Supplementary Materials for "Improved Video VAE for Latent Video Diffusion Model"

Pingyu Wu[1], Kai Zhu[1,*], Yu Liu[2], Liming Zhao[2], Wei Zhai[1,*], Yang Cao[1], Zheng-Jun Zha[1]
[1] MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China [2] Independent Researcher
{wpy364755620@mail., zkzy@mail., wzhai056@, forrest@, zhazj@}ustc.edu.cn

1. Introduction Details

1.1. Calculation of the information preservation degree

To measure the information completeness of the video after compression using different VAE models, we borrow from the way of calculating the information loss rate of the dimensionality data in Principal Component Analysis (PCA), which is formulated as:

$$\frac{\sum_i^m \|x_i - x_i'\|^2}{\sum_i^m \|x_i\|^2}, \tag{1}$$

where x_i is the original video, x_i' is the reconstructed video and m is the number of samples. However, considering that MAE loss is used instead of MSE loss in the training of VAE model, we adopt the following formula to measure the information preservation degree:

$$1 - \frac{\sum_i^m |x_i - x_i'|}{\sum_i^m |x_i|}. \tag{2}$$

1.2. Calculation process of SSIM results of different frames within a frame group

To measure the performance of frames with different positions in each frame group, we calculate the SSIM results for different frames in the 17 reconstructed frames on the Kinetics-600 validation set. After excluding the first image frame, the remaining 16 frames contain 4 frame groups. We calculate the average performance of frames at the same position in different frame groups, *e.g.*, the first frame of each frame group, whose positions in the 16 frames are 1, 5, 9, and 13, respectively. In this way, we obtain the average performance of each position in a group of frames for a not-so-short time series, to ensure the generalization of the experimental results.

*Kai Zhu and Wei Zhai are the corresponding authors.

2. Method Details

2.1. Group Causal Convolution

Grouping of the input video frames is done before input to the network, each t_c frames (t_c is the temporal compression rate) are labeled as a frame group and there is no overlap between the frame groups. During the forward process of the whole network, the t_c frames in a frame group will be compressed to one frame and then restored to t_c frames, so the number of frames in a frame group varies at different locations in the network. For each temporal downsampling/upsampling operation of the feature map, the number of frames contained in the frame group for that layer of the network is decreased/increased by the same factor. During the encoding process, the number of frames in the frame group is reduced from t_c to 1 and then restored to t_c after the decoding process.

2.2. Weight Initialization

When expanding an image VAE into a video VAE, in addition to expanding the 2D convolutions into 3D convolutions based on the center initialization [1], it is necessary to set the weights of the newly added temporal downsampling layers and temporal upsampling layers. To make the video VAE can initially recover a complete frame from the t_c frames (t_c is the temporal compression rate, *i.e.*, 4), we let the temporal downsampling layer initially behave as an identity mapping of the features of a particular frame, and then discard the features of the other frames. Specifically, noting the initial temporal downsampling layer weights as $W_{down} \in \mathbb{R}^{c \times c \times k_t \times k_h \times k_w}$, where the number of input and output channels are both c, k_t, k_h, k_w are the sizes of the convolution kernel in the dimension of time, height, width, respectively, and the convolution stride is 2. All values of matrix W_{down} are initialized to 0. Then we construct a 2D identity matrix $I \in \mathbb{R}^{c \times c}$, with diagonal values of 1 and other values of 0. The values of the matrix W_{down} are as-

signed as follows:

$$W_{down}[:,:,t,(k_h-1)//2,(k_w-1)//2] = I, \quad (3)$$

t represents which frame's features need to be inherited, *e.g.*, to initial recover the last frame's features, making t equal to -1 for causal convolution. $[(k_h-1)//2,(k_w-1)//2]$ denotes the center of the convolution kernel.

For the newly added temporal upsampling layer, we let it behave as a repeat of the existing frame features. Specifically, noting the temporal upsampling layer weights as $W_{up} \in \mathbb{R}^{2c\times c\times k_t\times k_h\times k_w}$, where the number of output channels is twice the number of input channels c. All values of matrix W_{up} are initialized to 0. Then the values of the matrix W_{up} are assigned as follows:

$$\begin{aligned} W_{up}[:c,:,t,(k_h-1)//2,(k_w-1)//2] = I, \\ W_{up}[c:,:,t,(k_h-1)//2,(k_w-1)//2] = I. \end{aligned} \quad (4)$$

For the generated feature maps, we use the stack operation to convert twice the number of channels to twice the number of frames, thus achieving temporal upsampling.

3. Experiments

3.1. Experimental Setup

Training Dataset. We use in-house data to train the video VAE. It is worth noting that the training data does not contain the test datasets used in the paper, such as Kinetics-400/600, ActivityNet, and OpenVid-0.4M.

Training losses. During the training process, we use MAE, LPIPS, KL, and GAN losses as training losses, where GAN loss is only added in the last stage of video VAE training. Considering that all the losses are computed on the image aspect using the form of sum, the scales of the losses are not the same for different resolutions. So we normalize the loss for different resolutions to the same magnitude as the loss for 256×256 resolution. The total loss on $h \times w$ resolution video is formulated in the following form:

$$L = \frac{256\times 256}{h\times w} \times (L_{MAE} + L_{LPIPS} + \alpha L_{KL} + \beta L_{GAN}), \quad (5)$$

where $\alpha = 3\times 10^{-6}$ and $\beta = 0.8$ if GAN loss is used, otherwise $\beta = 0$.

Latte setting. On the video generation task, we use exactly the same training settings for all methods, i.e., a total video data batch size of 32 (on 4 A800 GPUs), a learning rate of 1e-4, a frame number of 17, a sampling step number of 250, a frame interval of 3. In particular, image data is used for training on the SkyTimelapse dataset, and we set the total image data batch size to 20 for all methods.

3.2. Qualitative evaluation of reconstruction results

In Fig. 3, Fig. 4, Fig. 5, Fig. 6, and Fig. 7 we compare the reconstruction results of different methods in various scenes, including long sequence reconstruction, fast-motion, face reconstruction, textual reconstruction, and sudden change of neighboring frames, respectively. Specifically, in Fig. 3, we show the reconstruction results for a long sequence of frames (9 frames), where poorly reconstructed regions are circled. For other methods, there is the general problem that the performance of the first frame of each frame group is much lower than the average performance, leading to obvious reconstruction flicker within and between frame groups. In contrast, the reconstruction results of our method are more stable and consistent.

In Fig. 4, we show the reconstruction results within a frame group at a fast camera motion (a frame interval of 3) and list some common reconstruction problems. For CogX-VAE, the reconstruction performance of the first and second frames within a frame group is poor and there appears to be a performance imbalance. For OD-VAE, most of the frames show significant color deviations in the details, indicating that the model is difficult to achieve effective temporal compression when the motion is fast. OS-VAE and CV-VAE suffer from motion blur, making it difficult to reconstruct details. In contrast, our method has high quality and consistency of reconstruction results at both $Z = 16$ and $Z = 4$, which validates the stability and high performance of IV-VAE in fast-motion scenarios.

In Fig. 5, we compare the quality of face reconstruction by different methods. On the latent channel number of 16, our method can almost perfectly restore the face of the person at any frame, while CogX-VAE reconstructs roughly and unclearly in some details, especially on the first frame, where these areas are circled in red. On the latent channel number of 4, all methods struggle to reconstruct a very small face well. The reconstruction results of OD-VAE and CV-VAE have serious distortion problems. OS-VAE, although relatively consistent in the shape of the face, has large differences in color and structure from the original image, such as the position of the eyes, the shape of the lips, and the letter on the hood. In general, our method is closest to the original image in terms of color, structure, and texture.

In Fig. 6, we compare the quality of text reconstruction by different methods. On $Z = 16$, we can reconstruct almost the same clarity as the original image, and most of the text is recognizable. Compared to CogX-VAE, our reconstructed text is significantly more recognizable and almost without perturbations and distortions. The performance of IV-VAE is also more consistent for all four frames, while the performance of CogX-VAE for the third frame is significantly higher than the others. On $Z = 4$, reconstructing fine text is difficult. In this case, our method can reconstruct the headlines with large font well on all four frames, while other methods struggle to do so accompanied by blurring and distortion. Even on small text, IV-VAE ($Z = 4$) can

achieve significant clarity improvement compared to other methods.

Fig. 7 illustrates a special case when there is a drastic change in part of the region of two neighboring frames within a frame group, i.e., a caption that does not exist in the first frame appears in the second frame. In this case, although other methods reconstruct the caption in the second frame, the reconstruction result of the first frame is affected in different degrees, with the caption in the second frame appearing in the first frame or causing a perturbation distortion. We attribute this anomaly to the fact that causal convolution makes the interaction of different frames within a frame group unbalanced. For example, the first frame of a frame group can only interact with later frames in the middle layer of the network. This incomplete and insufficient interaction results in the process of compression and decompression of the frame group not being able to effectively separate the information of different frames. To solve this problem, the proposed group causal convolution enables different frames of the same frame group to interact with each other bidirectionally at any layer of the network, making our IV-VAE perform better in this case in Fig. 7.

3.3. More video reconstruction results

We randomly select 20K videos from kinetics-600 for testing, with a setting of 512×512 17-frame, and the results are shown in the Table 1. The results on larger datasets validate the effectiveness and generalization of our method.

Method	CV-VAE	OS-VAE	OD-VAE	IV-VAE(Z=4)	CogX-VAE	IV-VAE(Z=16)
PSNR↑	32.67	**34.80**	34.20	34.67	38.53	**39.46**
SSIM↑	0.9150	0.9289	0.9264	**0.9314**	0.9682	**0.9701**
LPIPS↓	0.08041	0.08016	0.05457	**0.05152**	0.02849	**0.02158**

Table 1. **Reconstruction comparison of 20K videos on Kinetics-600 dataset.** The experiments are performed using 512x512 resolution and 17 frames.

3.4. More video generation results

In addition to validating the generation performance using the Kinetics-400 [4] and SkyTimelapse [10] datasets in the main paper, we further explore class-conditional video generation in UCF-101 [8] and unconditional video generation on FaceForensics [5]. On the UCF-101 dataset, apart from FVD metric, we use Inception Score (IS) [7] as the metric. When computing IS, we remove the first frame of the generated results to utilize the remaining 16 frames for testing, since the C3D [6] model employed in IS is trained using 16 frames of video on UCF-101. As shown in Table 2, we also achieve state-of-the-art results on both datasets, especially on the FaceForensics dataset, where we obtain a decrease of 25.7 FVD compared to OD-VAE.

In Fig. 8 and Fig. 9, we further show the generated results of Latte using different video VAEs after training on the SkyTimelapse and FaceForensics datasets, respectively. Compared to other methods, the results generated by Latte using our VAE have higher realism, clarity, and more stable motion, validating the effectiveness of the proposed IV-VAE for the video generation task.

Method	UCF-101		FaceForensics
	FVD↓	IS↑	FVD↓
CV-VAE [12]	587.9	84.7	328.2
OS-VAE [3]	674.1	85.2	316.3
OD-VAE [2]	565.2	82.5	285.5
IV-VAE	**557.5**	**85.7**	**259.8**

Table 2. **Video generation results.**

3.5. Ablation Study

Comparison with baseline. In Table 3, we compare the computational effort and performance of the proposed IV-VAE with the baseline on $Z = 16$. Overall, IV-VAE increases the computational effort by 13% but reduces the parameters by 15%. With similar computational and parameter counts, the model performance is substantially improved as shown on the right side of Table 3. In addition to the Kinetics-600 dataset, the advantages of IV-VAE are more significant at larger motion speeds and higher resolutions, *e.g.*, on MotionHD at 1080P resolution, IV-VAE ($Z = 16$) achieves a 1.73 PSNR and 0.057 SSIM improvement compared to the baseline, which is a huge boost.

Method	Params↓	FLOPS↓	Kinetics-600		
			PSNR↑	SSIM↑	LPIPS↓
Baseline	127M	**30.5T**	38.07	0.9641	0.0289
IV-VAE	**108M**	34.6T	**39.02**	**0.9685**	**0.0228**
+Δ	(-15%)	(+13%)	(+0.95)	(+0.044)	(-0.0061)

Table 3. **Comparison with baseline.** FLOPS are calculated using a 17-frame 512×512 video. Reconstruction results for $Z = 16$ are reported.

Why choose a dual branch of 2D+3D. We ablate the branch structure of the KTC unit by replacing the Conv2D with GCConv3D. The experimental results are listed in Table 4, compared to the original 2D+3D structure, using the 3D+3D structure can achieve a slight enhancement. However, it also increases the parameter count by 23%, indicating insufficient parameter efficiency. This may be due to the fact that the spatial compression rate is higher than the temporal compression rate, and spatial compression is more challenging, thus replacing the 3D convolution with a 2D convolution causes only a slight performance penalty but reduces the parameter count significantly. So we finally choose 2D+3D as a more efficient structure.

	Params	PSNR↑	SSIM↑	LPIPS↓
(2D+3D)	**107M**	32.24	0.9158	0.04725
(3D+3D)	132M	**32.31**	**0.9163**	**0.04704**

Table 4. **Ablation for branch selection.**

Ablation of SSIM results of different frames within a frame group. In Table 5, we explore the performance change of each frame within the frame group after different architectures coupled with the proposed GCConv. The experiments utilize the $Z = 8$ ablation experiment weights from the main paper. As can be seen from the table, the addition of GCConv to both the baseline and the KTC architecture significantly improves the performance of the first two frames within the frame group, thus drastically reducing the performance imbalance between frames. Experiments demonstrate that the performance of different frames can be significantly balanced by allowing frames within a frame group to interact bidirectionally.

	Baseline + GCConv	KTC + GCConv
Frame 1	0.8923+0.0089	0.9027+0.0066
Frame 2	0.8986+0.0063	0.9103+0.0064
Frame 3	0.9191+0.0014	0.9203-0.0012
Frame 4	0.9056+0.0007	0.9111+0.0008
Max margin	0.0268-0.0075	0.0176-0.0078

Table 5. **Ablation of SSIM results of different frames within a frame group.** The experiments utilize the $Z = 8$ ablation experiment weights from the main paper. The results are tested using 17-frame 256×256 resolution videos. The red/gray values indicate the change in performance of the baseline or KTC architecture after using GCConv. Max margin: Maximum performance difference between frames within a frame group.

3.6. Analysis

Visualization of latent channels. In Fig. 10, we visualize the features of each channel in the latent space using $Z = 8$ IV-VAE model. Specifically, a frame group is compressed into a single frame after the encoder, and the feature map F_i for each channel $i(1 \leq i \leq 8)$ in the latent space is normalized by the following equation:

$$(F_i - \frac{max(F_i) + min(F_i)}{2}) \times \frac{2}{max(F_i) - min(F_i)}. \quad (6)$$

As shown in Fig. 10, we choose a simple motion scene to better analyze the role of the different latent channels, where the position of a fruit undergoes a longitudinal movement with the camera motion. The feature maps (1-4) generated by the 2D branch are usually clearer and focus on the information of the first frame of the frame group, while the feature maps generated by the 3D branch (5-8) are very blurry and contain information about the motion of multiple frames. This phenomenon is consistent with the description in the methodology, where the 2D branch focuses on the spatial information of the key frame (first frame) and the 3D branch focuses on the overall motion information.

Encoding and decoding time. We report the time of encoding and decoding by different methods on a 17-frame 720P video. The results are shown in Table 6. The proposed IV-VAE achieves the minimum reconstruction time, but due to not exploring more on the model variants, we do not perform as well as OD-VAE in terms of encoding time. On the $Z = 16$ model comparison, we have a very large advantage over CogX-VAE, both in terms of reconstruction quality and reconstruction time. In addition, we note that IV-VAE has only 45% parameters and 60% computation compared to OD-VAE, but the reconstruction time is somewhat long because of the inefficiency of the architecture. Therefore, in future work, we will further explore more efficient model variants and optimize the code to reduce the encoding and decoding time.

Method	Input: A 17-frame 1280×720 Video		
	Enc. time	Dec. time	All time
CV-VAE [12]	1.9s	3.5s	5.4s
OS-VAE [3]	1.2s	2.6s	3.8s
OD-VAE [2]	**0.9s**	2.9s	3.8s
CogX-VAE [11]	2.1s	4.2s	6.3s
IV-VAE (Cache)	1.3s	**2.0s**	**3.3s**

Table 6. **Encoding and decoding time on one A800 GPU.** Enc.:Encoding, Dec.:Decoding.

Applying GCConv to OD-VAE. We replace all causal convolutions with the proposed GCConvs based on pre-trained OD-VAE and fine-tune it for 200K steps, which not only achieves a 0.017 SSIM gain but also leads to a more balanced performance as shown in Fig. 1. Experiments verify that the proposed GCConv can effectively balance the performance between frames.

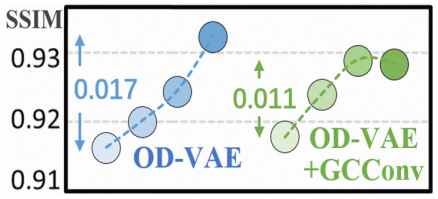

Figure 1. **Applying GCConv to OD-VAE.** Results are measured on Kinetics-600.

Unselected frames. We visualized the reconstruction results of the last unselected frame in the KTC architecture as shown in Fig. 2. We note that these unselected frames can still reconstruct the video albeit with color deviation and ghosting. This is because the convolution weights for outputting the unselected frames are inherited from the inflated pre-trained image VAE weights and are not updated afterwards due to the lack of direct constraints.

4. Limitation and Future Work

The overall architecture of the proposed method is still based on UNet following SD image VAE without explor-

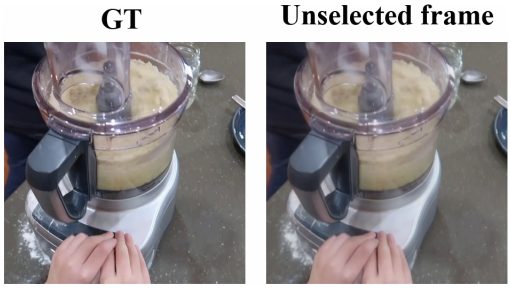

Figure 2. **Unselected frame reconstruction result.**

ing other architectures. Video VAE faces more unique difficulties compared to image VAE, *e.g.*, as the video resolution increases, the need for receptive field increases for video VAE. While the classical UNet lacks a global receptive field, in addition, the number of spatial downsampling layers of video VAE is usually aligned with the spatial compression ratio, which also limits the receptive field of video VAE. Therefore, it is worthwhile to consider introducing new architectures such as Dit or Mamba into video VAE in future work.

We believe that the proposed KTC architecture can play a more significant role in the video VAE of a larger latent channel number, and some works [9] on image VAE have attempted to train a VAE with a large latent channel number (*e.g.*, 32). In this case, the KTC architecture appears to be quite promising. In addition, the proposed GCConv also shows more potential in video VAEs with a larger temporal compression rate (*e.g.*, 8×), and we think that bidirectional interactions can effectively improve the performance of the model especially at a high temporal compression rate. Therefore, in future work, we would like to explore a larger latent channel number and a larger temporal compression rate for video VAE using the proposed IV-VAE.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[2] Liuhan Chen, Zongjian Li, Bin Lin, Bin Zhu, Qian Wang, Shenghai Yuan, Xing Zhou, Xinghua Cheng, and Li Yuan. Od-vae: An omni-dimensional video compressor for improving latent video diffusion model. *arXiv preprint arXiv:2409.01199*, 2024. 3, 4

[3] hpcaitech. Open-sora. https://github.com/hpcaitech/Open-Sora, 2024. 3, 4

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[5] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*, 2018. 3

[6] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 3

[7] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10): 2586–2606, 2020. 3

[8] 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. 3

[9] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024. 5

[10] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2364–2373, 2018. 3

[11] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 4

[12] Sijie Zhao, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Muyao Niu, Xiaoyu Li, Wenbo Hu, and Ying Shan. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024. 3, 4

Figure 3. **Reconstruction results for long sequences.** Frame 0 represents the first image frame. Areas of poor reconstruction are circled. **(Zoom-in for best view)**

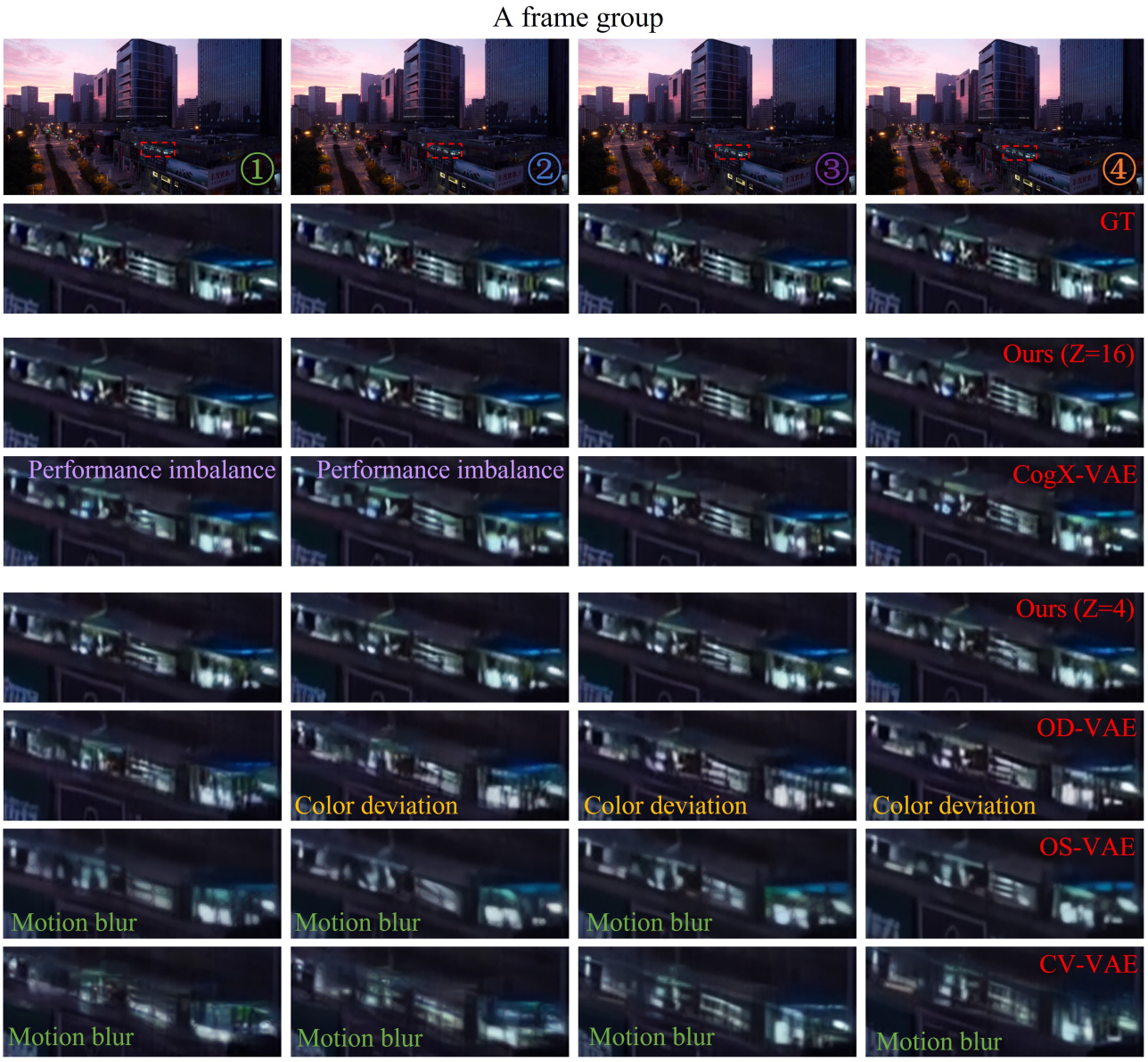

Figure 4. **Reconstruction results under a faster motion (frame interval of 3).** Reconstruction problems are listed for different frames.

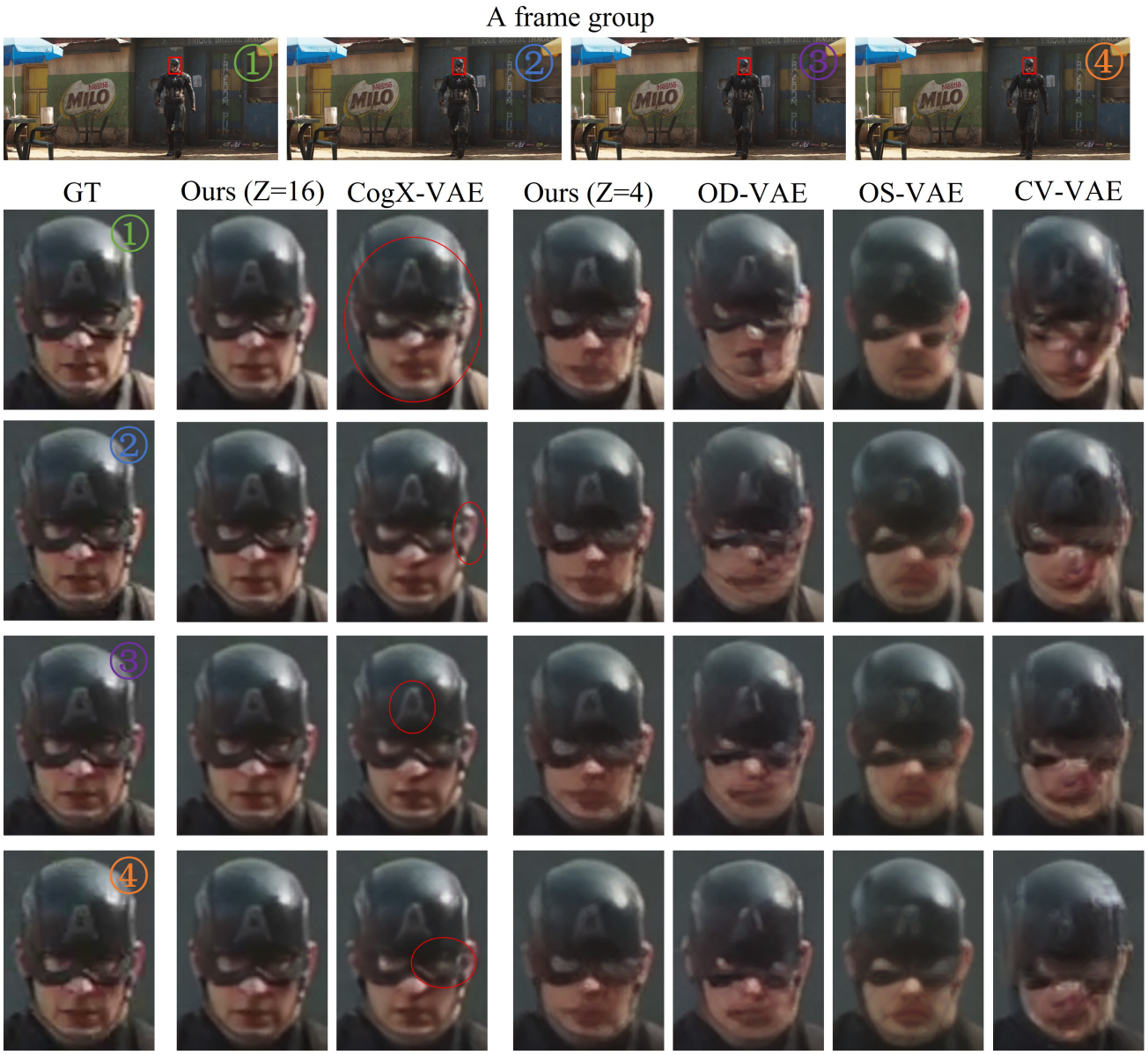

Figure 5. **Face reconstruction.** Rough and mismatched areas in CogX-VAE reconstruction results are circled.

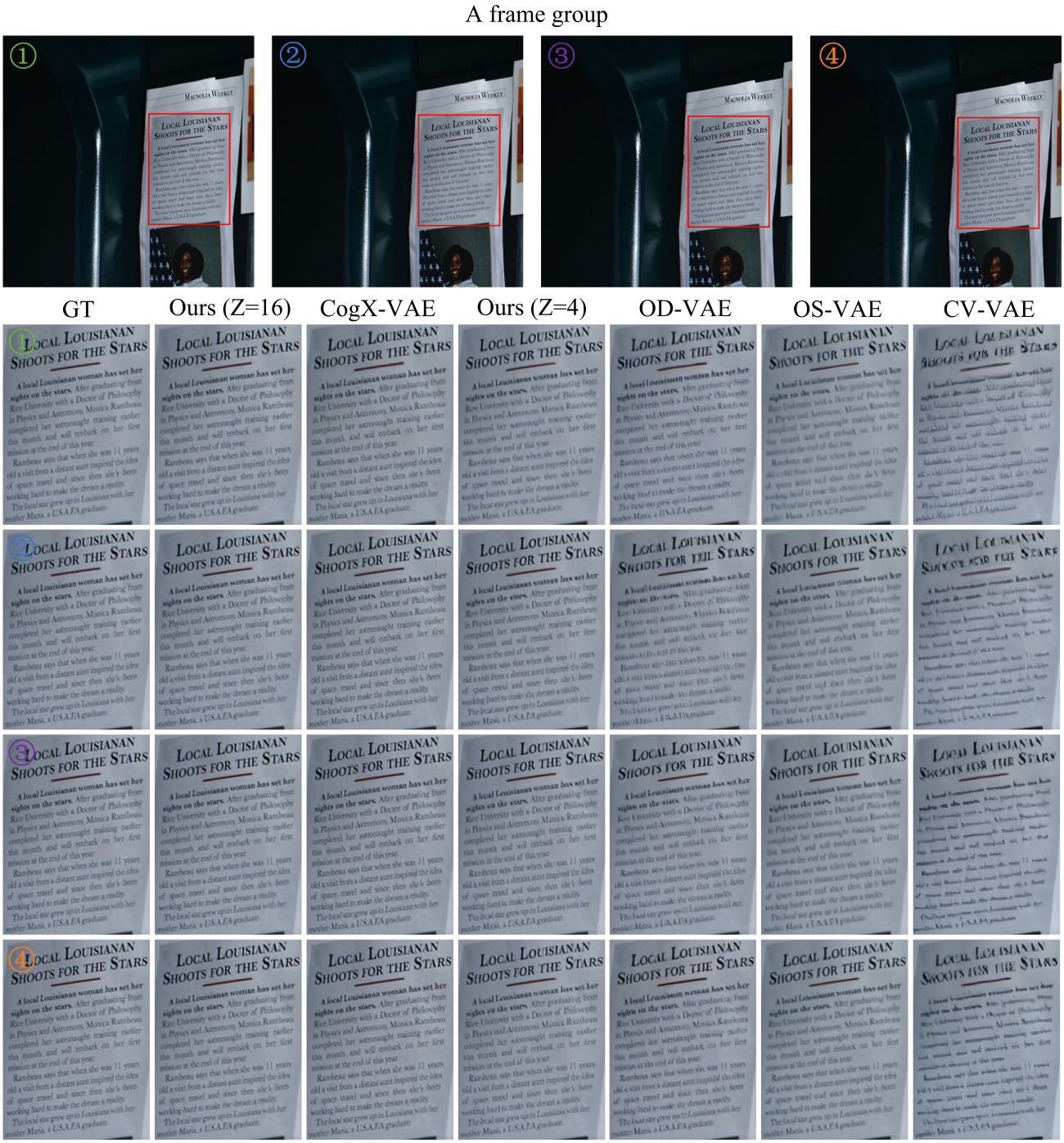

Figure 6. **Textual reconstruction. (Zoom-in for best view)**

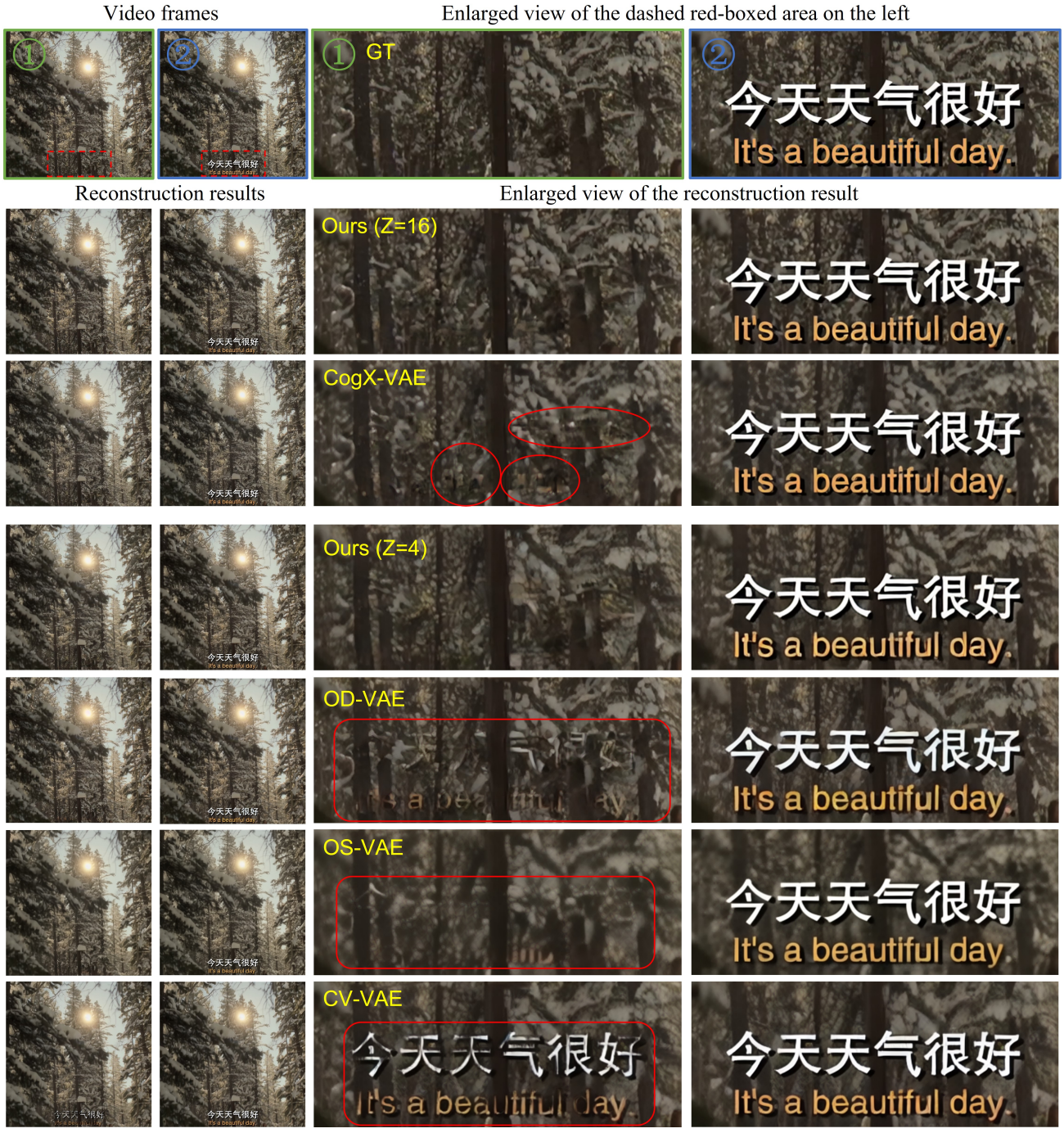

Figure 7. **Reconstruction results when text suddenly appears in the video frame.** Both frames belong to the same frame group. Areas with anomalies in the reconstruction results are circled.

Figure 8. **Latte's generation results using different video VAEs** after training on the SkyTimelapse dataset.

Figure 9. **Latte's generation results using different video VAEs** after training on the FaceForensics dataset.

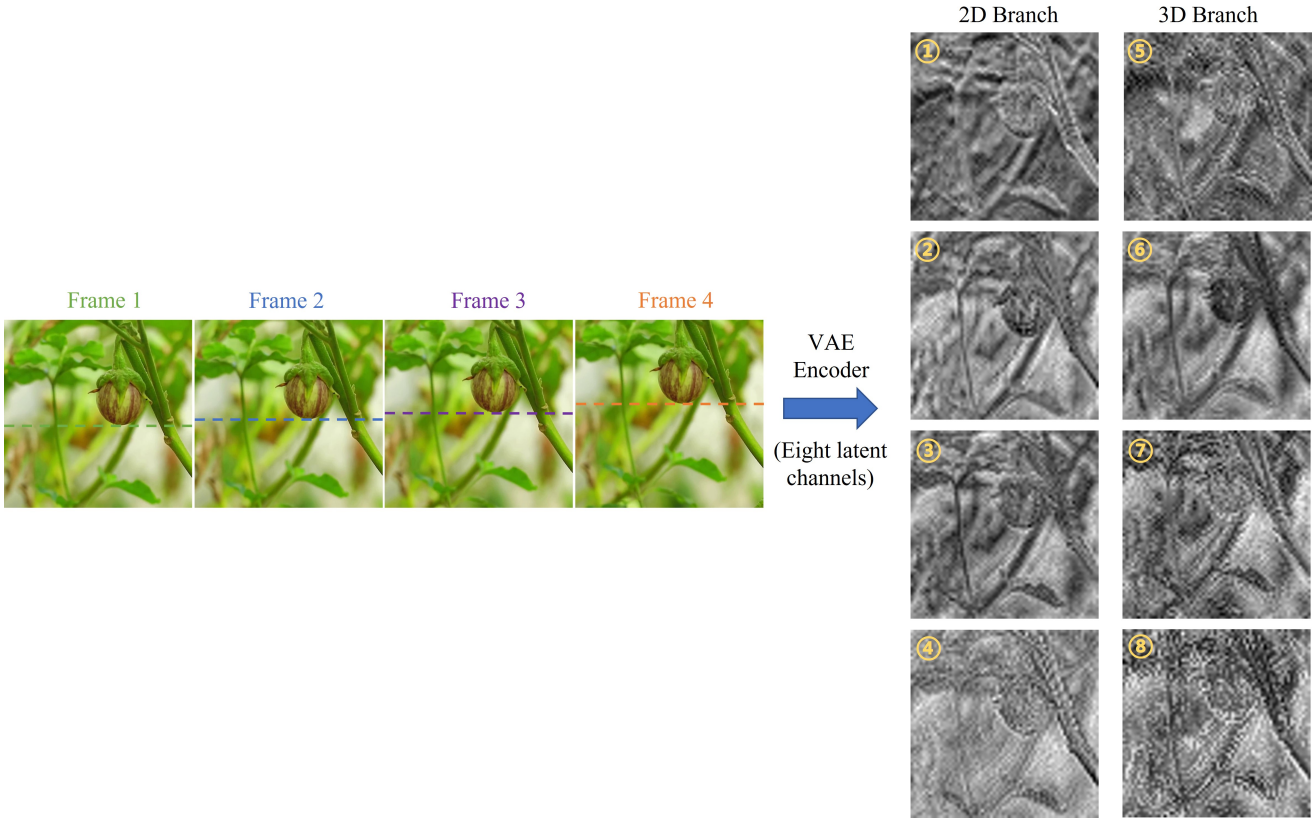

Figure 10. **Visualization of latent channels using $Z = 8$ model.** A frame group (4 frames) is mapped as one frame of 8 channels of features in the latent space. In the left figure, the positions of the fruits are labeled, indicating that the positions of the fruits are different in different frames. In the right figure, serial numbers 1 to 4 are the features of the four latent channels output from the 2D branch, and 5 to 8 are the features of the four latent channels output from the 3D branch.