Gaussian Déjà-vu: Creating Controllable 3D Gaussian Head-Avatars with Enhanced Generalization and Personalization Abilities

Supplementary Material

Peizhi Yan[1], Rabab Ward[1], Qiang Tang[2], Shan Du[3]
[1]University of British Columbia {yanpz, rababw}@ece.ubc.ca
[2]Huawei Canada qiang.tang@huawei.com
[3]University of British Columbia (Okanagan) shan.du@ubc.ca

1. Synthetic Dataset

We use the PanoHead [2] to synthesize 18,000 identities for training. Each identity is rendered with 25 pre-defined camera views (see Figure 1 for an example).

Figure 1. An example identity generated from PanoHead, with 25 rendered views.

Figure 2 shows an example face image with its corresponding aligned FLAME [4] mesh and a face parsing mask derived using Pytorch FaceParsing model [1].

Figure 2. The first image is an example face image, the second image shows its corresponding aligned FLAME mesh and the last image shows its corresponding face parsing mask.

We use the DeepFace Pytorch Toolkit [5] to estimate the age, gender, and emotion statistics on our synthetic dataset, see Figure 3.

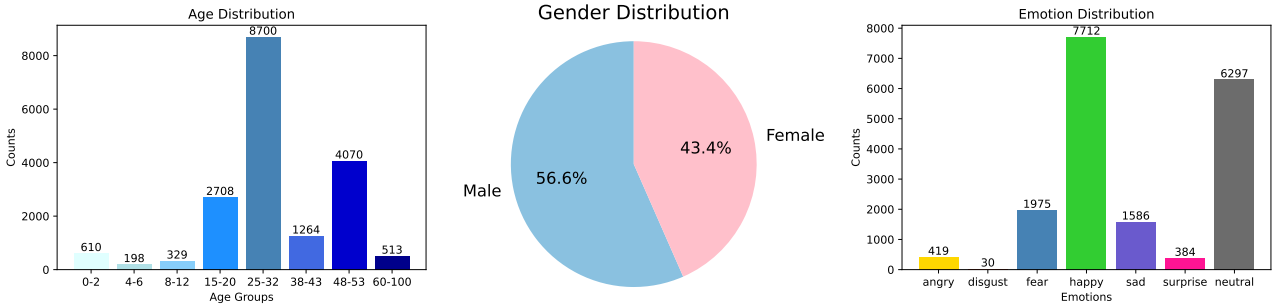

Figure 3. Age, gender, and emotion distributions evaluated on the synthetic dataset.

2. Real Dataset

Our real dataset for training the reconstruction network is derived from the FFHQ dataset [3] which originally contains 70,000 face images collected from the Internet. The reason for using the FFHQ dataset is its wide coverage of faces of various ethnicity, gender, and age groups. However, the original FFHQ dataset mainly focused on the facial region, therefore the facial region takes more space in the aligned image. This usually causes the head not to be fullying visible in the image. We first collect the raw images (without alignment) used by FFHQ and realign the faces using our standard to cover the entire head. See Figure 4 for an example.

Figure 4. We realign the raw FFHQ images to cover the entire head.

We also remove images that are not usable, for example, the images with occlusions and multiple faces (see Figure 5). The final number of images is 38,000.

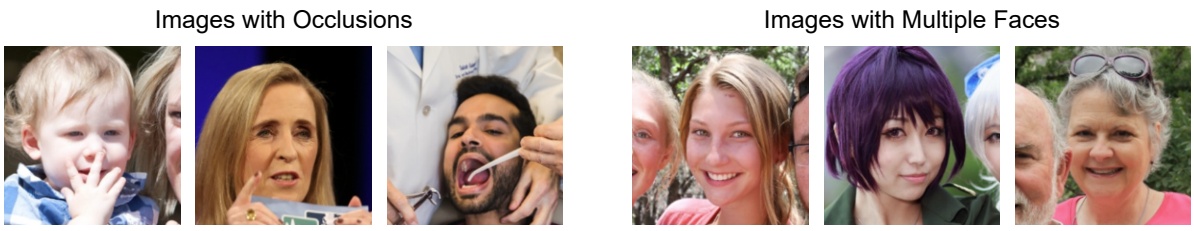

Figure 5. Example images cannot be used for training.

3. Our UV Gaussian Map

Our UV map follows the FLAME's template. However, we do not need the neck part, and the FLAME's mesh does not contain a mouth interior structure. Therefore, we remove the neck region from the original FLAME's UV map (which is the bottom part, see Figure 6), and add a mouth interior to the empty space. Figure 6 shows the UV position map and the UV position regularization weights.

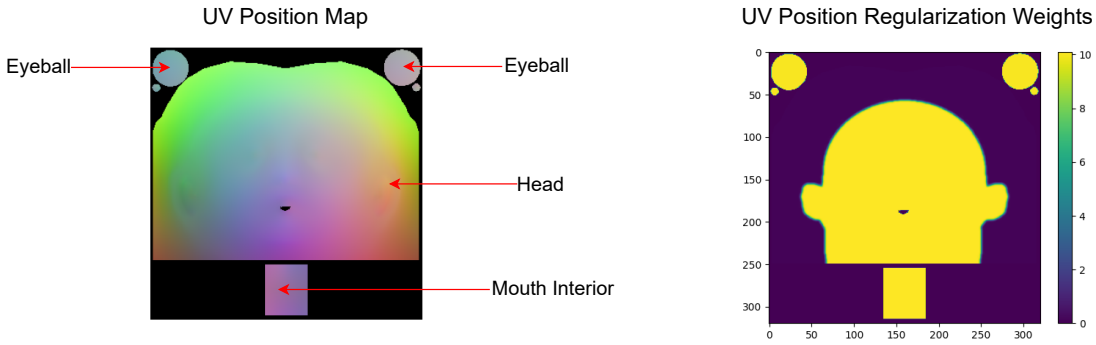

Figure 6. Our UV map structure (left) and UV position regularization weights (right).

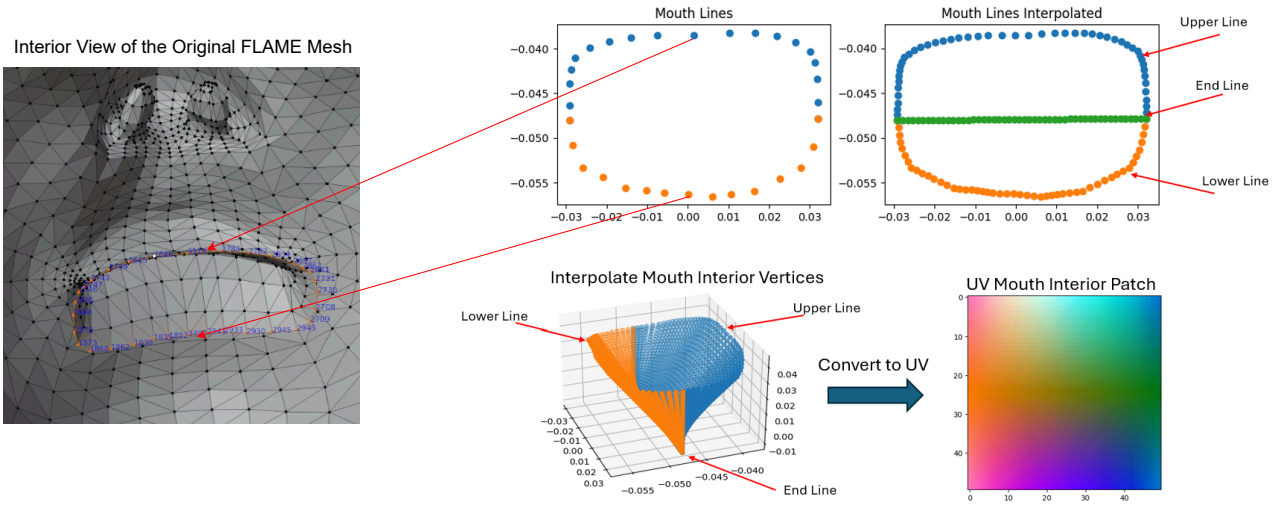

Figure 7. Our method for automatically creating the mouth interior.

We develop an automatic method to generate the mouth interior for the FLAME mesh. We first select the contour of vertices around the mouth inside the FLAME mesh and split it to the upper line and lower line. Then we add an end line that

defines the end of the mouth interior. We interpolate the vertices to create a dense funnel-shaped structure that can roughly represent the mouth interior. Figure 7 illustrates the approach. Note that we only need to manually select the contour of vertices once, since all the FLAME reconstructed head meshes share the same topology. Figure 8 shows an example UV Gaussian map, and Figure 9 shows its corresponding 3D Gaussian head.

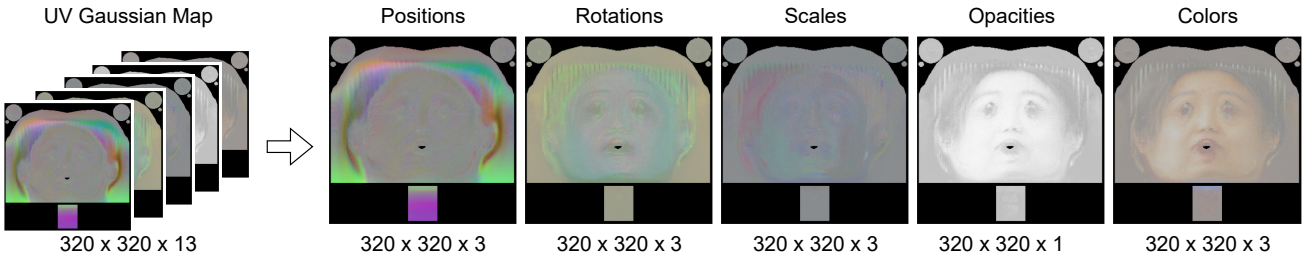

Figure 8. Example UV Gaussian Map. The black area indicates the invalid (unused) pixels.

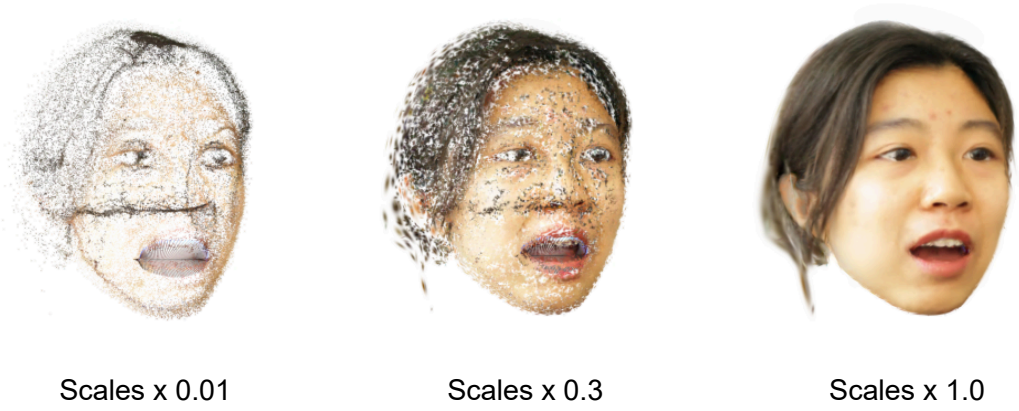

Figure 9. Effects of modifying the scales map.

4. Reconstruction Network Architecture

Our reconstruction network has an encoder-decoder structure. The encoder takes a single image as input and generates a feature map. The decoder takes both of the feature map and the initial UV position map as input and generates the final UV offsets. In each decoder block, we concatenate the feature map with the resized initial UV position map to inject the explicit shape information into the decoder. Figure 10 shows the detailed architecture of our reconstruction network.

5. Training Details

5.1. Single-Image-Based Reconstruction

We use Adam optimizer with an initial learning rate of 1e-4 on synthetic data, and an initial learning rate of 1e-5 on real data. We decrease the learning rate by 5% after every 10,000 training steps. We start the training with a rendering resolution of 128 and gradually increase the resolution after every 100,000 training steps till the resolution is 512×512. When rendering at 128×128, we do not fine-tune on real data. Every time the rendering resolution is increased, we reset the learning rates.

5.2. Monocular Video-Based Optimization

We use Adam optimizer with an initial learning rate of 0.05 for both stages of training. After every 100 training steps, we decrease the learning rate by 10%.

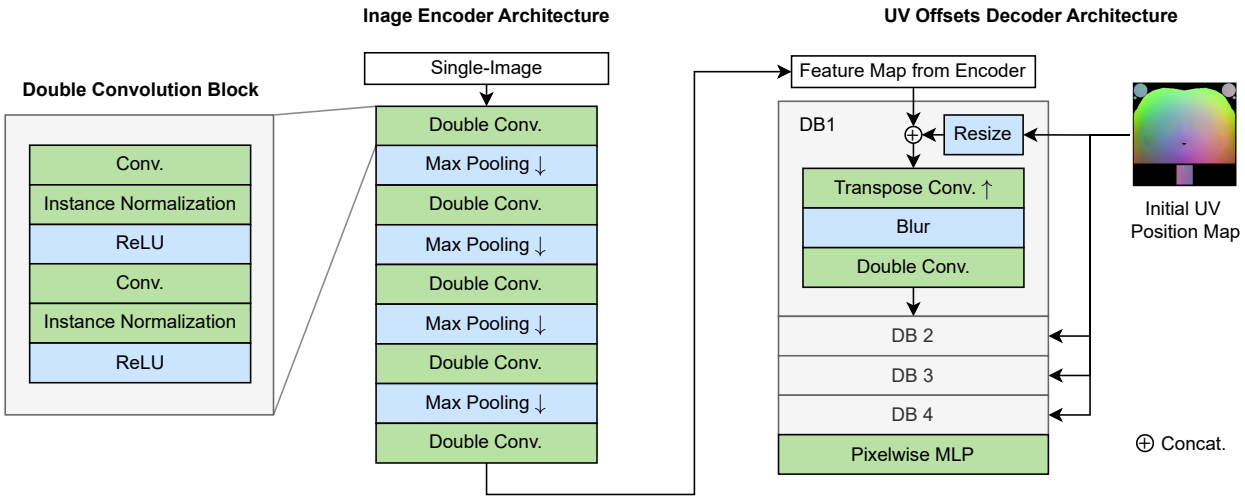

Figure 10. Our single-image-based reconstruction network architecture. DB stands for decoder block.

6. More Experimental Results

6.1. Ablation Study on Training Dataset for Single-Image Reconstruction Model

We conduct an ablation study on the training dataset used for training our single-image-based reconstruction model. Our full method uses both synthetic and real data for training. In this ablation study experiment, we train a reconstruction model solely on the synthetic dataset. Table 1 and Table 2 show the evaluation results on the CelebAMask-HQ dataset and the FFHQ dataset respectively. We can observe a noticeable degradation in the reconstruction quality of real images without the use of real data during training. This outcome highlights a discrepancy between synthetic data and real data, underscoring the essential role of real-image training in achieving balanced and accurate reconstructions.

Method	Region	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Ours (Synthetic Data)	Facial	0.043	0.072	0.922	26.411
Ours (Synthetic + Real Data)		**0.033**	**0.061**	**0.935**	**28.755**
Ours (Synthetic Data)	Head	0.188	0.250	0.692	16.769
Ours (Synthetic + Real Data)		**0.125**	**0.211**	**0.740**	**20.525**

Table 1. Ablation Study on Training Data: Reconstruction performances (CelebAMask-HQ).

Method	Region	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑
Ours (Synthetic Data)	Facial	0.061	0.092	0.896	23.707
Ours (Synthetic + Real Data)		**0.040**	**0.071**	**0.922**	**28.030**
Ours (Synthetic Data)	Head	0.178	0.213	0.747	17.157
Ours (Synthetic + Real Data)		**0.088**	**0.155**	**0.813**	**23.184**

Table 2. Ablation Study on Training Data: Reconstruction performances (FFHQ).

6.2. Ablation Study on Different Initialization Schemes

We test initializing the 3D Gaussians with the FLAME reconstruction's shape (FLAME-initialized). The convergence during training is shown in Figure 11. As illustrated, simply initializing the locations of 3D Gaussians without other Gaussian parameters properly initialized leads to slow convergence speed.

Figure 11. Convergence with different initialization schemes.

6.3. Ablation Study on UV Position Regularization Weights

We use an ablation experiment to demonstrate the importance of our UV position regularization weights mask (mask is shown in Figure 6). In this experiment, we re-trained a reconstruction model with the position regularization weights set to be uniform across all the 3D Gaussians. Figure 12 shows the comparisons between our original model (Best Result) and the re-trained model (Uniform Weight). As shown, if we use uniform position regularization weights, the 3D Gaussians on the scalp are constrained to move too far, and thus the scales of these 3D Gaussians are enlarged to match the hair. However, this will cause the loss of fine details and blurry artifacts. If we manually reduce the scales of each 3D Gaussian, we can observe that by giving more weight to the facial region while less weight to the rest (Best Result), we give non-facial-region 3D Gaussians more freedom to move and form the shape of hair.

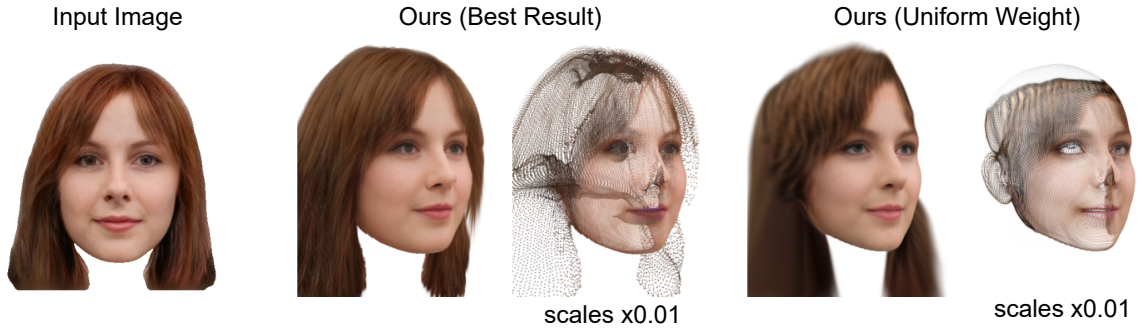

Figure 12. Ablation study on UV position regularization weights. The right images are the renderings of the reconstructed 3D Gaussian head of the input image.

6.4. More Expression Reconstruction Results

We show more expression reconstruction results in Figure 13. The second column displays the reconstructed face using our expression-aware blendmaps for rectification, while the third column illustrates the face with global rectification only.

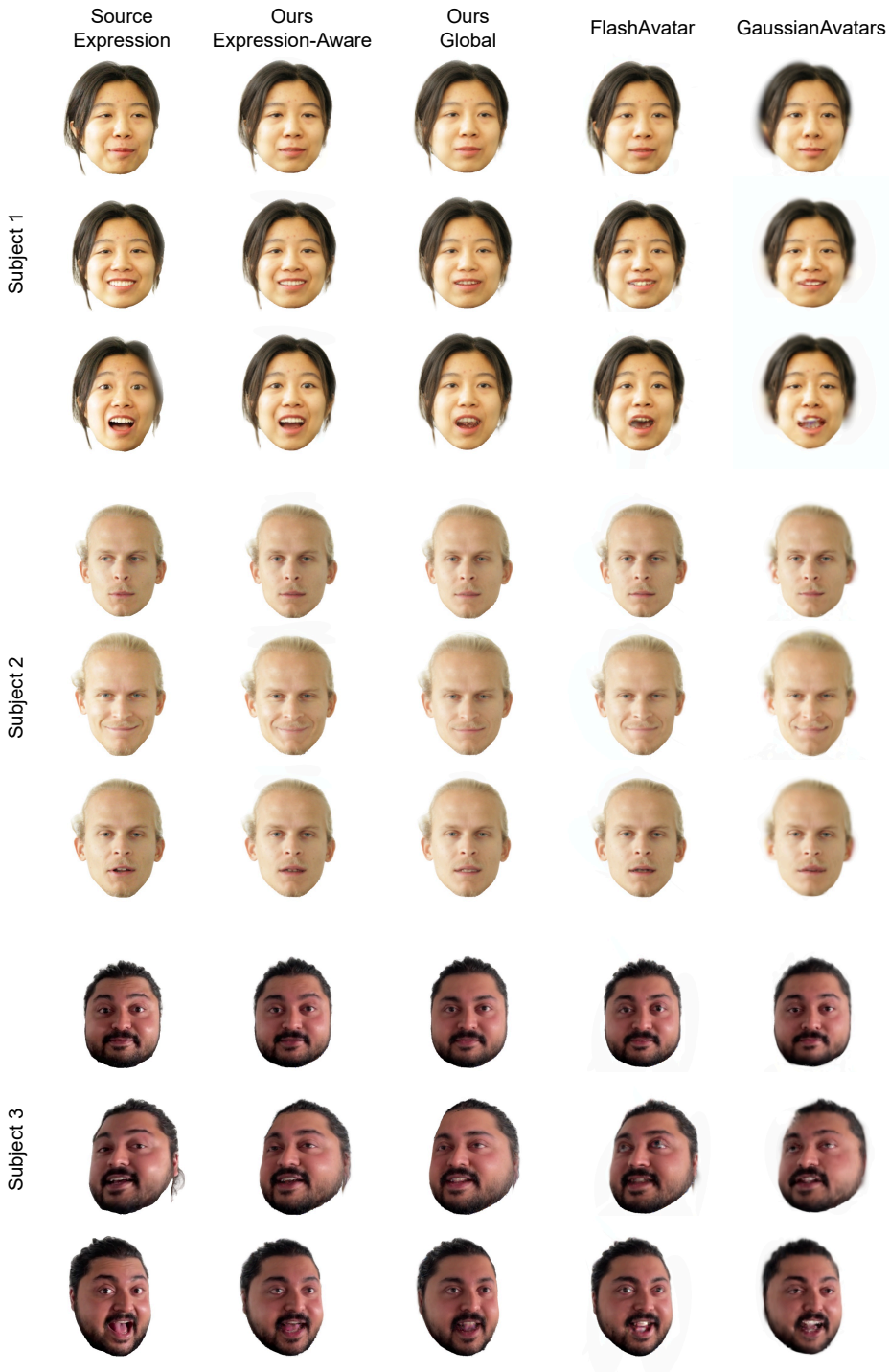

Figure 13. More expression reconstruction results.

7. Limitations

We found that the highest quality achievable by Gaussian Déjà-vu relies significantly on the accuracy of the FLAME reconstruction. Additionally, our current approach does not account for lighting variations, so if the personal video was recorded under changing lighting conditions, the resulting head avatar may exhibit visual artifacts.

References

[1] GitHub - zllrunning/face-parsing.PyTorch: Using modified BiSeNet for face parsing in PyTorch — github.com. https://github.com/zllrunning/face-parsing.PyTorch. [Accessed 08-07-2024]. 2

[2] Sizhe An, Hongyi Xu, Yichun Shi, Guoxian Song, Umit Y Ogras, and Linjie Luo. Panohead: Geometry-aware 3d full-head synthesis in 360deg. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20950–20959, 2023. 1

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[4] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 2

[5] Sefik Serengil and Alper Ozpinar. A benchmark of facial recognition pipelines and co-usability performances of modules. *Journal of Information Technologies*, 17(2):95–107, 2024. 2