Supplementary Material for FENeRF: Face Editing in Neural Radiance Fields

Jingxiang Sun[1*] Xuan Wang[2†] Yong Zhang[2] Xiaoyu Li[2]
Qi Zhang[2] Yebin Liu[3] Jue Wang[2]

[1]University of Illinois at Urbana-Champaign [2]Tencent AI Lab [3]Tsinghua University

1. Implementations

1.1. Mapping Network

The mapping networks in FENeRF are to map the shape and texture latent codes z_s, z_t into the geometry and texture latent spaces, respectively. The output of each mapping network are frequencies γ and phase shifts β for conditioning specific layers of neural radiance fields. We utilize two independent mapping networks: f_{geo} and f_{app} with the similar architectures. Each mapping network is parameterized as an MLP with 4 hidden layers of 256 units followed by LeakyReLU activations [6] each and a linear layer for output γ, β, as shown in Tab. 1. The output dimension of mapping network is $2 \times 256 \times$ n_layer, where n_layer is 8 and 3 for f_{geo} and f_{app}, respectively.

Layer Type	Activation	Output Dimension
Linear	LeakyReLU (0.2)	256
Linear	LeakyReLU (0.2)	256
Linear	LeakyReLU (0.2)	256
Linear	LeakyReLU (0.2)	256
Linear	–	$2 \times 256 \times$ n_layers

Table 1. Mapping network architecture

1.2. Joint Semantic Radiance Fields

Our joint semantic radiance fields are parameterized as MLPs which map a 3D point $\mathbf{x} \in \mathbb{R}^3$ and view direction $\mathbf{d} \in \mathbb{R}^2$ together with shape and texture codes $z_s, z_t \in \mathcal{N}(0, I)$ and learnable coordinate embedding $\mathbf{e}_{coord}$ into volume density $\sigma \in \mathbb{R}^+$, emitted color $\mathbf{c} \in \mathbb{R}^3$ and semantic labels $\mathbf{s} \in \mathbb{R}^k$, where k is set to 19 as the number of semantic classes. As shown in Fig. 1, we use 8 FiLMed-SIREN layers with a hidden dimension of 256 as did in [1] to get the intermediate feature $\mathbf{f}$ conditioning on $\mathbf{z}_s$. Then $\mathbf{f}$ is fed into a single fully connected layer for the volume density σ. Meanwhile, $\mathbf{f}$ is fed into 3 FiLMed-SIREN layers together with $\mathbf{d}, \mathbf{e}_{coord}$ to the color $\mathbf{c}$ conditioning on $\mathbf{z}_t$, and into three fully connected layers for the semantic labels $\mathbf{s}$. Therefore, we ensure that the facial geometry and semantic labels only depend on the input point $\mathbf{x}$ and the shape latent code $\mathbf{z}_s$. Further, the facial texture is conditioned on the texture latent code, coordinate embedding and view-dependent features.

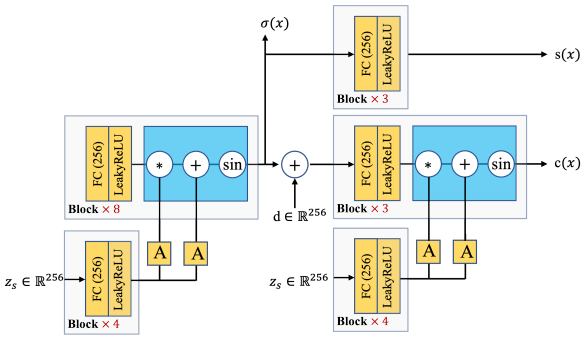


Figure 1. The FENeRF generator architecture

1.3. Discriminators

There are two discriminators in FENeRF: D_c, D_s, whose architectures are the same as that in π-GAN [1] except that the input feature of D_s are the concatenation of image and semantic map instead of single image in D_c.

1.4. Image Background

As mentioned in out main paper, we train FENeRF on the images without background masking and evaluate all metrics on the images with randomly generated backgrounds for fair comparison. Only for visualization, in order to highlight the facial region, we replace the messy BG with pure white/black/grey. Since the black BG mask encourages the densities out of facial region to be as small as possible in training, we replace the BG by setting a density threshold in volume rendering or replacing the generated pixels (with BG mask) with another color.

*Work done during an internship at Tencent AI Lab.
†Corresponding Author.

1.5. Training Details

We begin training FENeRF at a low ersolution of 32 $\times$ 32 with a initial batch size of 40. Then the resolution is doubled up to 128 $\times$ 128 with a batch size of 24 which is aggregated across 4 6-batch together. For both CelebAHQ and FFHQ, we first use 24 samples per ray at 32 $\times$ 32 resolution, and then increase to 48 samples per ray at 64 $\times$ 64 and 128 $\times$ 128 resolution. We sample camera poses from a normal distribution in both datasets. For CelebA, the vertical standard deviation and the horizontal standard deviation are 0.3 and 0.15 radians, respectively. In turns to FFHQ, these two hyper-parameters are increased to 0.4 and 0.2, respectively. We train models used for the image synthesis quality evaluation for 200k iterations on 8 Tesla V100 GPUs. The training process takes about 2 days.

2. GAN Inversion Details

Following the common practice, we try two different initialization ways for GAN inversion, as shown in Fig. 2. The first approach is to randomly sample several (*e.g.* 10k) latent codes and map them into frequencies and phase shifts, whose averages are taken as initialization. This method works well except the case that the initialized shape or texture is starkly different from the reference one.

We further adopt a hybrid inversion approach where we first train two encoders which map real images into the frequencies and phase shifts of shape and texture latent spaces independently: $E_s : (\mathbf{I}, \mathbf{T}) \rightarrow (\gamma_s, \beta_s)$ and $E_t : (\mathbf{I}, \mathbf{T}) \rightarrow (\gamma_t, \beta_t)$. Here $\mathbf{I}$ and $\mathbf{T}$ are input real images and camera poses. We select real images $\mathbf{I}$ from the CelebAHQ [3]. For camera poses, recall that we append two channels in our discriminators to predict the sampled camera pose which is trained with L1 loss. Therefore, we use our pretrained discriminator to predict the input camera pose $\mathbf{T}$. Following [5], each encoder adopts a feature pyramid network [4] as backbone to extract multi-level features from a input image. Then the features are projected to frequencies and phase shifts through fully convolutional layers. During training E_t and E_s, parameters of the FENeRF generator are fixed. During inversion, we feed the reference image into two encoders and use the mapped frequencies and phase shifts as initialization. We find that the second method gains better image quality due to closer initialization to the distribution of the latent spaces.

Followed by initialization, we optimize shape and texture latent codes jointly given the reference image. We adopt the Perceptual Loss [2] with the weight of 1 and Mean Squared Error Loss with the weight of 0.5.

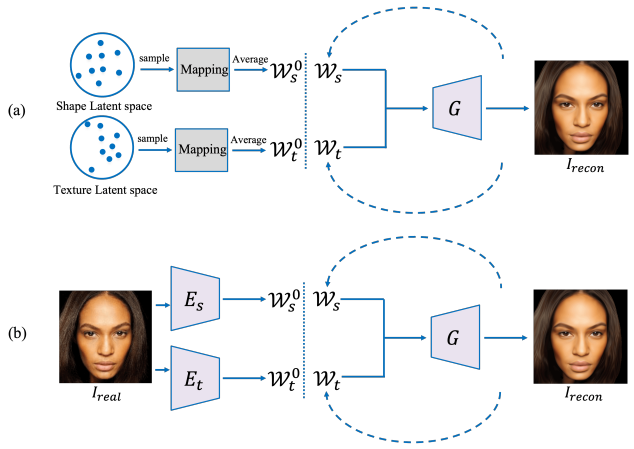


Figure 2. Two GAN Inversion approaches of FENeRF. (a): We randomly sample $\mathbf{z}_s, \mathbf{z_t}$ in two latent spaces and project them into γ, β in $\mathcal{W}_s, \mathcal{W}_t$. Then the average of γ, β are taken as initialization; (b): we directly project the real image into $\mathcal{W}$ space for initialization through encoders.

3. Additional Experimental Results

3.1. Ablation Studies

Effects of learnable coordinate embedding. As mentioned in our paper, the introduced learnable coordinate embedding $\mathbf{e}_{coord}$ facilitates the fine-grained image details. In practice, we find that $\mathbf{e}_{coord}$ also improves the disentanglement of shape and texture, as shown in Fig. 3. In the pure SIREN-based frameworks, the color branch tends to ignore texture code $\mathbf{z}_t$ since the texture is coupled with shape in the intermediate feature $\mathbf{f}$ and the color branch is less likely to response to texture code $\mathbf{z}_t$ while overfits to the input $\mathbf{f}$ towards faster training converge. By contrast, the injected learnable embeddings $\mathbf{e}_{coord}$ increases the hardness of the network overfitting to $\mathbf{f}$ thus enforces the color branch to condition on the injected texture code $\mathbf{z}_t$. Moreover, we also find that shrinking the number of feature channels of $\mathbf{f}$ and increasing the hidden layers of the color branch also benefit to the disentanglement of shape and texture.

4. Additional Visual Results

We show additional visual results of the qualitative comparison with other baseline models and fancy applications of our method. Fig. 4 shows the comparison of geometry interpretation with π-GAN to prove that our FENeRF learns more accurate generative facial geometry and the background segmentation without supervision. Fig. 5 shows the inversion ability of our semantic rendering. Fig. 6 shows more results of our disentangled control of shape and texture. We provide more visual examples of facial local editing in Fig. 7, proving that FENeRF enables fine-grained facial editing in a free-viewed manner. Please refer to our

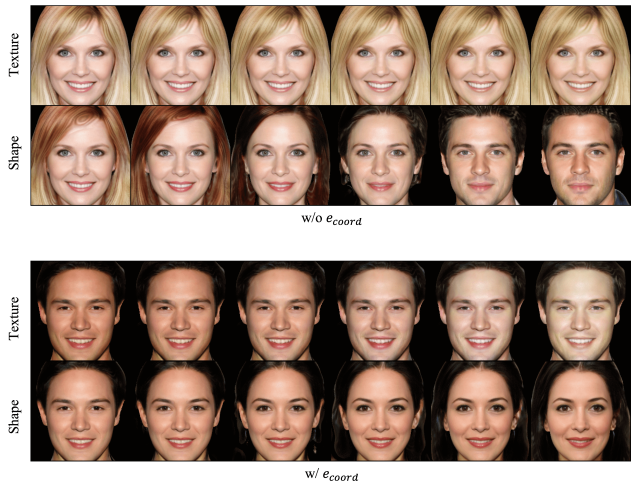


Figure 3. Effect of learnable feature embedding $\mathbf{e}_{coord}$ on the disentanglement of shape and texture. The first two rows show the disentanglement of shape and texture w/o $\mathbf{e}_{coord}$ and the bottom two rows show that w/ $\mathbf{e}_{coord}$. Pure SIREN-based framework couples both shape and texture styles in $\mathbf{z}_s$ while $\mathbf{z}_t$ has no effect on the facial texture. Instead, the injection of our learnable feature embedding improves the disentanglement of shape and texture significantly.

supplemental demo video for more dynamic visual results.

References

[1] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 1

[2] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[3] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020. 2

[4] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[5] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021. 2

[6] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015. 1

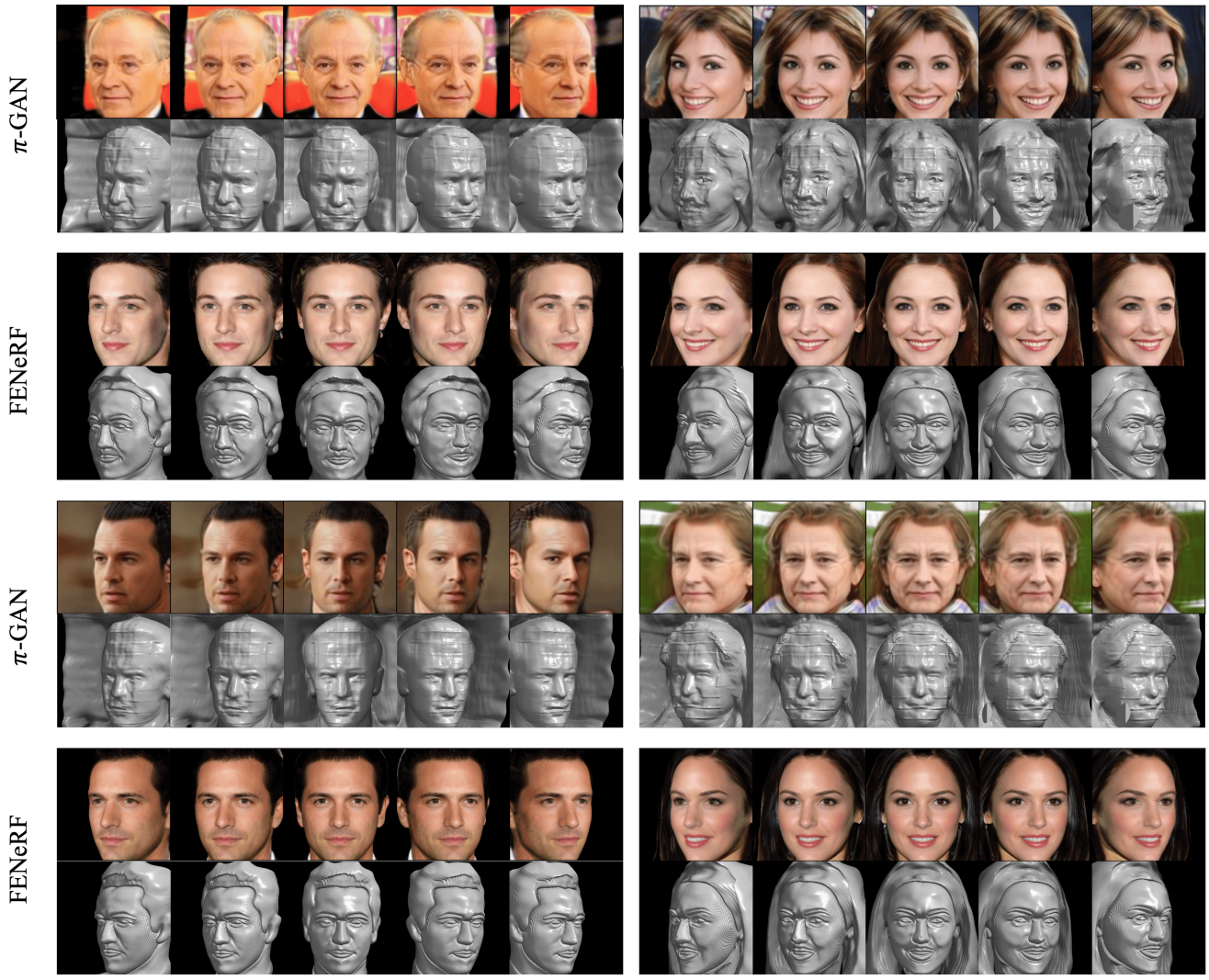
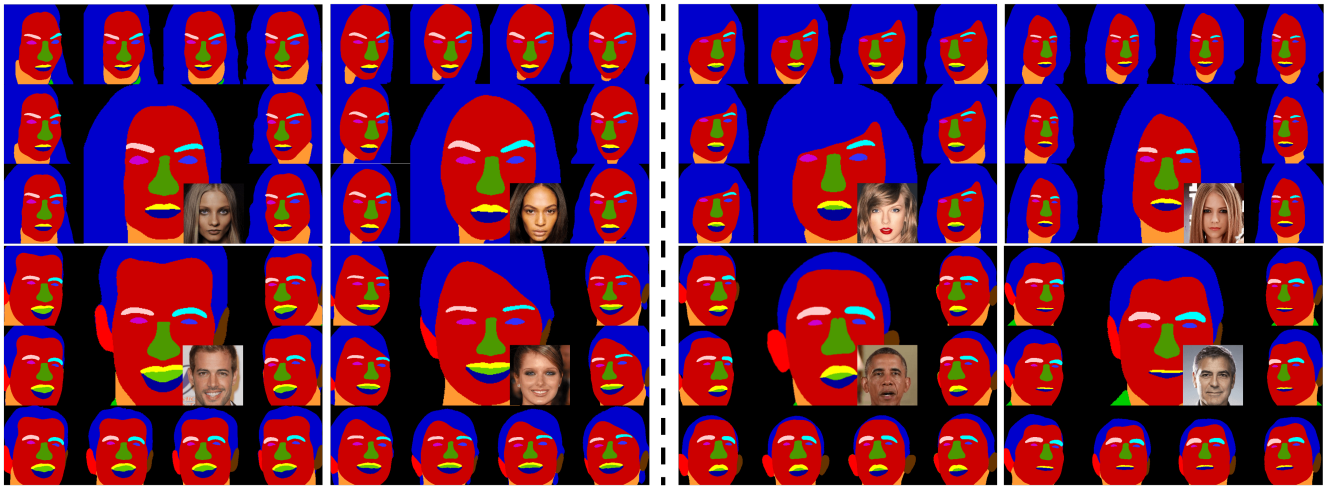
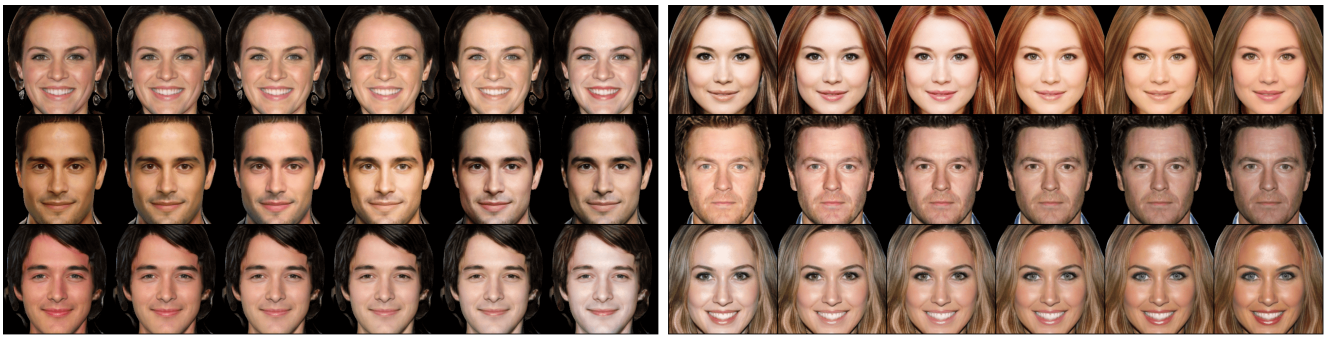


Figure 4. Visual comparison of geometry interpretation with π-GAN.



CelebAHQ Portraits in the Wild

Figure 5. Free-viewed semantic inversion of real images. Given a real portrait image, FENeRF is capable of projecting it into the shape latent space with its reconstructed semantic field. The semantic field can be rendered into free-viewed semantic maps with view consistency.



Interpolation on Texture

Interpolation on Shape

Figure 6. Results of disentangled control of shape and texture.

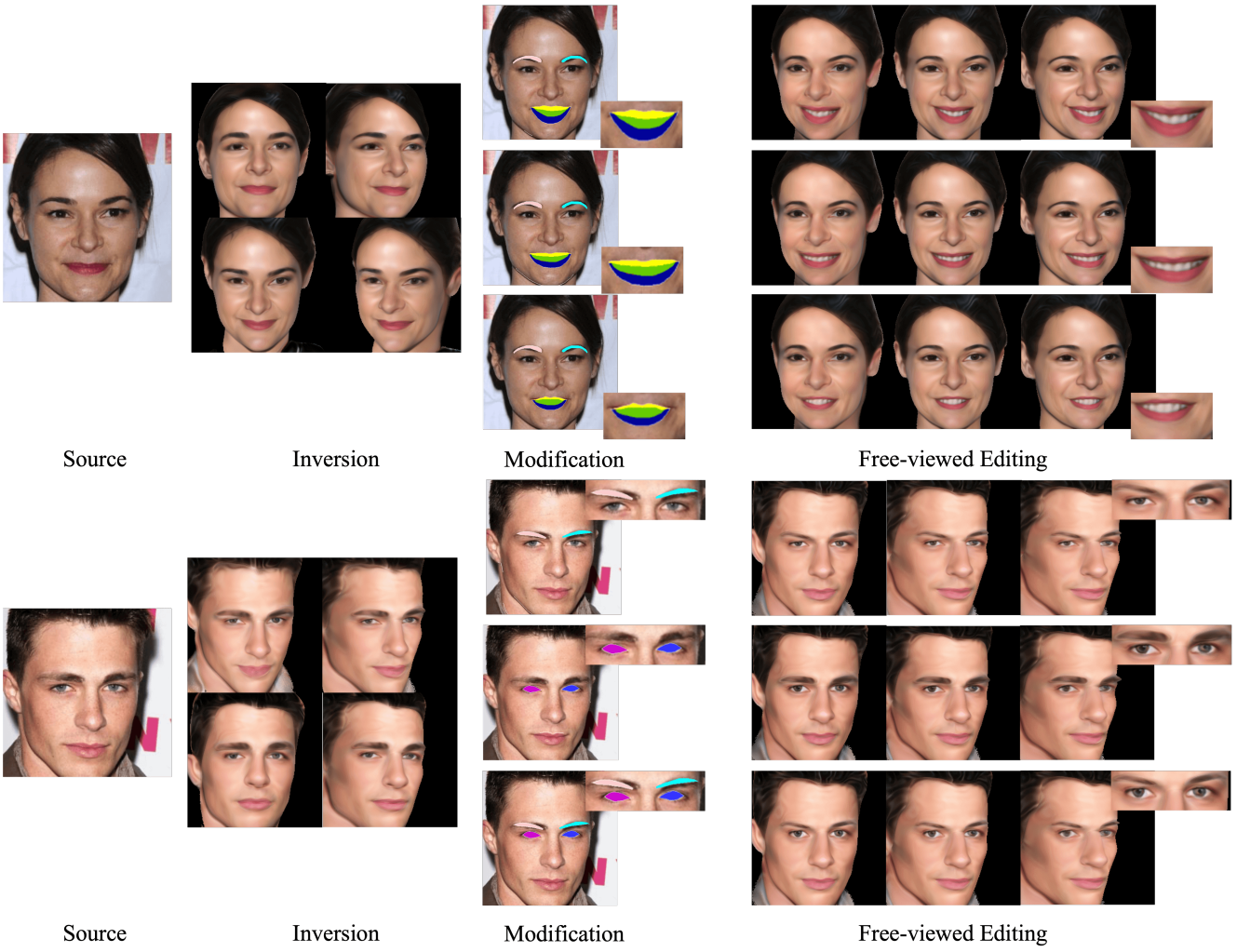


Figure 7. Results of local facial editing. FENeRF achieves fine-grained local facial editing, showing that the latent spaces are region disentangled.