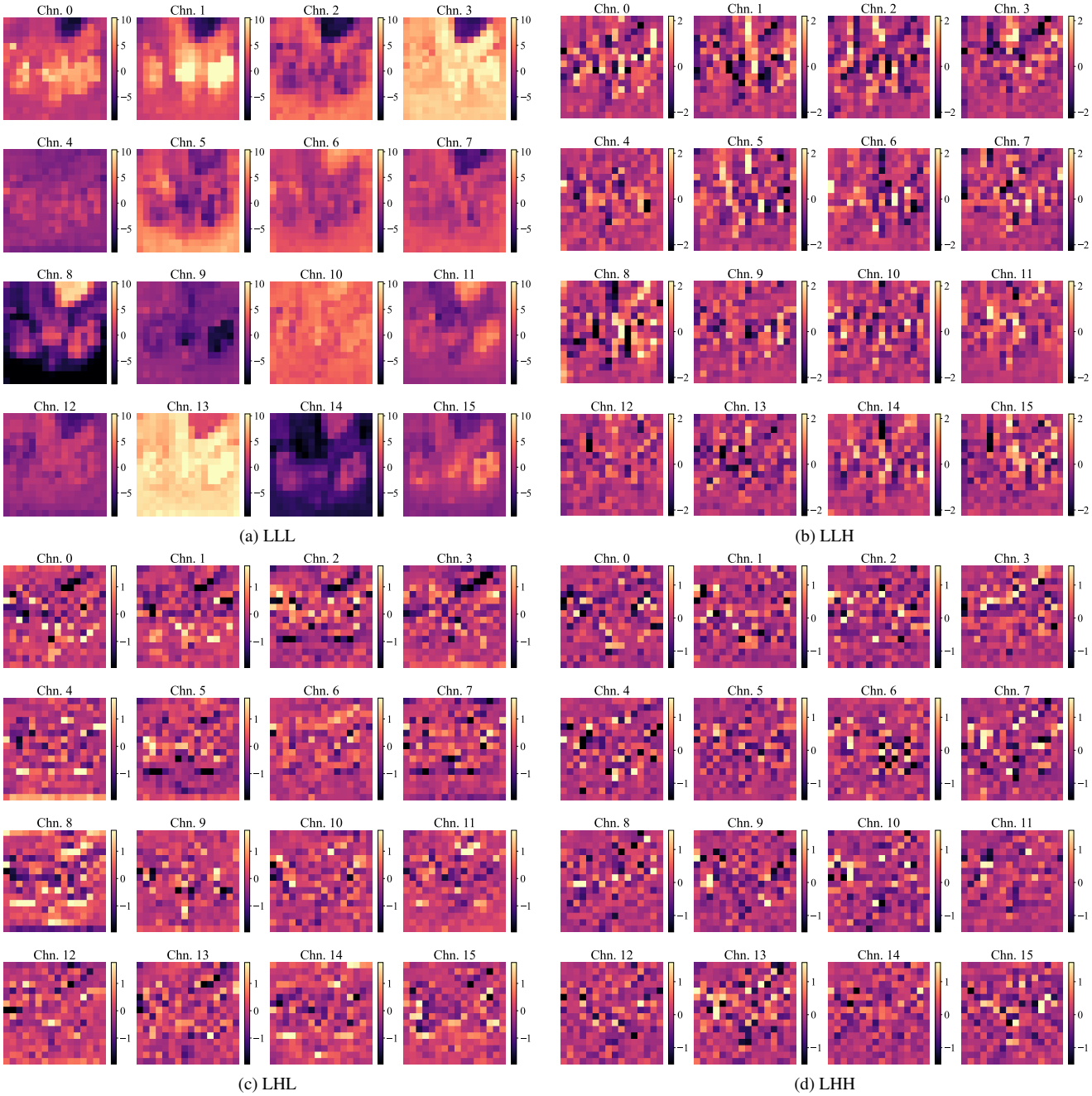
Table 11. **Video generation FVD_{16} (↓) with Wan-2.1 (2.1B) on WebVid-10M.**

Method	Chn.	WebVid-10M
WF-VAE	16	487.15
LC-VAE (Ours)	16	**434.41**

F. Additional Qualitative Results

Non-curated reconstruction. Fig. 15 shows non-curated reconstruction results produced by LC-VAE on OpenVid-1M.

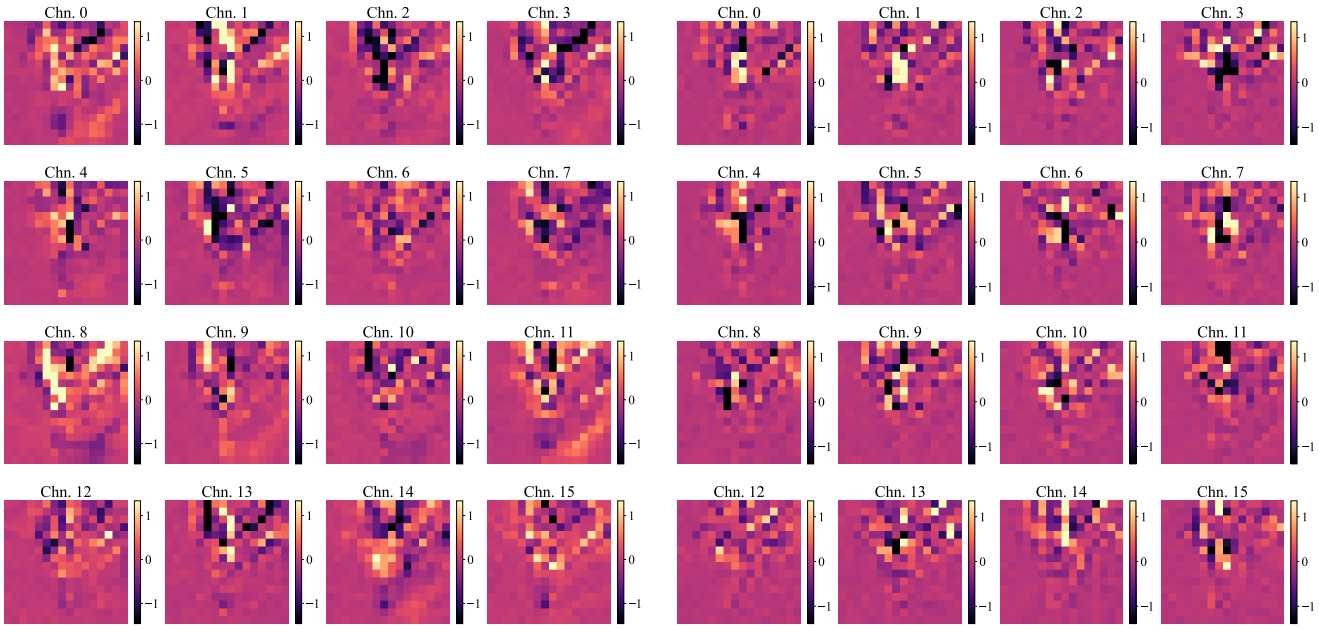
Non-curated video generation. Fig. 16 shows non-curated generation results obtained by integrating LC-VAE into a latent diffusion video generator, evaluated on the SkyTime-lapse dataset.



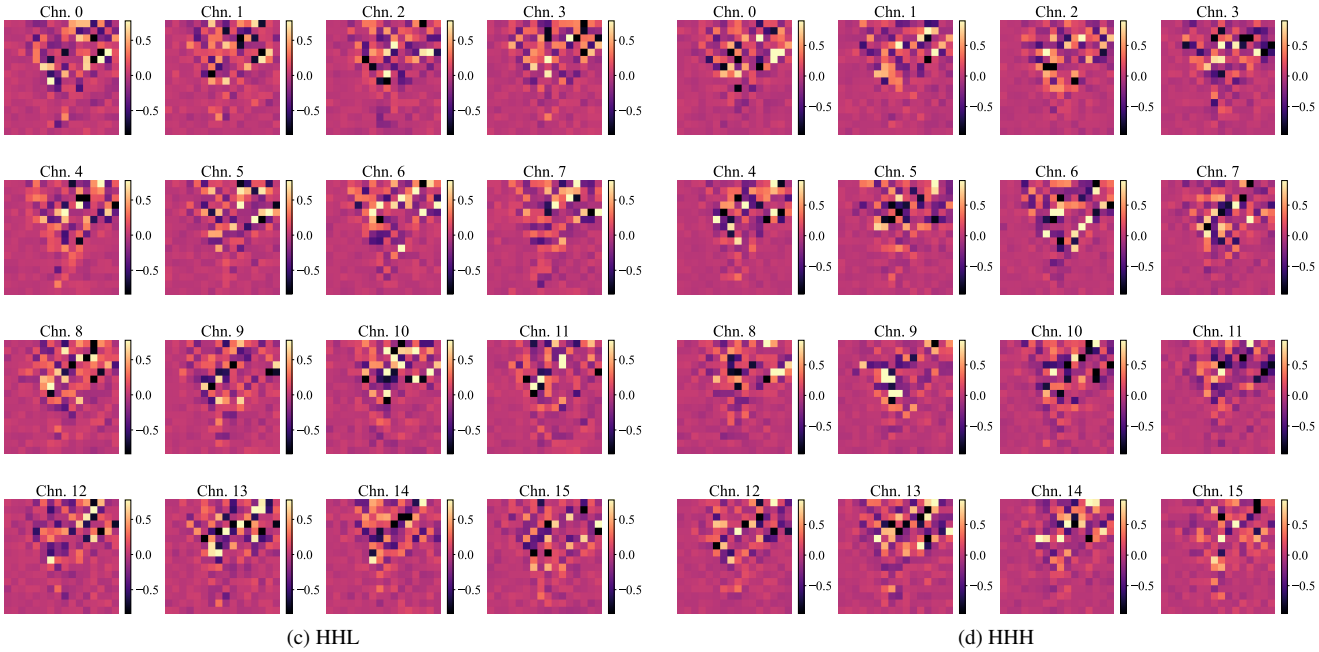
(a) LLL (b) LLH

(c) LHL (d) LHH

Figure 13. **Visualization of low-frequency wavelet subbands.** Low-frequency components exhibit smooth spatial variations and clear structural patterns, encoding the majority of semantic content. Diverse per-channel activation patterns suggest that each channel captures distinct semantic factors.

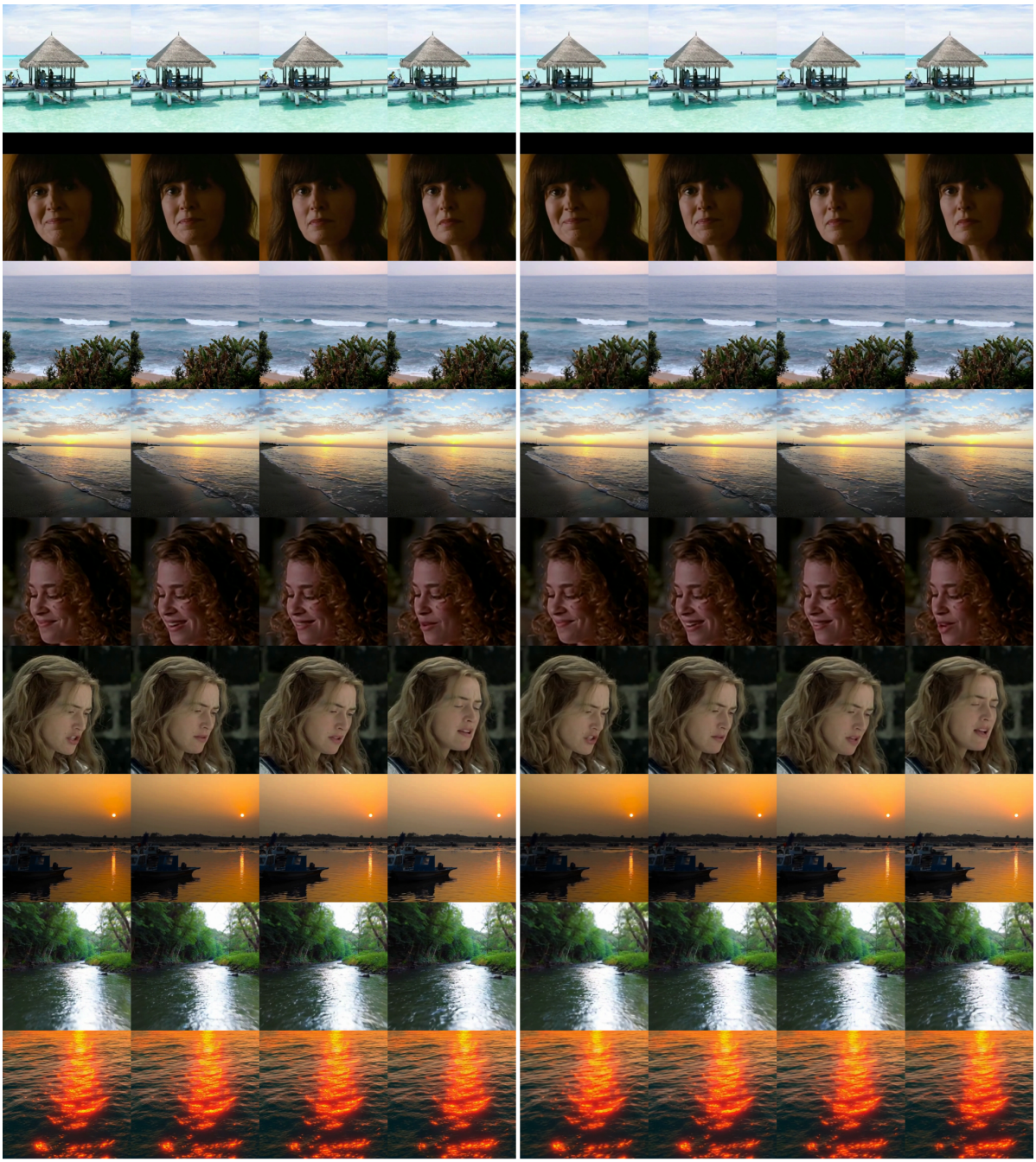


(a) HLL (b) HLH



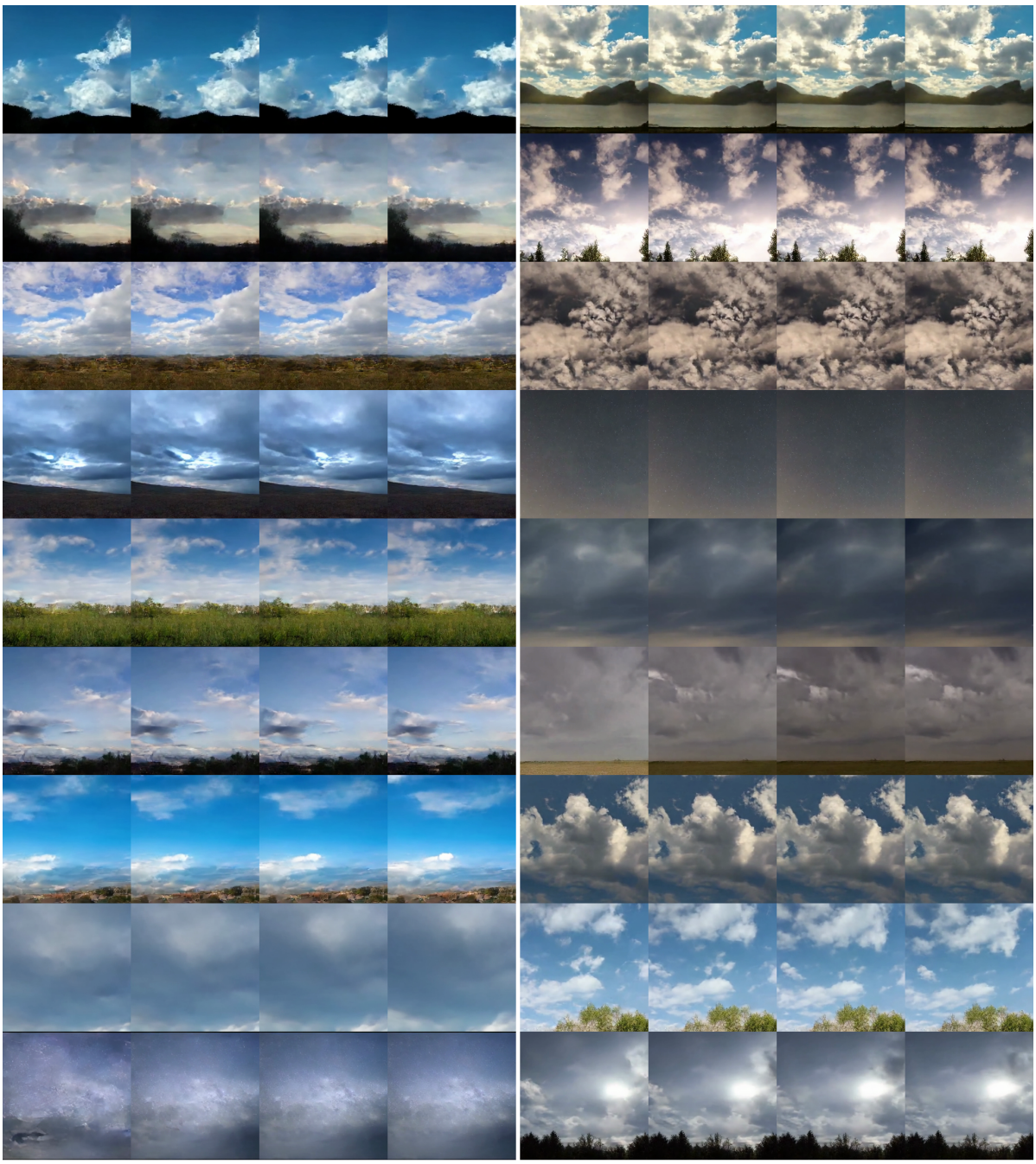
(c) HHL (d) HHH

Figure 14. **Visualization of high-frequency wavelet subbands.** High-frequency components contain rapid local fluctuations with little channel-wise variation, resembling noise-like textures and contributing minimal semantic information.



LC-VAE 16 Chn. WF-VAE 16 Chn.

Figure 15. **Non-curated reconstruction on OpenVid-1M.** LC-VAE (left) vs. WF-VAE (right).



LC-VAE 16 Chn. WF-VAE 16 Chn.

Figure 16. **Non-curated video generation on SkyTimelapse.** Latte [31] under guidance-free sampling trained with LC-VAE (left) vs. WF-VAE (right).