Supplementary Materials of MoStGAN-V: Video Generation with Temporal Motion Styles

Contents

1. Experiments

1.1. Motion Diversity Loss

In Section 3.4 we introduce motion diversity loss L_{div} to encourage motion styles to be disentangled from each other for modeling various motions pattern. Table 1 shows the ablations for w/ or w/o L_{div}. When adding L_{div}, FaceForensics and CelebV-HQ gain a little improvement with FVD_{128} from 79.8 to 72.6 and 166.1 to 132.1 respectively, however, for SkyTimelapse the performance drops a little. One possible reason is that SkyTimelapse contains only simple motions like clouds slowly moving in one direction.

Method	**FaceForensics 256^2**		**SkyTimelapse 256^2**		**CelebV-HQ 256^2**	
	FVD$_{16}$ ↓	FVD$_{128}$ ↓	FVD$_{16}$ ↓	FVD$_{128}$ ↓	FVD$_{16}$ ↓	FVD$_{128}$ ↓
MoStGAN-V(ours)	39.7	72.6	65.3	162.4	56.1	132.1
w/o L_{div}	40.4	79.8	65.8	143.2	58.4	166.1

Table 1. Ablations on different rank.

1.2. Magnitude of Latent Motion Vectors

In addition to content latent vector w in original StyleGAN2 [6], we propose to separate motion style from original content vector for better motion synthesis controlment. Since both will later be transformed into style parameters to modulate the weights of each Synthesis Layer, here we want to explore the importance of each contribution by changing the dimension of each motion latent vector $\{m_t^k\}_{k=1,2,...,K}$ after the Motion Network $\mathsf{F_m}$ and remaining the dimension of latent vector w as 512, same as the original model.

Dimension	**FaceForensics 256^2**		**SkyTimelapse 256^2**		**CelebV-HQ 256^2**	
	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$
d=64	72.5	153.6	78.6	162.3	76.3	158.0
d=128 (default)	39.7	72.6	65.3	162.4	56.1	132.1
d=256	52.4	125.9	74.7	161.3	64.9	150.7

Table 2. Different dimensions of latent motion vector m_t^k.

1.3. Number of rank

Table 3 shows the result of different rank R for low-rank factorization. Increasing the rank could not improve the performance much but introduce more parameters instead, thus we set $R = 1$ as default.

Rank	**FaceForensics 256^2**		**SkyTimelapse 256^2**		**CelebV-HQ 256^2**	
	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$
R=1(default)	39.7	72.6	65.3	162.4	56.1	132.1
R=3	40.6	70.4	65.7	154.1	56.6	136.9
R=5	39.5	74.9	75.5	163.2	53.8	152.8
R=10	37.6	64.2	83.3	174.0	55.7	157.6

Table 3. Ablations on different rank.

1.4. Generating videos with diverse background

UCF101 [10] is a challenging dataset with large variations, e.g., actions, camera motion, object appearance. As Table 4 shows, all models struggle to achieve a satisfying FVD and our method performs slightly better. In addition, we also conducted experiments on Horseback [1] (resized to 256^2) with moving camera. Compared with StyleGAN-V [9] with FVD_{16} performance of 168.14, our model achieves better result of 146.65.

Methods	$\text{FVD}_{16}\downarrow$	$\text{FVD}_{128}\downarrow$
MoCoGAN-HD [11]	1729.6	2606.5
DIGAN [13]	1630.2	2293.7
StyleGAN-V [9]	1431.0	1773.4
MoStGAN-V (ours)	1380.3	1695.6

Table 4. Comparison on UCF101 [10] dataset.

Methods	$\text{FVD}_{16}\downarrow$	Params $\downarrow$
StyleGAN-V$_{text}$ [9]	191.1	58M
MUGEN [5]	112.7	120M
TATS [4]	89.3	562M
MoStGAN-V$_{text}$ (ours)	129.8	66M

Table 5. Comparison on text-conditional MUGEN [5].

1.5. Extending to text-conditional video generation

We extended our approach to a text-conditional model (denoted as MoStGAN-V$_{text}$) with two granularity information as inputs. In sentence level, we concatenate noise vector z^c with text embedding and pass it through Mapping Network F_c to obtain latent content vector w; while in word level, we extract words embeddings for latent motion vectors m_t^k (since MUGEN [5] dataset contains frame-level fine-grained information). We enable text-conditional StyleGAN-V [9] (denoted as StyleGAN-V$_{text}$) by concatenating noise vector z^c with text embedding. Table 5 shows our method is superior to StyleGAN-V [9]. Compared to two-stage models TATS [4] and MUGEN [5], i.e., firstly train a VQGAN [3] and a Transformer in the second stage, our model achieves comparable result with efficiency. This is a toy model to show our method is extendable for conditional generation and we leave further improvement for future exploration.

2. Properties

2.1. Motion Style Interpretation

We investigate how different motion styles exert influence to generated frames. During inference, we analyze how different motion styles response to the output feature by calculating attention map with dimesionality $K \times H \times W$ between attention matrix $\mathbf{A_t} \in \mathbb{R}^{K \times c_{out}}$ and output feature of the `Modconv2d` $F_t \in \mathbb{R}^{c_{out} \times H \times W}$ in time step t, where K corresponds to K motion styles. For better visualization, we calculate attention map in last `Modconv2d` layer of synthesis block 256^2 and add it on top of the final generated frames. Note that we do not focus on disentangled representations, i.e., considering a latent representation to be perfectly disentangled if each latent dimension controls a single visual attribute [2, 7], but only want to explore the influence of our proposed motion styles. Figure 1 shows that attention map for each motion style tends to focus on different regions correspond to different motion patterns, e.g., blinking eyes or open mouth.

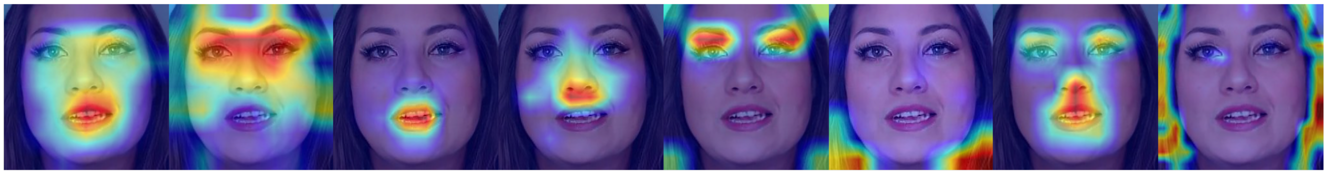


Figure 1. Each grid represents for implementing attention map on top of generated frames for different motion styles at the same time step.

2.2. Motion Content Decomposition

For better observation, we control univariate to show how additional motion style achieves better motion synthesis. Each column of Figure 2 shows diverse motions originates the same content z^c that controls the appearance variances, while each row shares same motion noises $z^m_{t_0}, ..., z^m_{t_n}$ with variant identities. Our model perform better than StyleGan-V that can ensure sampling from same motion codes will lead to consistent motions (see each rows, e.g., blink eyes, head posture, mouth opening size and direction.) Please watch grid videos in our anonymous webpage for better observation.

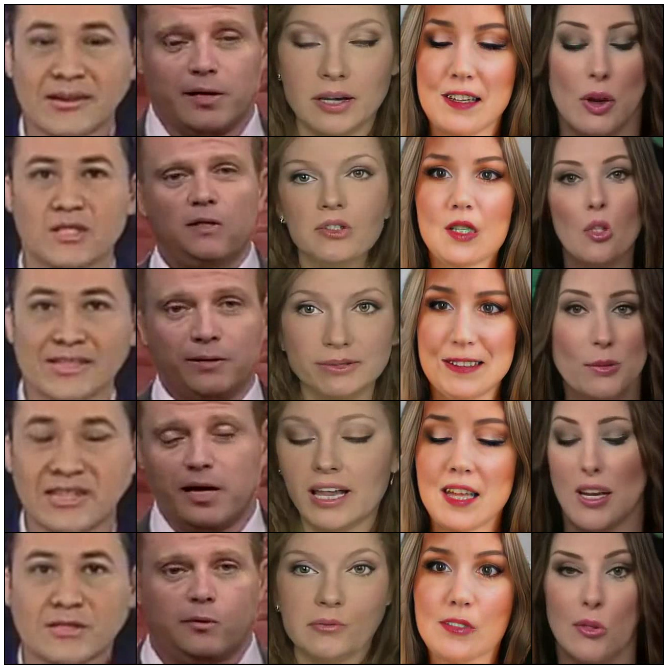
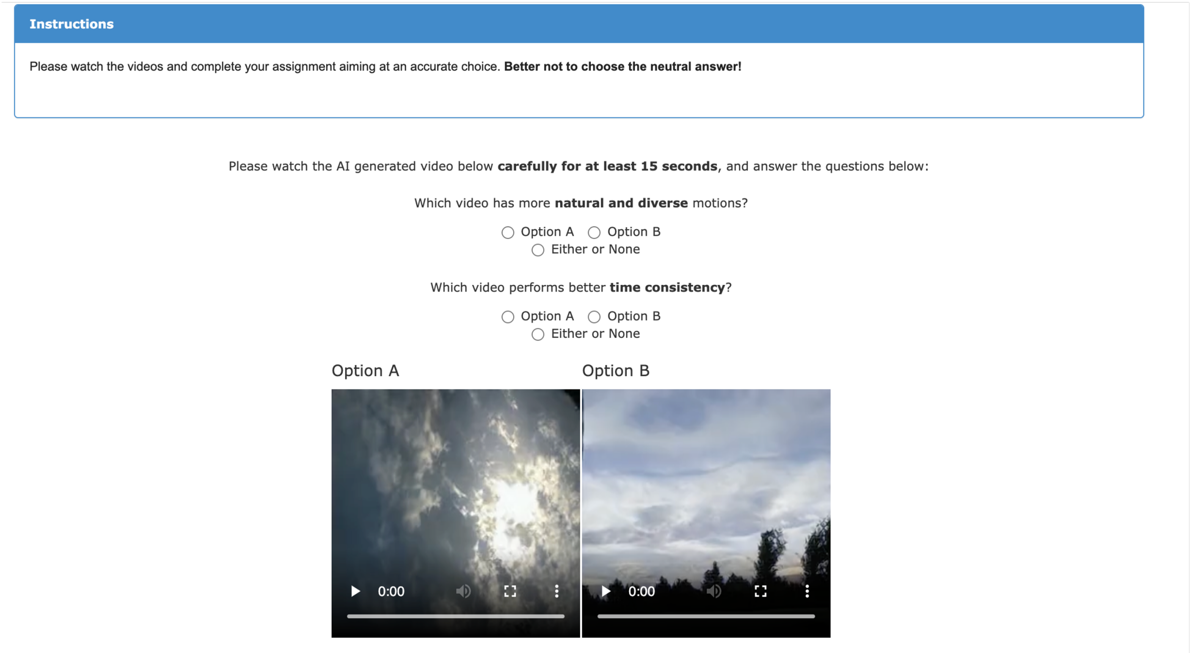


Figure 2. Each row sample from same motion noise $z^m_{t_0}, ..., z^m_{t_n}$ while each column starts from same content noise z^c and each grid presents different videos at same time step.



Instructions

Please watch the videos and complete your assignment aiming at an accurate choice. **Better not to choose the neutral answer!**

Please watch the AI generated video below **carefully for at least 15 seconds**, and answer the questions below:

Which video has more **natural and diverse** motions?

○ Option A ○ Option B
○ Either or None

Which video performs better **time consistency**?

○ Option A ○ Option B
○ Either or None

Option A Option B

Figure 3. Webpage interface for human evaluation.

3. Efficiency of low-rank factorization

The hypernetworks are introduced to produce modulation matrix for weights of each `ModConv2d` layer. If the hidden layer size of hypernetwork is of dimensionality $d_h = 128$ and the deconvolutional weight tensor is the size of $d_o = c_{out} \times c_{in} \times k_h \times k_w = 512 \times 512 \times 3 \times 3 \approx 2.4$ million, where c_{out}, c_{in} is the dimensionality of output and input channel, and k_h, k_w are the kernel size of deconvolutional filters. Then the output weight matrix in the hypernetwork will be of size $d_h \times d_o \approx 0.3$ billion, which is memory-intensive for each layer. To alleviate this issue, we leverage low-rank factorization for the modulation matrix. The modulation tensor for each motion style of each `ModConv2d` layer within our proposed MoStAtt as Section 3.3 mentioned should have the dimensionality of $c_{in} \times k_h \times k_w$. Here we omit the rank R for simplicity. To make a light-wise architecture in a parameter-effective manner, we decompose the modulation matrix with low-rank factorization. Therefore, the output dimensionality of hypernetwork will be reduced from $c_{in} \times k_h \times k_w$ to $c_{in} + k_h + k_w$. Therefore, by using low-rank factorization, the number of parameters of hypernetwork in current `ModConv2d` layers will be reduced from 0.3 billion to $d_h \times (c_{in} + k_h + k_w) \approx 0.07$ million, which largely reduces the computation burden.

4. Human Evaluation

We conducted a human evaluation on Amazon Mechanical Turk to assess videos generated by our method in comparison to StyleGAN-V [9] w.r.t motion diversity as well as temporal consistency. We provide 100 pairs of videos each for three datasets FaceForensics, RainbowJelly and SkyTimelapse and pairwise compare 2 random videos in random order from either our method or StyleGAN-V [9] trained on the same dataset. And we provide the Mturker with two questions: *'Which video has more natural and diverse motions?'* and *'Which video performs better time consistency?'*, based on which they will choose their preference and if it is hard to decide a better one, we also provide a neutral option. Figure 3 shows how the interface guide users to conduct the evaluation. Each video pair is assigned to 5 unique workers, resulting in 500 responses for each dataset. The average time per assignment is 10 minutes and 37 seconds.

5. Datasets details

FaceForensics [8] FaceForensics is a video dataset where all videos are downloaded from Youtube and cut down to short continuous clips that contain mostly frontal faces. We follow previous work[1] to extract face crops for this talking head

[1] https://github.com/universome/stylegan-v/blob/master/src/scripts/preprocess_ffs.py

datasets, which crops each frame independently and somehow makes unstable shaking.

CelebV-HQ [14] CelebV-HQ is a large-scale high-quality video dataseet with rich celebrity identities and actions. We preprocess the dataset with official link[2].

SkyTimelapse [12] SkyTimelapse contains slow motions of sky changing under different time and weather conditions. We use the released 2,377 videos from official dataset although they [12] claims that 5, 000 videos are collected[3].

RainbowJelly [9] RainbowJelly dataset is an 8-hour-long movie of 4K resolution underwater video with colorful jellyfishes. We follow previous work[4] and divide it into clips of 512 frames each. We use it as a video generation benchmark to evaluate the effect of our MoStAtt since it contains complex hierarchical motions.

For all the datasets, we center crop to 256^2 resolution and use train split if available to train all the models. All the dataset are 25 fps except RainbowJelly is 30 fps.

References

[1] Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei A Efros, and Tero Karras. Generating long videos of dynamic scenes. *arXiv preprint arXiv:2206.03429*, 2022. 2

[2] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. 3

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[4] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. *arXiv preprint arXiv:2204.03638*, 2022. 2

[5] Thomas Hayes, Songyang Zhang, Xi Yin, Guan Pang, Sasha Sheng, Harry Yang, Songwei Ge, Qiyuan Hu, and Devi Parikh. Mugen: A playground for video-audio-text multimodal understanding and generation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 431–449. Springer, 2022. 2

[6] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 1

[7] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems*, 31, 2018. 3

[8] 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. 4

[9] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. 2, 4, 5

[10] 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. 2

[11] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021. 2

[12] 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. 5

[13] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571*, 2022. 2

[14] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. *arXiv preprint arXiv:2207.12393*, 2022. 5

[2] https://github.com/CelebV-HQ/CelebV-HQ/blob/main/download_and_process.py

[3] https://github.com/weixiong-ur/mdgan

[4] https://github.com/universome/stylegan-v/blob/master/src/scripts/convert_video_to_dataset.py