Face2Diffusion for Fast and Editable Face Personalization

Supplementary Material

Figure 8. **Expression-conditional generation.**

7. Expression-Conditional Generation

Although our expression guidance aims to disentangle face expressions from face embeddings S^*, it also enables F2D to generate conditioned face images by reference expressions. We show the examples of the expression-conditioned generation in Fig. 8. The reference images are sampled from the DFD [2] dataset.

8. Comparison with More Recent Models

We additionally compare our model with the two recent models, OFT [17] and DVAR [23], in Table 6. Our method significantly outperforms such the recent models on hMean and gMean.

9. Test Prompts

We give the set of text prompts used in our experiments in Table 7. Our prompts include various scenes related to job, activity, expression, and location.

10. More Visual Comparisons

We show additional examples in Figs. 9 and 10. For enhanced visibility, we compare our method with CustomDiffusion [16], CelebBasis [27], and FastComposer [25] that are ranked in the top-5 in Identity×Text scores in Table 1.

	AdaFace	SphereFace	FaceNet	CLIP	dCLIP	SigLIP	hMean	gMean
OFT	0.3446	0.3980	0.4673	0.2245	0.1364	0.3515	0.0615	0.0993
DVAR	0.0452	0.0939	0.1201	0.2710	0.1852	0.4261	0.0369	0.0548
Ours	0.3143	0.4215	0.5313	0.2486	0.2020	0.3856	**0.1749**	**0.2252**

Table 6. **Comparison with the more recent methods.**

Prompts
A photo of S^* as a firefighter
A photo of S^* as a cowboy
A photo of S^* as a chef
A photo of S^* as a racer
A photo of S^* as a king
A photo of S^* as a scientist
A photo of S^* as a tennis player
A photo of S^* as a DJ
A photo of S^* as a knight
A photo of S^* as a pilot
A photo of S^* walking in a city under an umbrella
A photo of S^* surrounded by tall bookshelves
A photo of S^* trying on hats in a vintage boutique
A photo of S^* sipping coffee at a café terrace
A photo of S^* in a busy subway station
A photo of S^* eating ice cream at a rooftop terrace
A photo of S^* playing the saxophone on a stage
A photo of S^* running in a meadow
A photo of S^* playing chess at a wooden table
A photo of S^* knitting in a comfortable armchair
A photo of S^* yawning during a study session
A photo of S^* smiling warmly at the camera
A photo of S^* with a hand covering their mouth
A photo of S^* hugging a friend tightly
A photo of S^* flexing muscles in a gym
A photo of S^* looking shocked
A photo of S^* giving a thumbs-up
A photo of S^* looking angry
A photo of S^* sitting cross-legged on a rock
A photo of S^* wearing an oversized sweater
A photo of S^* at the Great Wall of China
A photo of S^* exploring Machu Picchu
A photo of S^* sailing near the Sydney Opera House
A photo of S^* walking through the streets of Rome near the Colosseum
A photo of S^* at the Grand Canyon in sunset
A photo of S^* enjoying cherry blossoms in Tokyo
A photo of S^* on a gondola in Venice
A photo of S^* at the Taj Mahal in India
A photo of S^* at Mount Everest Base Camp
A photo of S^* in front of Niagara Falls

Table 7. **Our prompt set.**

Figure 9. **Visual comparisons with previous methods.**

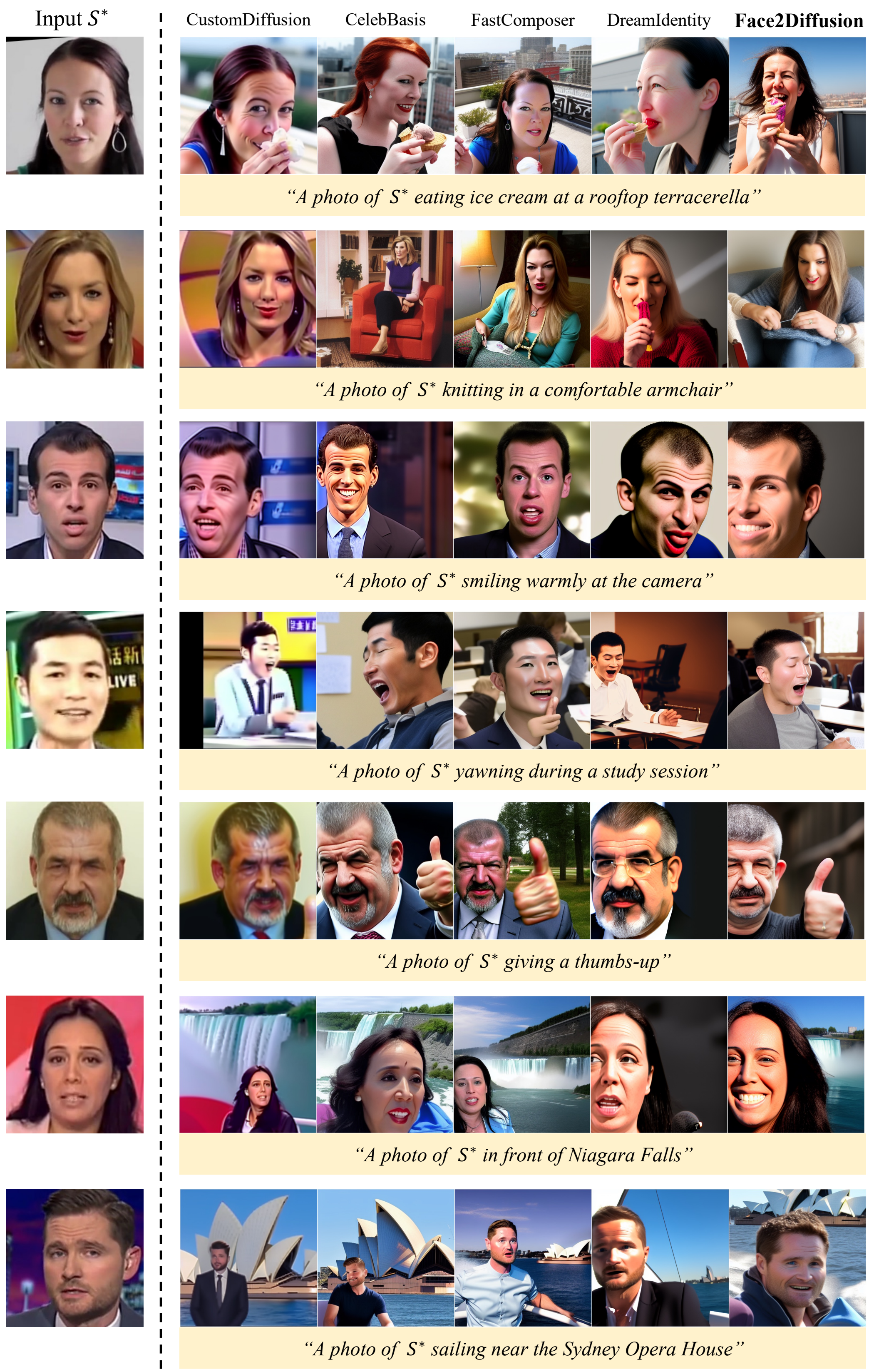

Figure 10. **Visual comparisons with previous methods.**

11. Implementation Details

11.1. Previous Methods

To conduct fair comparisons, we implement previous methods by strictly following the official instructions as much as possible. Commonly in existing methods and our Face2Diffusion, we use StableDiffusion-v1.4 (SD1.4) [19], Euler ancestral discrete scheduler [14] with 30 denoising steps, and classifier-free guidance [13] with a scale parameter of 7.0. Specific details of each method are as follows:

TextualInversion. We use the diffusers' implementation [22]. The inverted embedding is initialized by *"person"*.

DreamBooth. We use the diffusers' implementation [22]. For the prior preservation loss, we use other face images from our test set, *i.e.*, 99 identities. We set the class word for the regularization to *"a person"*.

CustomDiffusion. We use the official implementation integrated into diffusers [22]. We use the same regularization as DreamBooth.

Perfusion. We directly use the official training code [21]. The inverted embedding is initialized by *"person"*.

E4T. We directly use the official training code [11].

CelebBasis. We directly use the official training code [27].

FastComposer. We use the official implementation [25]. Because the released checkpoint is based on SD1.5, we train it from scratch on SD1.4 using the official training code. We use *"a person"* for delayed subject conditioning (DSC).

ELITE. We directly use the official pretrained model [24]. For segmentation masks during inference, we use a face-parsing model [26] that is the same one used in our CGDR.

DreamIdentity. Because there is no public implementation, we re-implement it. For the mapping MLP, we use the same architecture as our Face2Diffusion because the implementation detail is not described in the original paper. Due to the limitations of our computational resource, we train the model with eight NVIDIA A100 (40GB) GPUs which is the same cost as our F2D though the original paper [7] use its 80GB version. For self-augmented data, we collect 1K celebrity names from Internet that are consistently generated by SD1.4. Because some of proposed editing prompts do not work on SD1.4, we remove them and add alternative ones tested in the original paper. In total, we generate 8K augmented images (1K identities $\times$ 8 editing prompts).

11.2. Variants of F2D

Reconstruction. We use the same loss as Eq. 1.

Masked Reconstruction. We use a masked reconstruction loss as follows:

$$\mathcal{L} = \|(\epsilon - \epsilon_\theta(z_t, t, \tau(p))) \odot M\|_2^2. \tag{1}$$

Reconstruction w/ DSC. We implement DSC [25] on the "Reconstruction" model above. Following the official implementation, we adopt the ratio of $\alpha = 0.8$ for DSC.

ArcFace. We implement ViT [9] trained with ArcFace loss [8] using an unofficial implementation [4]. We input only the deepest layer's outputs corresponding the classifier token into the mapping network f_{map}.

ArcFace w/ MSF. We extract multi-scale features (MSF) [7] from the ArcFace model above. We use the same depth set as our F2D for MSF, *i.e.*, $\{3, 6, 9, 12\}$.

w/o Expression Guidance. We remove the concatenation before the mapping network. Therefore, the identifier S^* is computed during both training and inference as follows:

$$S^* = f_{map}(f_{id}(x)). \tag{2}$$

ControlNet. We adopt an unofficial implementation [1] of ControlNet for facial landmarks. Because the pretrained model is built on SD1.5, we train our model without expression guidance on SD1.5 and then we combine them.

11.3. Metrics

AdaFace/SphereFace/FaceNet. We use the official and unoffical implementations [3, 5, 15]. We compute the cosine similarity between extracted features of an input image x and generated image y, and then clip the value to $[0, 1]$:

$$\text{ID} = \max(\cos(f_{fr}(x), f_{fr}(y)), 0), \tag{3}$$

where $\cos$ and f_{fr} represent the cosine similarity and each feature extractor of face recognition models, respectively.

CLIP. The CLIP score [12] evaluates the cosine similarity between a generated image and input prompt p:

$$\text{CLIP} = \max(\cos(E_I(y), E_T(p)), 0), \tag{4}$$

where E_I and E_T are CLIP image and text encoders, respectively. We use the official implementation [18] of CLIP ViT-H/14 model trained on the LAION-2B [20] dataset.

dCLIP. The directional CLIP (dCLIP) score [10] evaluates the cosine similarity between the difference vectors from the reference points in the image and text space. We set the reference prompt p_r to "*A photo of a person*" and the reference image y_r to an image generated by "*A photo of S^**". The dCLIP score is computed as:

$$\text{dCLIP} = \max(\cos(\Delta y, \Delta p), 0), \tag{5}$$

$$\Delta x = E_I(y) - E_I(y_r), \quad \Delta p = E_T(p) - E_T(p_r). \tag{6}$$

We use the same encoders as the CLIP score.

SigLIP. The SigLIP score is a scaled cosine similarity between an image and text, which is computed as:

$$\text{SigLIP} = \sigma(s \cdot \cos(E_I(y), E_T(p)) + b), \tag{7}$$

where s and b are the scale and bias parameters optimized during the pretraining of SigLIP. σ represents the sigmoid function. We use the official implementation [28] of SigLIP trained on the WebLI [6] dataset.

References

[1] Face landmark controlnet. https://huggingface.co/georgefen/Face-Landmark-ControlNet. 4

[2] Deepfake detection dataset. https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html. 1

[3] Facenet pytorch. https://github.com/timesler/facenet-pytorch/. 4

[4] Insightface. https://github.com/deepinsight/insightface. 4

[5] Sphereface pytorch. https://github.com/clcarwin/sphereface_pytorch. 4

[6] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 4

[7] Zhuowei Chen, Shancheng Fang, Wei Liu, Qian He, Mengqi Huang, Yongdong Zhang, and Zhendong Mao. Dreamidentity: Improved editability for efficient face-identity preserved image generation. *arXiv preprint arXiv:2307.00300*, 2023. 4

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive Angular Margin Loss for Deep Face Recognition. In *CVPR*, 2019. 4

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 4

[10] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 4

[11] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023. 4

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 4

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021. 4

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 4

[15] Minchul Kim, Anil K Jain, and Xiaoming Liu. Adaface: Quality adaptive margin for face recognition. In *CVPR*, 2022. 4

[16] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, 2023. 1

[17] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *NeurIPS*, 2023. 1

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4

[20] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 4

[21] Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. In *SIGGRAPH*, 2023. 4

[22] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4

[23] Anton Voronov, Mikhail Khoroshikh, Artem Babenko, and Max Ryabinin. Is this loss informative? speeding up textual inversion with deterministic objective evaluation. In *NeurIPS*, 2023. 1

[24] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *ICCV*, 2023. 4

[25] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *arXiv preprint arXiv:2305.10431*, 2023. 1, 4

[26] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In *ECCV*, 2018. 4

[27] Ge Yuan, Xiaodong Cun, Yong Zhang, Maomao Li, Chenyang Qi, Xintao Wang, Ying Shan, and Huicheng Zheng. Inserting anybody in diffusion models via celeb basis. In *NeurIPS*, 2023. 1, 4

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 4