Editing in Style: Uncovering the Local Semantics of GANs
CVPR 2020 supplementary material

Edo Collins[1] Raja Bala[2] Bob Price[2] Sabine Süsstrunk[1]
[1]School of Computer and Communication Sciences, EPFL, Switzerland
[2]Interactive and Analytics Lab, Palo Alto Research Center, Palo Alto, CA

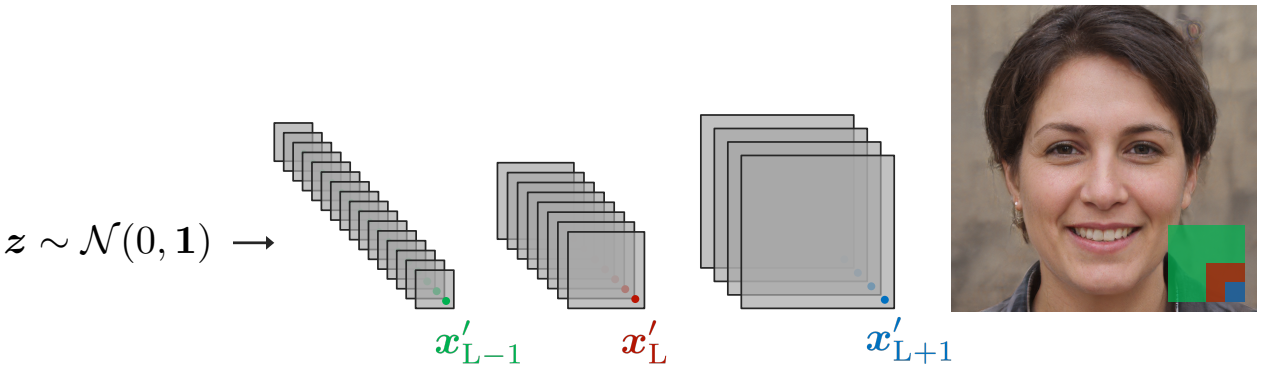

Figure 8: In convolutional generative networks, the vector x' at a single spatial position on a hidden feature map at some layer L corresponds to a whole *patch* on the RGB image.

1. Spherical k-means for semantic clustering

In this section we elaborate on the layer-wise analysis described in Section 3 of the paper.

For a hidden layer L with C channels, let $\mathbf{A} \in \mathbb{R}^{(N\times C\times H\times W)}$ be a tensor of instance-normalized activations of N images, where at each channel the feature map has spatial dimensions $H \times W$.

As show in Fig. 8, a vector $\boldsymbol{a} \in \mathbb{R}^C$ sampled at a single spatial location on $\mathbf{A}$ represents a whole patch (e.g., 32×32) in the RGB image, and acts as a *patch embedding*.

We apply spherical k-means to this C-dimensional space by first partially flattening $\mathbf{A}$ to $\boldsymbol{A} \in \mathbb{R}^{(N\cdot H\cdot W)\times C}$, i.e., to a "bag-of-patch-embeddings" representation, with no explicit encoding of spatial position or even partitioning into different samples. The process can thus be viewed as clustering patches whose embeddings at layer L are similar, in the cosine similarity sense.

Performing spherical k-means with K clusters can be viewed as a matrix factorization $\boldsymbol{A} \approx \boldsymbol{UV}$, where the binary matrix $\boldsymbol{U} \in \{0, 1\}^{(N\cdot H\cdot W)\times K}$ encodes cluster membership and the matrix $\boldsymbol{V} \in \mathbb{R}^{K\times C}$ encodes the centroid of each cluster.

The matrix $\boldsymbol{U}$ can be reshaped to a tensor $\mathbf{U} \in \{0, 1\}^{N\times K\times H\times W}$ which represents K sets of N *masks* (one per image), where each mask spatially shows the cluster memberships.

In Figs. 9, 10 we show examples produced with StyleGAN (Karras et al., 2019), where the tensor $\mathbf{U}$ is upsampled and overlaid on RGB images. The color-coding in these figures indicates to which cluster a spatial position belongs. In Figs. 11, 12 we show similar results for ProgGAN (Karras et al., 2018).

The main observation of this analysis is that at certain layers (e.g., layer 6 of StyleGAN), activations capture abstract semantic concepts (e.g., *eyes* for faces, *pillow* for bedrooms).

By manually examining the cluster membership masks of a few (five to ten) samples, an annotator can easily label a cluster as representing a certain object. Thus, we randomly generated 200 samples and recorded all their activations. We tested several layers and K combinations and selected the one that qualitatively yielded the most semantic decomposition into objects, as shown in Figures 9 and 10. We then manually labeled the resulting clusters. In the case that multiple clusters matched a part of interest, we merged their masks into a single mask. Note that this process is a one-time offline process (per dataset/GAN) that then drives a fully automated semantic editing operation.

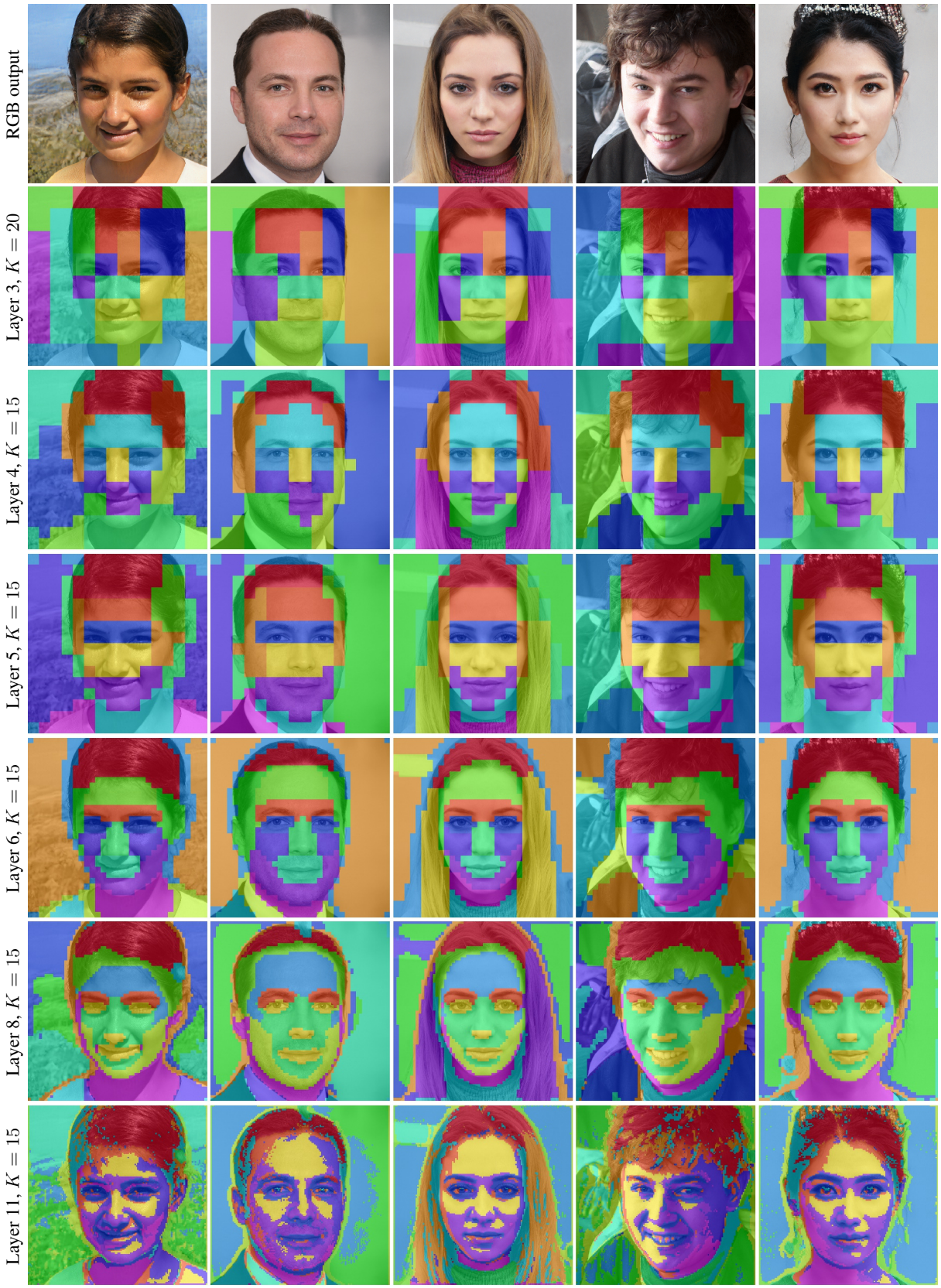

Figure 9: Spherical k-means cluster membership maps for various FFHQ-StyleGAN layers. Color-coding signifies different clusters, and is arbitrarily determined per layer.

Figure 10: Spherical k-means cluster membership maps for various LSUN-Bedroom-StyleGAN layer. Color-coding signifies different clusters, and is arbitrarily determined per layer.

Figure 11: Spherical k-means cluster membership maps for various CelebA-HQ-ProgGAN layer. Color-coding signifies different clusters, and is arbitrarily determined per layer.

Figure 12: Spherical k-means cluster membership maps for various LSUN-Bedroom-ProgGAN layer. Color-coding signifies different clusters, and is arbitrarily determined per layer.

2. Squared-error maps

Squared-error "diff" maps between edited outputs and the target image help detect changes between the two images and evaluate the locality of the edit operation. We compute the error in CIELAB color-space.

In Figs. 13 and 14 we show the diff maps corresponding to Figs. 3 and 4 respectively.

3. Additional qualitative results with StyleGAN2

In this section we show additional results with StyleGAN2. Figs. 15 and 17 are extended versions of Fig. 6. Figs. 16 and 18 show their diff maps. Figs. 19 and 20 show results for StyleGAN2 trained of FFHQ.

Figure 13: Mean-squared error maps between the edited outputs shown in Fig. 3 and the target image, shown in the same figure. Editing is primarily focused on the object of interest, though some subtle changes do occur elsewhere in the scene.

Figure 14: Mean-squared error maps corresponding to Fig. 3. Correlations learned and respected by the GAN sometimes lead to unintentional changes, e.g., changes to the picture on the wall when editing the bed (first row).

Figure 15: Our local editing method applied to StyleGAN2 trained on LSUN-Cats.

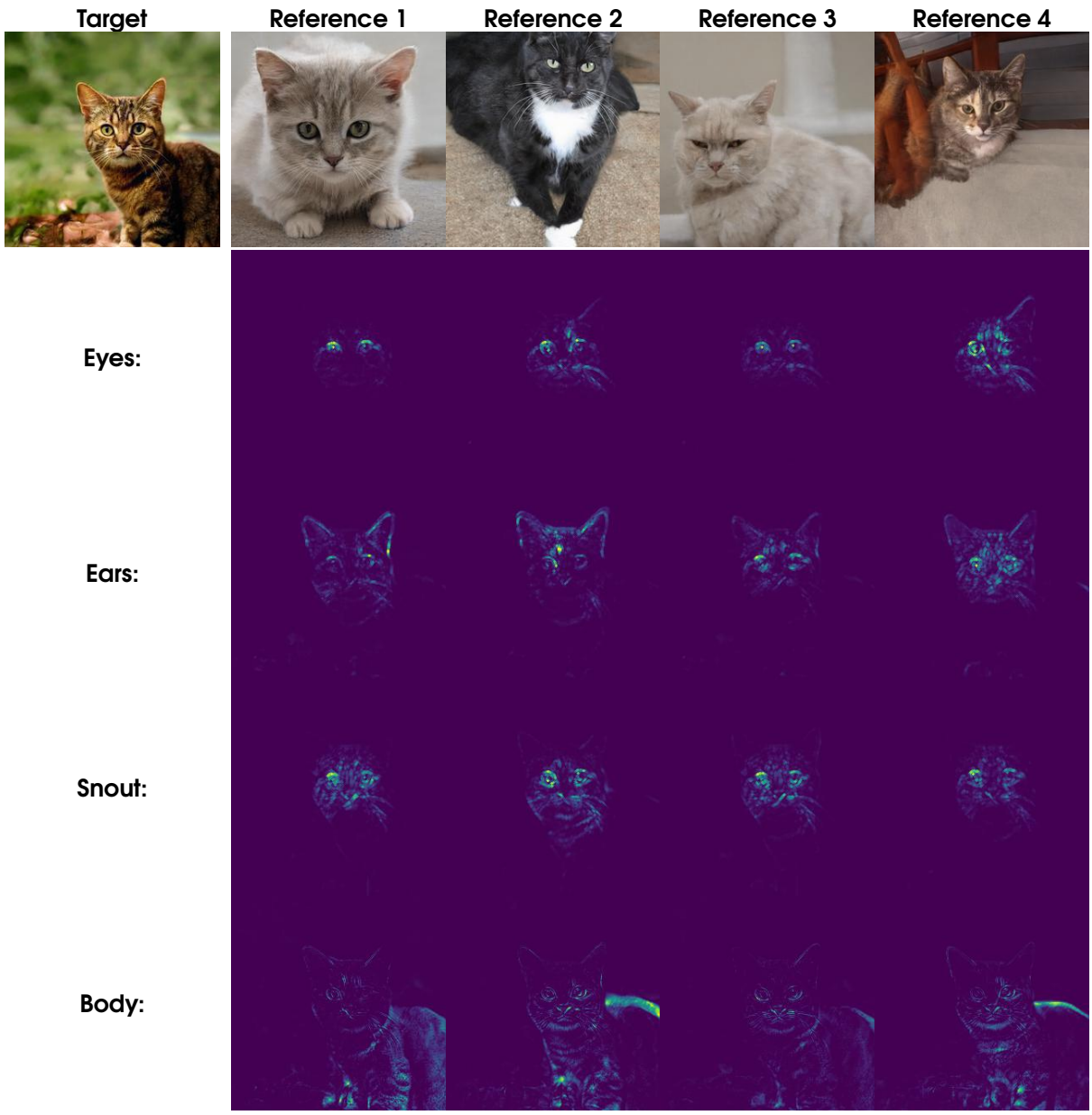

Figure 16: Diff maps corresponding to the results in Fig. 15, for StyleGAN2 trained on LSUN-Cats.

Figure 17: Our local editing method applied to StyleGAN2 trained on LSUN-Cars.

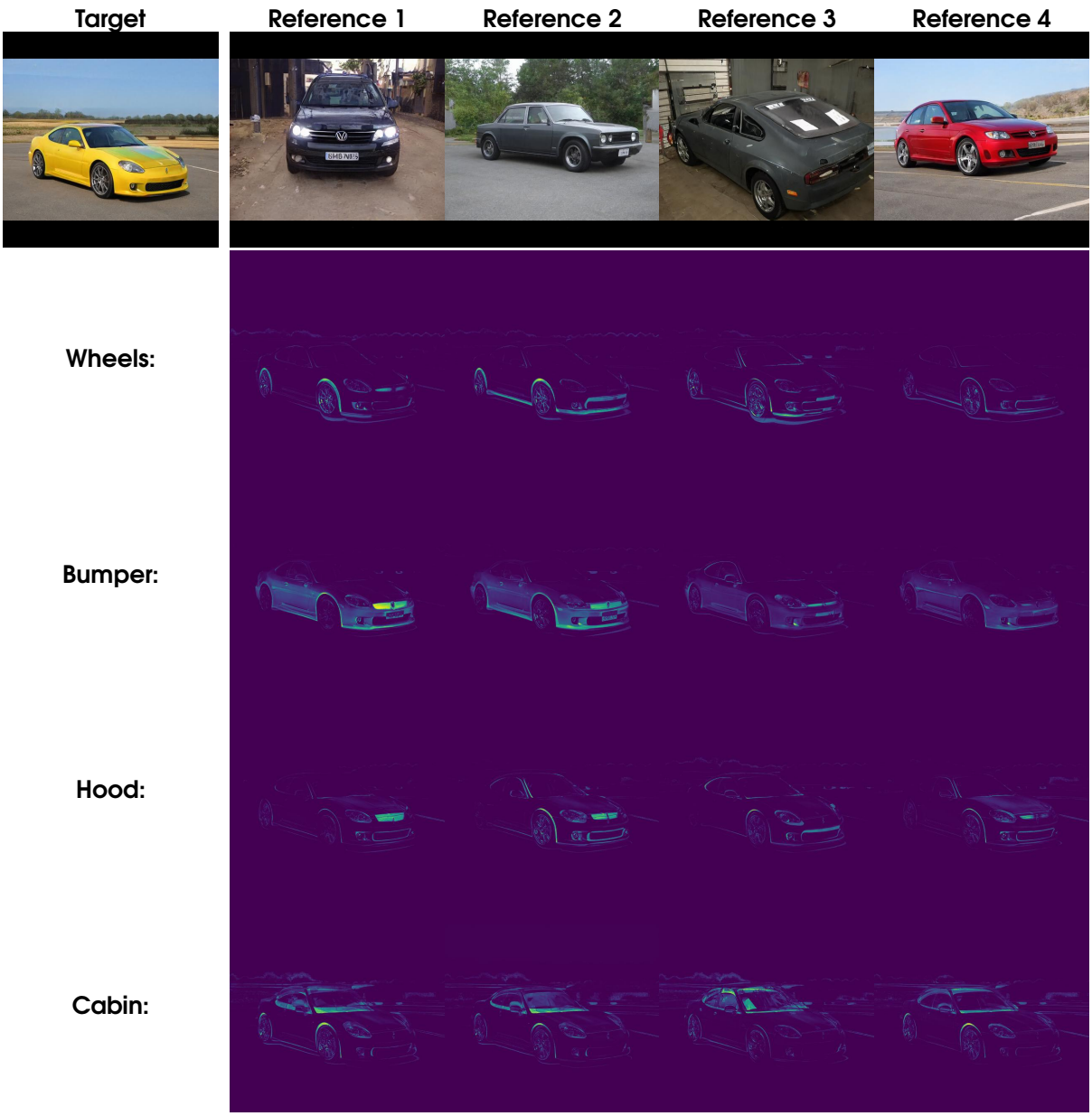

Figure 18: Diff maps corresponding to the results in Fig. 17, for StyleGAN2 trained on LSUN-Cars.

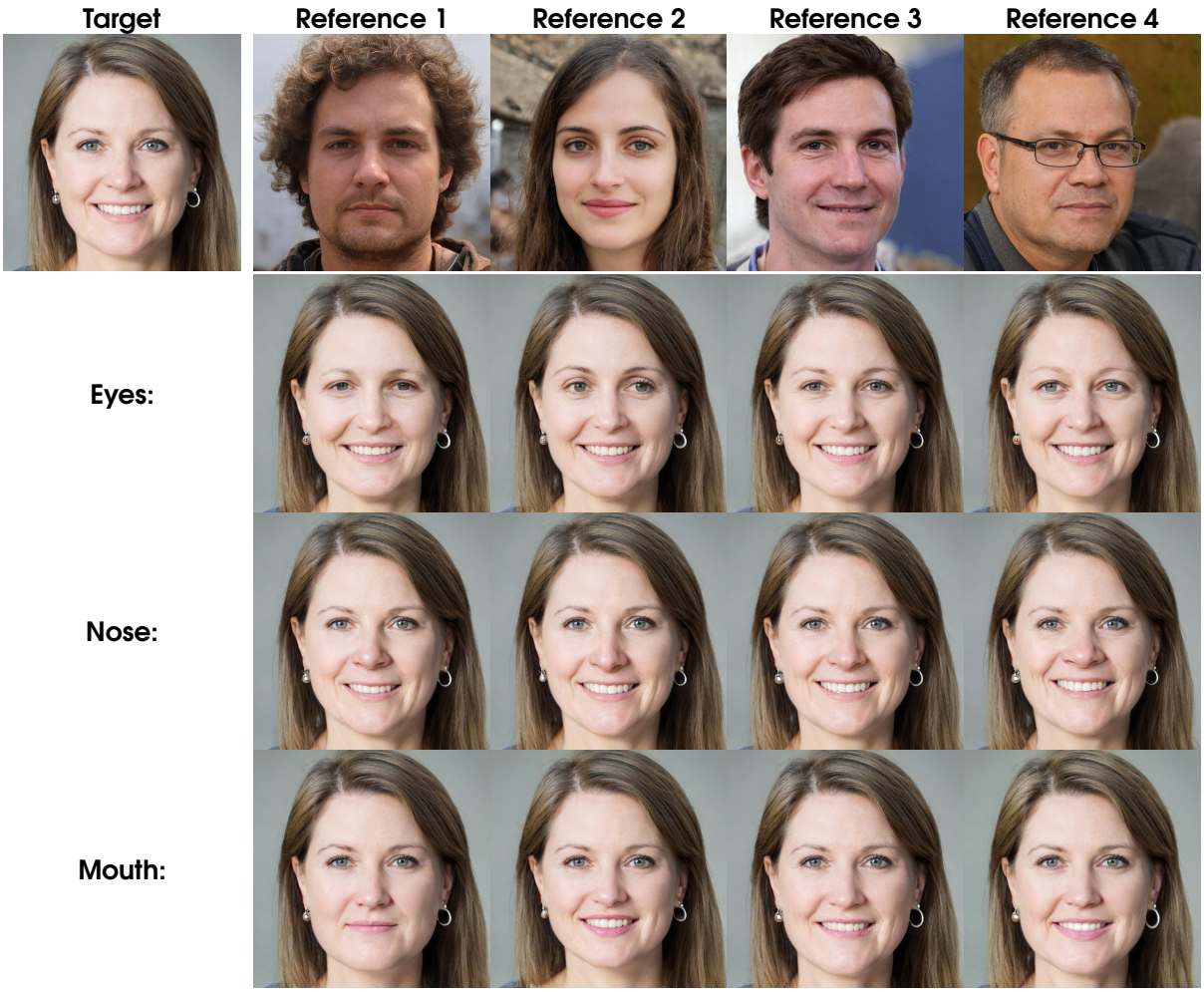

Figure 19: Our local editing method applied to StyleGAN2 trained on FFHQ.

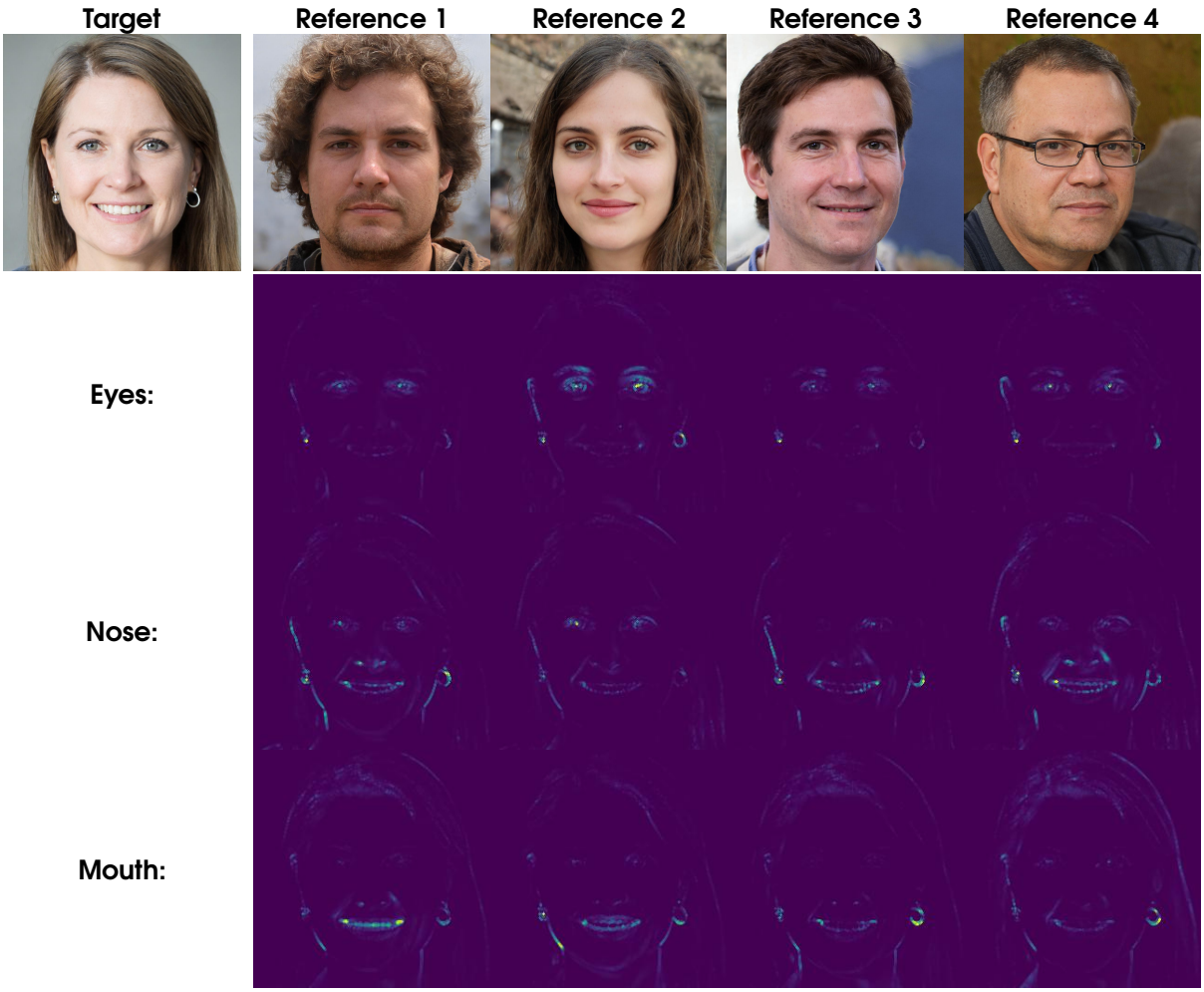

Figure 20: Diff maps corresponding to the results in Fig. 19, for StyleGAN2 trained on FFHQ.