Supplementary Materials for CVPR'23 Paper Titled "Conditional Image-to-Video Generation with Latent Flow Diffusion Models"

A1. Potential Negative Social Impact

Conditional image-to-video models can be used for unethical purposes [8], *e.g.*, creating videos of celebrities for fake news spreading. We will restrict the usage of our models to research purposes only. We also plan to investigate some fake video detection techniques [1] that may be effective in detecting fake videos like the ones generated by our methods.

A2. Additional Experiments

A2.1. Additional Ablation Study on Network Architecture

To evaluate the performance difference of our proposed LFDM with different architectures, we change the depth of the image decoder Ω in stage-one LFAE (Table A1) and the 3D U-Net ϵ_θ in stage-two DM (Table A2). We experiment with different settings on MUG dataset to generate videos of 128×128 frame resolution.

In our default setting, the image decoder Ω in stage-one LFAE is implemented with a network including 6 residual blocks and 2 up-sampling blocks. In Table A1, we compare using different network depths for the image decoder Ω in stage-one LFAE . We add four extra residual blocks to the decoder Ω. So the number of residual blocks is increased from 6 to 10. Then we only retrain this deeper decoder in stage one, while keeping all the remaining modules unchanged. As Table A1 shows, using a deeper image decoder shows slightly better self-reconstruction performance (as measured by L_1 error) but fails to generate higher-quality videos (as measured by FVD). Therefore, we keep using 6 residual blocks in our experiments.

In our default setting, the denoising network ϵ_θ employs a 3D U-Net architecture including 4 down-sampling and 4 up-sampling 3D convolutional blocks, where the *channel multipliers* are (1, 2, 4, 8) with a base channel of 64. That is, from highest to lowest resolution, the 4 down- or up-sampling blocks in ϵ_θ use (1×64, 2×64, 4×64, 8×64) channels, respectively. In Table A2, we compare using different channel multipliers in stage-two DM . We add one more layer to the down-sampling and up-sampling blocks of the 3D U-Net and the channel multipliers are (1, 2, 4, 8, 16) with a base channel of 64. We retrain this deeper DM in stage two with 1,200 training epochs as in our previous simpler DM training. We keep using the same stage-one LFAE. From Table A2, one can observe that using more layers in DM led to decreased performance. Therefore, we adopt the simpler (1, 2, 4, 8) as the default setting of channel multipliers in our stage-two DM.

# Residual Blocks	L_1 error↓	FVD↓
6	0.418	**32.09**
10	**0.371**	32.83

Table A1. Comparison using different numbers of residual blocks in the image decoder Ω of stage-one LFAE.

Channel Multipliers	FVD↓
(1, 2, 4, 8)	**32.09**
(1, 2, 4, 8, 16)	68.07

Table A2. Comparison using different channel multipliers in the network ϵ_θ of stage-two DM.

A2.2. Additional Analysis of Flow and Occlusion Maps

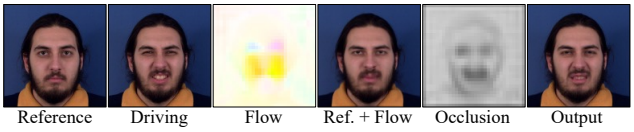

Figure A1. Visualization of flow and occlusion map. [Ref. + Flow] is generated by applying the flow to reference image; note the change in head pose and the shape of eyes and mouth after applying flow. Occlusion map further masks the eyes and mouth to help decoder generate novel pixels for these parts in output image.

Figure A1 shows the visualization of flow and occlusion map of one example video frame from MUG dataset. As illustrated in the caption of Fig. A1, without using occlusion map, decoder may need to learn which regions should be kept and which regions should be masked and repainted. Our additional experiments show that retraining LFAE without occlusion maps increases the L_1 error of self-reconstruction from 0.418 to 0.450 on MUG dataset.

A2.3. Comparison of Inference Time among Different Models

Table A3 shows the average inference time of each method to generate one video when using batch size 10 on one NVIDIA A100 GPU on MUG dataset. Note that VDM uses 200-step DDIM while both LDM and LFDM employ 1000-step DDPM.

Model	ImaGINator	VDM	LDM_{64}	LFDM_{64}	LDM_{128}	LFDM_{128}
Time(s)	0.9	23.1	8.0	8.8	25.5	36.0

Table A3. Inference time comparison among different methods.

A3. More Discussion about Future Work

Several limitations and some future work are discussed in Section 5 of the paper. Here we elaborate more on future work about LFDM. One future direction is to enable the generation of a video with changing background (or context). We plan to first utilize our LFDM to generate a video describing the motion of foreground subject, and then design another generative network conditioned on each generated foreground frame to synthesize the changing background for each frame. In addition, to enhance the generalization ability of LFDM on generating diverse motions of more categories, we plan to collect more labeled training video datasets and apply some continual/incremental learning techniques such as [4–7] to train our LFDM. Finally, in our experiments (Table 6), we noticed that 10-step DDIM can achieve acceptable generation performance with faster sampling speed, suggesting it may have greater potential with better hyperparameter settings. To explore these settings, including diffusion sampling steps, we plan to employ some recent hyperparameter optimization techniques such as [2, 3, 9].

A4. Information about Attached Videos

We attach seven MP4 files of example video clips generated by our proposed method in Supp. materials[1]. All the given images are testing (*unseen*) images.

- **mug.mp4** shows the synthesized video clips displaying all 7 expressions of one subject from MUG dataset.
- **mhad1.mp4** and **mhad2.mp4** include the generated video clips for 26 actions of one subject from MHAD dataset. We exclude the action *sit to stand* because the subject in the given image is standing.
- **natops.mp4** shows the synthesized video clips containing all 24 gestures of one subject from NATOPS dataset.
- **new_domain.mp4** shows the synthesized video clips including 4 expressions of four subjects from FaceForensics dataset. "Original" means directly applying our LFDM pretrained on MUG dataset. "Finetuned" means that the image decoder is finetuned with the *training* videos from FaceForensics dataset. Note that other modules including stage-two DM are still unchanged during finetuning. From this video, one can observe that our original LFDM can generate acceptable results for given subject images from a new domain and achieve better performance when the decoder is finetuned with training videos from the new domain.
- **mug_ddim.mp4** shows the synthesized video clips containing 4 expressions of four subjects from MUG dataset. "DDIM-10" means using 10-step DDIM for diffusion sampling while "DDPM-1000" is our default 1000-step DDPM sampling strategy. From this video, one can observe that 10-step DDIM can generate visually-acceptable videos with faster sampling speed (0.3s per video *vs*. 36s per video when using DDPM-1000). But note that the FVD score of DDPM-1000 is still noticeably better than DDIM-10 (32.09 *vs*. 50.18) so we keep DDPM-1000 as our default setting.
- **sota.mp4** is a video for comparison between our proposed LFDM and several other models including ImaGINator, VDM, and LDM. We show synthesized video clips by each model on 3 subjects from MUG, MHAD, and NATOPS datasets. The video frames of ground truth (GT) and results of LDM and our LFDM have 128×128 resolution while results of ImaGINator and VDM are 64×64. The original video clips generated by ImaGINator only contain 32 frames. So we repeat the first frame and the last frame four times to make all the displaying videos have 40 frames.

[1]These videos are also available in https://github.com/nihaomiao/CVPR23_LFDM.

References

[1] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and Alberto Del Bimbo. Deepfake video detection through optical flow based cnn. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 1

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012. 2

[3] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017. 2

[4] Mingfu Liang, Jiahuan Zhou, Wei Wei, and Ying Wu. Balancing between forgetting and acquisition in incremental subpopulation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 364–380. Springer, 2022. 2

[5] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4453, 2021. 2

[6] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022. 2

[7] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2

[8] 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. 1

[9] Shaokun Zhang, Feiran Jia, Chi Wang, and Qingyun Wu. Targeted hyperparameter optimization with lexicographic preferences over multiple objectives. In *The Eleventh International Conference on Learning Representations*, 2023. 2