Supplementary: Unsupervised Facial Performance Editing via Vector-Quantized StyleGAN Representations

Berkay Kicanaoglu Pablo Garrido Gaurav Bharaj

Flawless AI

{berkay.kicanaoglu, pablo.garrido, gaurav.bharaj}@flawlessai.com

1. Architectural Details

In the following, we provide more details of the network architecture used in Stage-II.

Stage-II Generator's Architecture Tab. 1 describes the layers of the generator architecture G_s, which is illustrated with a default codebook of size $K = 128$. We borrow SPADE residual blocks from the original implementation described in [3][1]. First, we upsample the feature map resolution by a factor of two using *bilinear interpolation*. Then, we pass the resulting feature maps through two blocks of `[Conv2d, BatchNorm2d, ReLU]`. Here, we set the convolution *kernel size* and *stride* to 3 and 1, respectively.

Stage-II Discriminator's Architecture We employ a patch discriminator as described in Tab. 2. Specifically, we utilize a residual *Discriminator Block* where the input feature maps are decimated by a factor of 2 at the end of each block. The last discriminator layer outputs a 16×16 feature map that we filter with a sigmoid activation function before computing the Binary Cross Entropy loss.

2. Quantitative Comparisons

In this section, we quantitatively compare the quality of our synthesized videos with existing baselines, namely FOMM [5], PTI [4], and DVP [2]. In the following, we provide details of our baselines, datasets, and evaluation settings.

Baselines. We compare our method against existing baselines regarding reconstruction and self-reenactment quality.

[1]The original implementation of the SPADE resnet block is available at the following link: github.com/NVlabs/SPADE/blob/fecacc920c1367a038995c45a39c15f6521ca64f/models/networks/architecture.py

Layer	Modules	Mask Dim.	Input Shape (num_cin, H, W)	Output Shape (num_cout, H, W)
1	Constant	-	-	(1024 ,4 ,4)
	SPADE Res. Block	384	(1024, 4, 4)	(512, 4, 4)
	Upsample	-	(512, 4, 4)	(512, 8, 8)
2	SPADE Res. Block	384	(512, 8, 8)	(256, 8, 8)
	Upsample	-	(256, 8, 8)	(256, 16, 16)
3	SPADE Res. Block	384	(256, 16, 16)	(128, 16, 16)
	Upsample	-	(128, 16, 16)	(128, 32, 32)
4	SPADE Res. Block	384	(128, 32, 32)	(128, 32, 32)
	Upsample	-	(128, 32, 32)	(128, 64, 64)
5	SPADE Res. Block	384	(128, 64, 64)	(128, 64, 64)
	Upsample	-	(128, 64, 64)	(128, 128, 128)
6	SPADE Res. Block	384	(128, 128, 128)	(64, 128, 128)
	Upsample	-	(64, 128, 128)	(64, 256, 256)
7	SPADE Res. Block	384	(64, 256, 256)	(64, 256, 256)
	Upsample	-	(64, 256, 256)	(64, 512, 512)
8	SPADE Res. Block	384	(64, 512, 512)	(32, 512, 512)
	Upsample	-	(32, 512, 512)	(32, 1024, 1024)
9	ToRGB	-	(32, 1024, 1024)	(3, 1024, 1024)

Table 1. **Stage-II generator (G_s) architecture.** Since we give a temporal window of size 3, input dimensionality of SPADE blocks is three times the codebook size K.

When comparing PTI and FOMM, we test the ability to synthesize the input training videos at high fidelity. We deem such a comparison fair as PTI cannot do inference. To run PTI, we reconstruct each video sequence separately using inversion and *pivotal tuning*. As suggested by the authors, we run pivotal tuning for 80 epochs to finetune the pretrained

Layer	Modules	Input Shape	Output Shape
1	Discriminator Block	(3, 1024 ,1024)	(64, 512, 512)
2	Discriminator Block	(64, 512, 512)	(128, 256, 256)
3	Discriminator Block	(128, 256, 256)	(256, 128, 128)
4	Discriminator Block	(256, 128, 128)	(256, 64, 64)
5	Discriminator Block	(256, 64, 64)	(256, 32, 32)
6	Discriminator Block	(256, 32, 32)	(1, 16, 16)

Table 2. **Stage-II discriminator (D_s) architecture.**

StyleGAN. We do not overfit PTI to the input training video for a fair comparison. To generate a video with FOMM, we take the first frame of the video as the source image. We then animate the source image with the input training video as the driving sequence. Note that we utilize the latest pretrained checkpoint provided at the author's official website[2].

To compare our approach with DVP, we drive the learned model on a held-out video sequence not used for training. Here, we test the capability of DVP and our method to reanimate the driving video faithfully. Following [2], we train a DVP model from scratch on training videos until convergence. Specifically, we train the DVP model for 26 epochs, whereas we stick to our standard 6-epoch training. Once the model is trained, we synthesize the held-out sequence by running DVP on input conditioning images generated from the tracked parameters in the driving video.

For a fair comparison with the baselines that synthesize artifact-prone and noisy backgrounds, such as PTI, we segment out the background from all the generated results when computing the metrics. To compare our method with DVP, we particularly employ their dilated face masks for quantitative comparisons. However, we have found that in running our method with tighter non-dilated masks we obtain slightly better image-based metrics.

Datasets. For comparisons with PTI and FOMM, we curate a subset of video sequences containing three subjects (*M13*, *M27*, and *W09*) from MEAD dataset [6]. For each subject, we pick seven sequences, each captured under three different viewpoints: front, left, and right view. The first sequence shows the subjects reciting a dialog at neutral pose, i.e., no emotions. The three subsequent sequences show the subjects speaking at different anger levels, while the last three sequences show the subjects speaking at different happiness levels.

[2] github.com/AliaksandrSiarohin/first-order-model

3. Additional Experiments

In this section, we present additional results and a user study to understand editing capabilities.

3.1. VoxCeleb2HQ

In addition to MEAD and Obama sequences, we test our method on VoxCeleb2HQ sequences. We particularly downloaded two celebrity videos of Viggo Mortensen (`7xaAmL5lPFs`) and Richard E. Grant (`nnLOrbt6D74`) from YouTube[3]. Using the provided meta information, we partition each video into shots that show the celebrity speaking. After face detection and cropping, we end up with approximately $4k$ and $1.5k$ video frames from each subject, respectively. These videos exhibit rapid head pose changes, complex lighting variations, and occlusions. Our method can still cope with these challenges, as shown in the supplementary video.

We further devise a cross-shot expression transfer experiment as follows: We utilize a non-overlapping shot as a driver sequence to synthesize novel mouth expressions for another source shot of the same subject. More specifically, we simply copy the tracked expression parameters of the driver onto the source, then render novel half-face input masks, and finally run our approach with the newly generated masks. Our method successfully handles expression transfer even when the driver and source videos exhibit different head poses, which can generally be deemed as a non-trivial task, thanks to our face-tracking-enabled synthesis approach. We demonstrate expression transfer results in the supplementary video. In certain sections of generated videos, we observe some light-flicker-like artifacts. These artifacts might be ascribed to high-frequency artifacts of the rendered half-face masks during expression transfer. This is an interesting research problem that requires further study in spatio-temporal modeling to improve robustness.

3.2. User Study

We conduct a user study to assess how convincing our edits are. To this end, we prepared a survey and collected different opinions, mainly from subjects with post-production/VFX backgrounds. In total, we reached out to 20 subjects. In the survey, we ask subjects to rate edits between 1 (not convincing) and 5 (convincing) for three different groups: *eye and gaze*, *mouth and mouth interior*, and *teeth removal*. Since some of the edits can be uncanny, e.g., showing very prominent front teeth, we ask users to rate as if they came across the actors for the first time. The user preferences are aggregated across different edits under each group, as shown in Fig. 1.

From the results in Fig. 1, we observe that our edits are deemed plausible as most users score them as "convincing"

[3] www.youtube.com

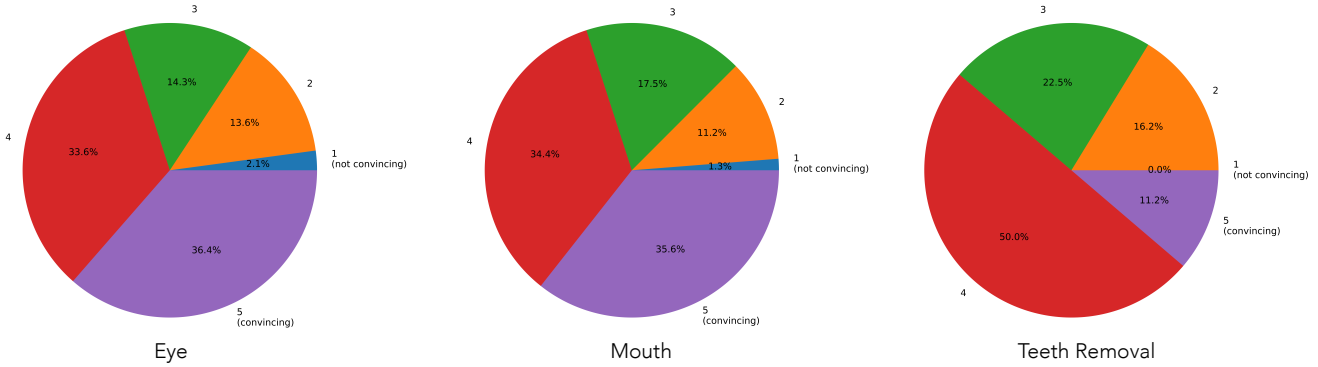

Figure 1. **User study for semantic edits.** We ask users to rate manual semantic edits over three groups: eyes, mouth and teeth. Overall, the edits are found plausible except when editor mistakes lead to uncanny facial appearances. This is especially a good example of uncanny valley effect as human perception is very sensitive to facial appearance and geometry.

across all groups. A deeper analysis of our results reveals that small errors in manual editing leads to lower ratings. One of such errors is the mismatch in pupil size (after semantic editing). Users usually detect such abnormal appearance of facial parts and find it less convincing. Another example of users giving lower scores is when the quality with which topological changes, such as eye and mouth opening, are executed is subpar, i.e., there is human editor error. This relates to the well-known uncanny valley effect[4]. Similarly, when the proportion of edited facial features looks physically incorrect, users find the edits less plausible. However, this is mainly as a result of clumsy edits but not of the model's editing or reconstruction sensitivity.

3.3. Comparison against MegaPortraits [1]

We run a qualitative comparison against the one-shot talking-head generation method by Drobyshev *et al*. [1]. For this comparison, we use the held-out sequence of Obama video as in DVP comparisons. We have asked the authors to run a self-reenactment task over this sequence. Similar to ours, MegaPortrait can synthesize videos at a resolution of 1024×1024. However, their output is sharper compared to ours. On the other hand, in comparison to ours, we observe that MegaPortraits struggles with mouth interior in terms of temporal stability/coherence and fails to maintain the teeth identity of the original subject, see Fig. 2. Furthermore, it exhibits texture-sticking phenomenon known to exist with StyleGAN2-like generators. Please refer to the supplementary video for comparison.

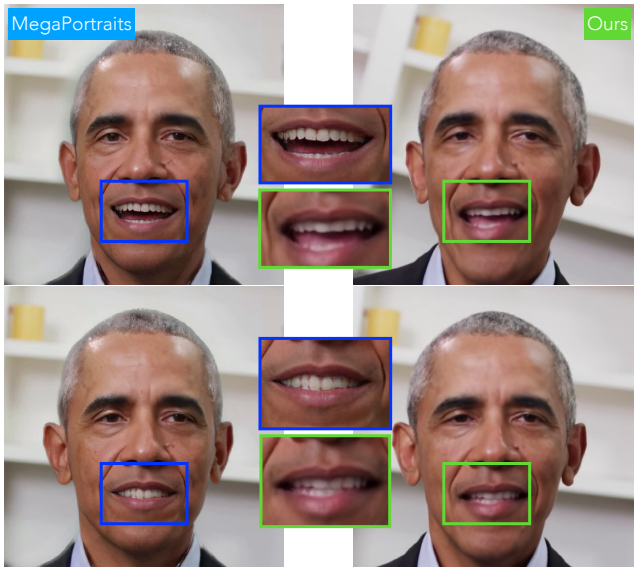

Figure 2. **Megaportraits vs Ours.** Our method preserves the person-specific mouth geometry and teeth identity in comparison. Note that the shown frames are representative but not exactly the same, mainly because MegaPortraits result uses an unknown driver sequence for self-reenactment.

References

[1] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2663–2671, 2022. 3

[2] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM TOG*, 37(4):163, 2018. 1, 2

[3] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346. Computer Vision Foundation / IEEE Computer Society, 2019. 1

[4] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM TOG*, 42(1), 2022. 1

[5] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, pages 7135–7145, 2019. 1

[4] en.wikipedia.org/wiki/Uncanny_valley

[6] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *ECCV*, 2020. 2