

Editing in Style: Uncovering the Local Semantics of GANs

Edo Collins¹ Raja Bala² Bob Price² Sabine Ssstrunk¹

¹School of Computer and Communication Sciences, EPFL, Switzerland

²Interactive and Analytics Lab, Palo Alto Research Center, Palo Alto, CA

{edo.collins, sabine.susstrunk}@epfl.ch {rbala, bprice}@parc.com

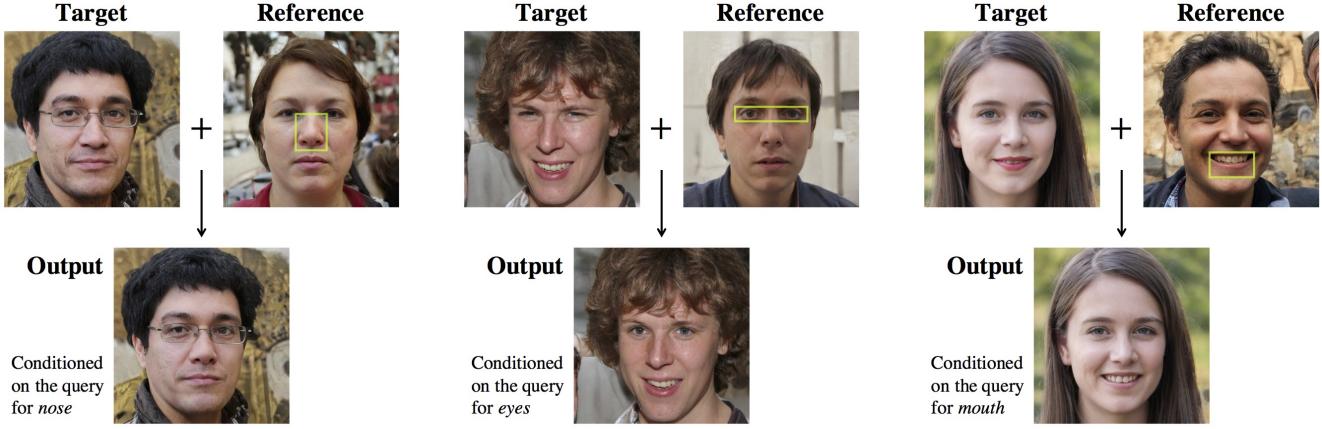


Figure 1: Our method performs local semantic editing on GAN output images, transferring the appearance of a specific object part from a reference image to a target image.

Abstract

While the quality of GAN image synthesis has improved tremendously in recent years, our ability to control and condition the output is still limited. Focusing on StyleGAN, we introduce a simple and effective method for making local, semantically-aware edits to a target output image. This is accomplished by borrowing elements from a source image, also a GAN output, via a novel manipulation of style vectors. Our method requires neither supervision from an external model, nor involves complex spatial morphing operations. Instead, it relies on the emergent disentanglement of semantic objects that is learned by StyleGAN during its training. Semantic editing is demonstrated on GANs producing human faces, indoor scenes, cats, and cars. We measure the locality and photorealism of the edits produced by our method, and find that it accomplishes both.

1. Introduction

In the short span of five years, generative adversarial neural networks (GANs) have come to dominate the field

of data-driven image synthesis. Like most other neural network models, however, the exact model they learn for the data is not straightforwardly interpretable.

There have been significant steps towards alleviating this issue. For instance, state-of-the-art image GANs such as PG-GAN [15] and StyleGAN [16], by virtue of their progressive training, encourage each layer to model the variation exhibited at given image resolutions (e.g., 8×8 images capture coarse structure, 32×32 add finer details, etc.).

The notion of a *disentangled* representation has been used to describe such phenomena. While definitions of disentanglement are many and varied [12], the common idea is that an attribute of interest, which we often consider *semantic*, can be manipulated independently of other attributes.

In this paper we show that deep generative models like PG-GAN, StyleGAN and the recent StyleGAN2 [17] learn a representation of objects and object-parts that is disentangled in the sense that various semantic parts (e.g., the *mouth* of a person or the *pillows* in a bedroom) have a significant ability to vary independently of the rest of the scene.

Based on this observation we propose an algorithm that performs *spatially-localized semantic editing* on the outputs

Important

Nice!

of GANs - primarily StyleGAN. Editing is performed by transferring semantically localized *style* from a reference image to a target image, both outputs of a GAN. Our method is simple and effective, requiring no more than an off-the-shelf pre-trained GAN. Our method is unique in that it enacts a localized change through a global operation, akin to style transfer. As a result, unlike other GAN editing methods that make use of additional datasets and trained networks, or traditional image morphing methods requiring complex spatial operations, our method relies upon and benefits solely from the rich semantic representation learned by the GAN itself. Applications include forensic art where a human face is composed from various sources; and interior design where various combinations of design elements such as furniture, upholstery, etc., can be visualized. Extension to semantic editing of real images can be envisioned by combining our approach with the recent work that embeds natural images into the latent space of StyleGAN [1, 17].

We make the following contributions:

- We provide insight into the structure of hidden activations of the StyleGAN generator, showing that the learned representations are largely disentangled with respect to semantic objects in the synthesized image.
- We exploit this structure to develop a novel image editor that performs semantic part transfer from a reference to a target synthesized image. The underlying formulation is simple and elegant and achieves naturalistic part transfer without the need for complex spatial processing, or supervision from additional training data and models.

The paper is structured as follows. In Section 2 we review work related to GAN editing and interpretability. In Section 3 we detail our observations regarding spatial disentanglement in GAN latent space and introduce our local editing method. In Section 4 we show experimental results that validate our claims, and in Section 5 we conclude with a discussion of the results and future work.

2. Related Work

The literature on the use of GANs for image synthesis has exploded since the seminal work by Goodfellow et al. [9], with today’s state of art methods such as StyleGAN [16], StyleGAN2[17], and BigGAN [4] producing extremely realistic outputs. For a thorough review of the GAN literature we refer the reader to recent surveys in [7, 13, 27]. Our goal here is *not* to propose another GAN, but to offer a local editing method for its output, by changing the style of specific objects or object parts to the style given in a reference image. We next review past work germane to semantic image editing, paying particular attention to recent GAN-based methods.

2.1. GAN-based Image Editing

Several works have explored the use of deep generative models for semantic image editing. We distinguish between two flavors: latent code-based methods for global attribute editing and activation-based methods for local editing.

Latent code-based techniques learn a manifold for natural images in the latent code space facilitated by a GAN and perform semantic edits by traversing paths along this manifold [23, 32]. A variant of this framework employs auto-encoders to disentangle the image into semantic subspaces and reconstruct the image, thus facilitating semantic edits along the individual subspaces [2, 25]. Examples of edits accomplished by these techniques include global changes in color, lighting, pose, facial expression, gender, age, hair appearance, eyewear and headwear [2, 19, 25, 30, 31]. AttGAN [10] uses supervised learning with external attribute classifiers to accomplish attribute editing.

Activation-based techniques for local editing directly manipulate specific spatial positions on the activation tensor at certain convolutional layers of the generator. In this way, GAN Dissection [3] controls the presence or absence of objects at given positions, guided by supervision from an independent semantic segmentation model. Similarly, feature blending [26] transfers objects between a target GAN output and a reference by “copy-pasting” activation values from the reference onto the target. We compare that technique, together with traditional Poisson blending [24], to our approach in Fig. 5.

Distinct from all these works, our approach is a latent code-based approach for local editing. Crucially, it neither relies on external supervision by image segmentation models nor involves complex spatial blending operations. Instead, we uncover and exploit the disentangled structure in the embedding space of the generator that naturally permits spatially localized part editing.

2.2. Face Swapping

Our technique for object-specific editing, when applied to face images, is akin to the problems of face swapping and transfer. Previous efforts [18, 21, 22] describe methods for exchanging global properties between a pair of facial images. Our method stands out from these approaches by offering editing that is localized to semantic object parts. Furthermore, a primary motivation for face swapping is de-identification for privacy preservation, which is not relevant for our goal of editing synthetic images. Yang et al. [28] present a method for transferring expression from one face to another. Certain specific cases of expression transfer (e.g., smile) involve localized part (e.g., mouth) transfer, and are thus similar to our setting. However, even in these common scenarios, our editing framework is unique in that it requires no explicit spatial processing such as warping and compositing.

Activation-based shit

3. Local Semantics in Generative Models

3.1. Feature factorization

Unclear:
Read
about DFF
to
understand
this fully.

Maybe has
to do with
activation
stuff?

Deep feature factorization (DFF) [6] is a recent method that explains a convolutional neural network’s (CNN) learned representation through a set of saliency maps, extracted by factorizing a matrix of hidden layer activations. With such a factorization, it has been shown that CNNs trained for ImageNet classification learn features that act as semantic object and object-part detectors.

Inspired by this finding, we conducted a similar analysis of the activations of generative models such as PG-GAN, StyleGAN, and StyleGAN2. Specifically, we applied spherical k-means clustering [5] to the C -dimensional activation vectors that make up the activation tensor $\mathbf{A} \in \mathbb{R}^{N \times C \times H \times W}$ at a given layer of the generator, where N is the number of images, C is the number of channels, and H, W are spatial dimensions. The clustering generates a tensor of cluster memberships, $\mathbf{U} \in \{0, 1\}^{N \times K \times H \times W}$, where K is user-defined and each K -dimensional vector is a one-hot vector which indicates to which of K clusters a certain spatial location in the activation tensor belongs.

The main result of this analysis is that at certain layers of the generator, clusters correspond well to semantic objects and parts. Fig. 2 shows the clusters produced for a 32×32 layer of StyleGAN generator networks trained on Flickr-Faces-HQ (FFHQ) [16] and LSUN-Bedrooms [29]. Each pixel in the heatmap is color-coded to indicate its cluster. As can be seen, clusters spatially span coherent semantic objects and object-parts, such as *eyes*, *nose* and *mouth* for faces, and *bed*, *pillows* and *windows* for bedrooms.

The cluster membership encoded in \mathbf{U} allows us to compute the contribution $M_{k,c}$ of channel c towards each semantic cluster k as follows:

$$M_{k,c} = \frac{1}{NHW} \sum_{n,h,w} \mathbf{A}_{n,c,h,w}^2 \odot \mathbf{U}_{n,k,h,w}. \quad (1)$$

Assuming that the feature maps of \mathbf{A}_l have zero mean and unit variance, the contribution of each channel is bound between 0 and 1, i.e., $M \in [0, 1]^{K \times C}$.

Furthermore, by bilinearly up- or down-sampling the spatial dimensions of the tensor \mathbf{U} to an appropriate size, we are able to find a matrix M for all layers in the generator, with respect to the same semantic clusters.

Using this approach we produced a semantic catalog for each GAN. We chose at which layer and with which K to apply spherical k-means guided by a qualitative evaluation of the cluster membership maps. This process requires only minutes of human supervision.

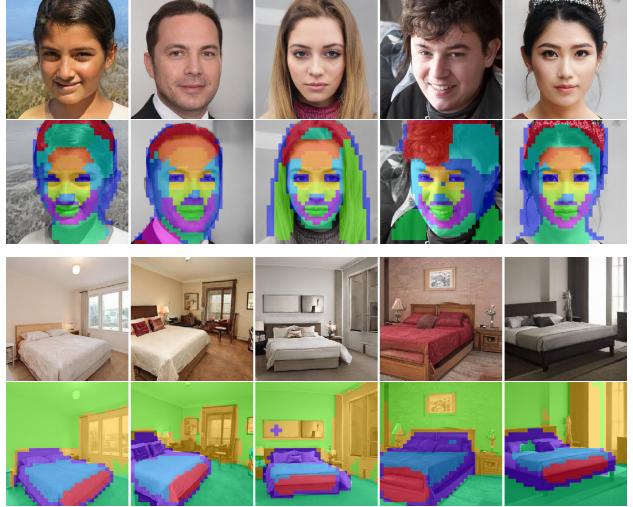


Figure 2: Applying k-means to the hidden layer activations of the StyleGAN generator reveals a decomposition of the generated output into semantic objects and object-parts.

3.2. Local editing

3.2.1 StyleGAN overview

We briefly review aspects of StyleGAN and StyleGAN2 relevant to our development. First, in order to generate a sample from the generator distribution, a latent vector \mathbf{z} is randomly sampled from the prior of the sampling space \mathbb{Z} . Next, \mathbf{z} is transformed to an intermediate latent vector $\mathbf{w} \in \mathbb{W}$, that has been shown to exhibit better disentanglement properties [16], [30].

The image generator is implemented as a convolutional neural network. Considering a batch consisting of a single image, let $\mathbf{A} \in \mathbb{R}^{(C \times H \times W)}$ be the input to a convolutional layer, which is assumed or explicitly normalized to have per-channel unit variance. Prior to the convolution operation, the vector \mathbf{w} alters the feature maps via a per-layer *style*. Common to the application of style in both StyleGAN and StyleGAN2 is the use of per-channel *scaling*, $\sigma_c \mathbf{A}_c$, where the layer-wise coefficients σ are obtained from a learned affine transformation of \mathbf{w} .

This style-based control mechanism is motivated by *style transfer* [8], [20], where it has been shown that manipulating per-channel mean and variance is sufficient to control the style of an image [14]. By fixing the input to the StyleGAN convolutional generator to be a constant image, the authors of StyleGAN showed that this mechanism is sufficient to determine all aspects of the generated image: the style at one layer determines the content at the next layer.



Figure 3: Our method localizes the edit made to the target image (top left) by conditioning the style transfer from the reference (top row) on a specific object of interest (left column). This gives users fine control over the appearance of objects in the synthesized images. Best viewed enlarged on screen.

3.2.2 Conditioned interpolation

Given a target image S and a reference image R , both GAN outputs, we would like to transfer the appearance of a specified local object or part from R to S , creating the edited image G . Let σ^S and σ^R be two style scaling coefficients of the same layer corresponding to the two images.

For global transfer, due to the properties of linearity and separability exhibited by StyleGAN’s latent space, a mixed style σ^G produced by linear interpolation between σ^S and σ^{R1} ¹ produces plausible fluid morphings between the two images:

$$\sigma^G = \sigma^S + \lambda(\sigma^R - \sigma^S) \quad (2)$$

for $0 \leq \lambda \leq 1$. Doing so results in transferring *all* the

¹Karras et al. (2019) [16] interpolate in the latent space of w , but the effect is similar.

properties of σ^R onto σ^G , eventually leaving no trace of σ^S .

To enable selective local editing, we control the style interpolation with a matrix transformation:

$$\sigma^G = \sigma^S + Q(\sigma^R - \sigma^S) \quad (3)$$

where the matrix Q is positive semi-definite and is chosen such that σ^G effects a local style transfer from σ^R to σ^S . In practice we choose Q to be a diagonal matrix whose elements form $q \in [0, 1]^C$, which we refer to as the query vector.

3.2.3 Choosing the query

For local editing, an appropriate choice for the query q is one that favors channels that affect the region of interest (ROI), while ignoring channels that have an effect outside

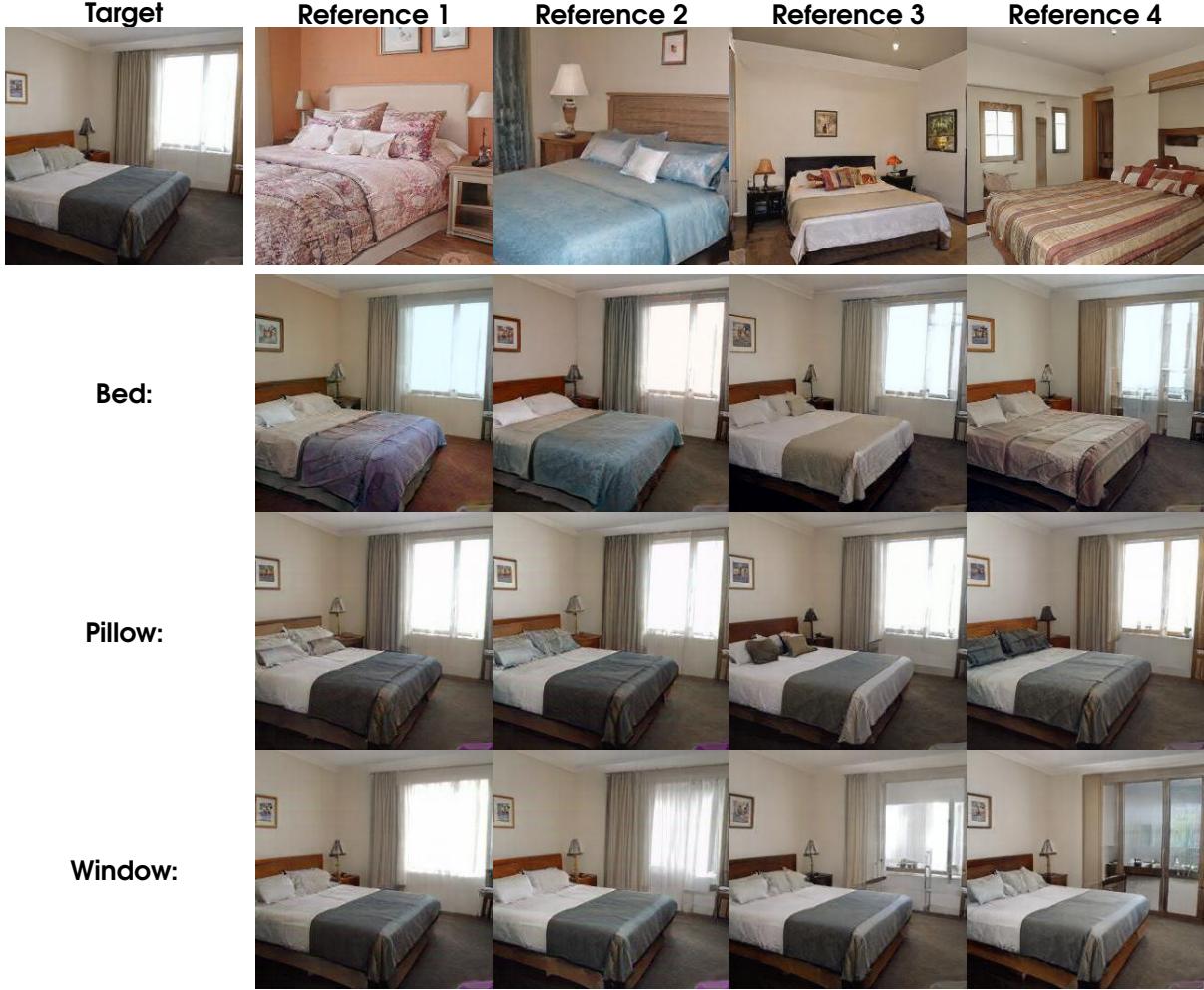


Figure 4: Unlike previous blending methods[24, 26], our method does not require images to be aligned or of similar scale. In this case, the style of, e.g., the bed is successfully transferred from reference to target in spite of drastic changes in view point.

the ROI. When specifying the ROI using one of the semantic clusters computed in section 3.1, say k' , the vector $\mathbf{M}_{k',c}$ encodes exactly this information.

A simple approach is to use $\mathbf{M}_{k',c}$, computed offline from Eq. (1) for a given genre and dataset of images, to control the slope of the interpolation, clipping at 1:

$$\mathbf{q}_c = \min(1, \lambda \mathbf{M}_{k',c}) \quad (4)$$

where \mathbf{q}_c is the c -th channel element of \mathbf{q} , and λ , as in Eq. (2), is the global strength of the interpolation. We refer to this approach as *simultaneous* as it updates all channels at the same time. Intuitively, when λ is small or intermediate, channels with large $\mathbf{M}_{k',c}$ will have a higher weight, thus having an effect of localizing the interpolation.

We propose an approach which achieves superior localization compared to Eq. (4), referred to as *sequential*. We first set the most relevant channel to the maximum slope

of 1, before raising the slope of the second-most relevant, third-most, etc. This definition of the query corresponds to solving for the following objective:

$$\begin{aligned} & \arg \min_{\mathbf{q}_c} \mathbf{q}_c [\mathbf{M}_{k',c} - \rho(1 - \mathbf{M}_{k',c})] \\ & \text{s.t. } \sum_{c=1}^C \mathbf{q}_c (1 - \mathbf{M}_{k',c}) \leq \epsilon \\ & \quad 0 \leq \mathbf{q}_c \leq 1 \end{aligned} \quad (5)$$

We solve this objective by sorting channels based on $\mathbf{M}_{k'}$, and greedily assigning $\mathbf{q}_c = 1$ to the most relevant channels as long as the total effect outside the ROI is no more than some budget ϵ . Additionally, a non-zero weight is only assigned to channels where $\mathbf{M}_{k',c} > \frac{\rho}{1+\rho}$, which improves the robustness of local editing by ignoring irrelevant channels even when the budget ϵ allows more change.

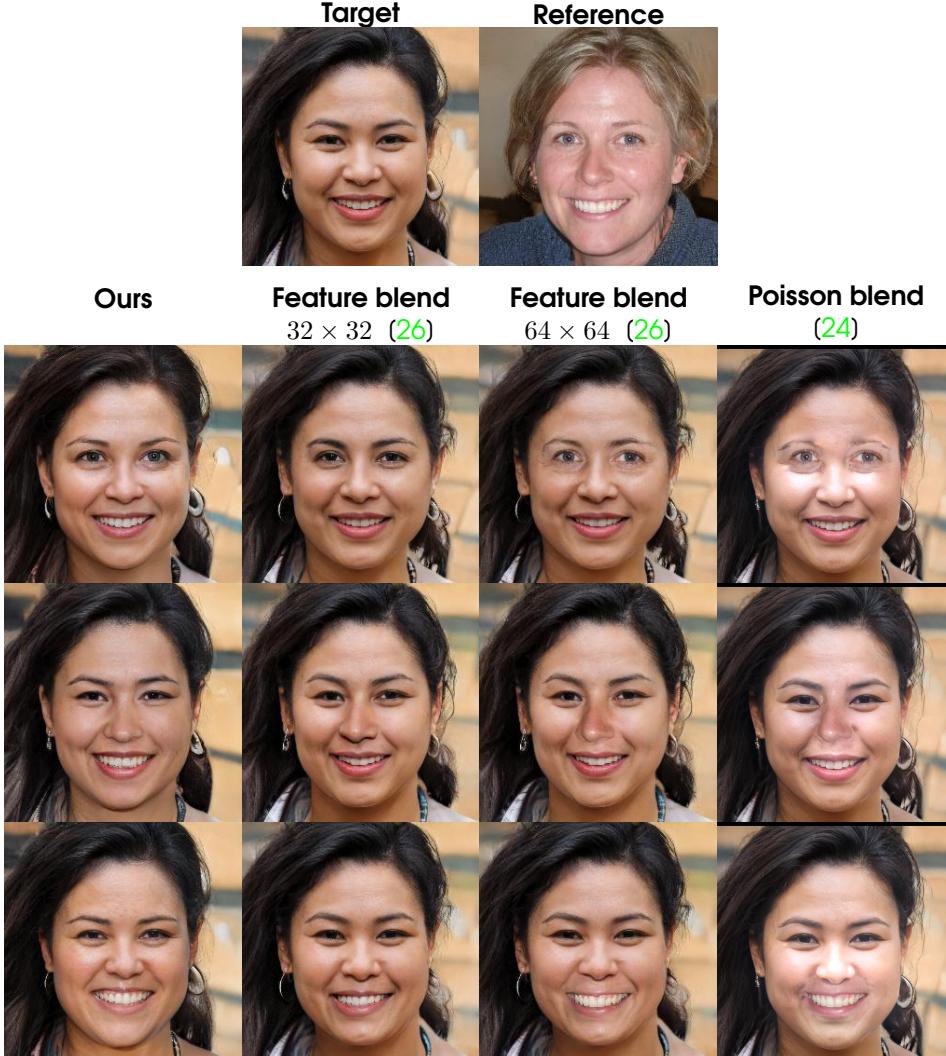


Figure 5: Even the well aligned FFHQ-generated faces prove challenging for existing blending methods, as they do not consider differences in pose and scale, and lack any notion of semantics or photorealism. In contrast, our method makes use of the correlation GANs learn from real data to maintain a natural appearance, while exploiting feature disentanglement for effectively localizing the change.

4. Experiments

4.1. Qualitative evaluation

In Figs. 3 and 4 we demonstrate our editing method² with StyleGAN generators trained on two datasets: FFHQ [16] comprising 70K facial images and LSUN-Bedrooms [29] comprising about 3M color images depicting bedrooms.

In both datasets, we found the first 32×32 resolution layer of the generator to be “most semantic”. We therefore chose this layer to apply spherical k-means clustering. We

²Our code is available online at: <https://github.com/IVRL/GANLocalEditing>

set ρ such that $\frac{\rho}{1+\rho} = 0.1$ and tune $20 \leq \epsilon \leq 100$ for best performance. We found that the tuning of ϵ depends mostly on the target image and object of interest, and not the style reference. Note that by nature of the *local* edit, changes to the target image may be subtle, and best viewed on screen.

Fig. 5 compares our method with feature-level blending [26] and pixel-level (Poisson) blending [24] methods. Feature blending is applied once to all layers of resolution 32×32 or lower, and once to those of 64×64 or lower. While these approaches are strictly localized (see section 2.1), their outputs lack photorealism. For instance, the target and reference faces are facing slightly different directions, which causes a misalignment problem most visible in

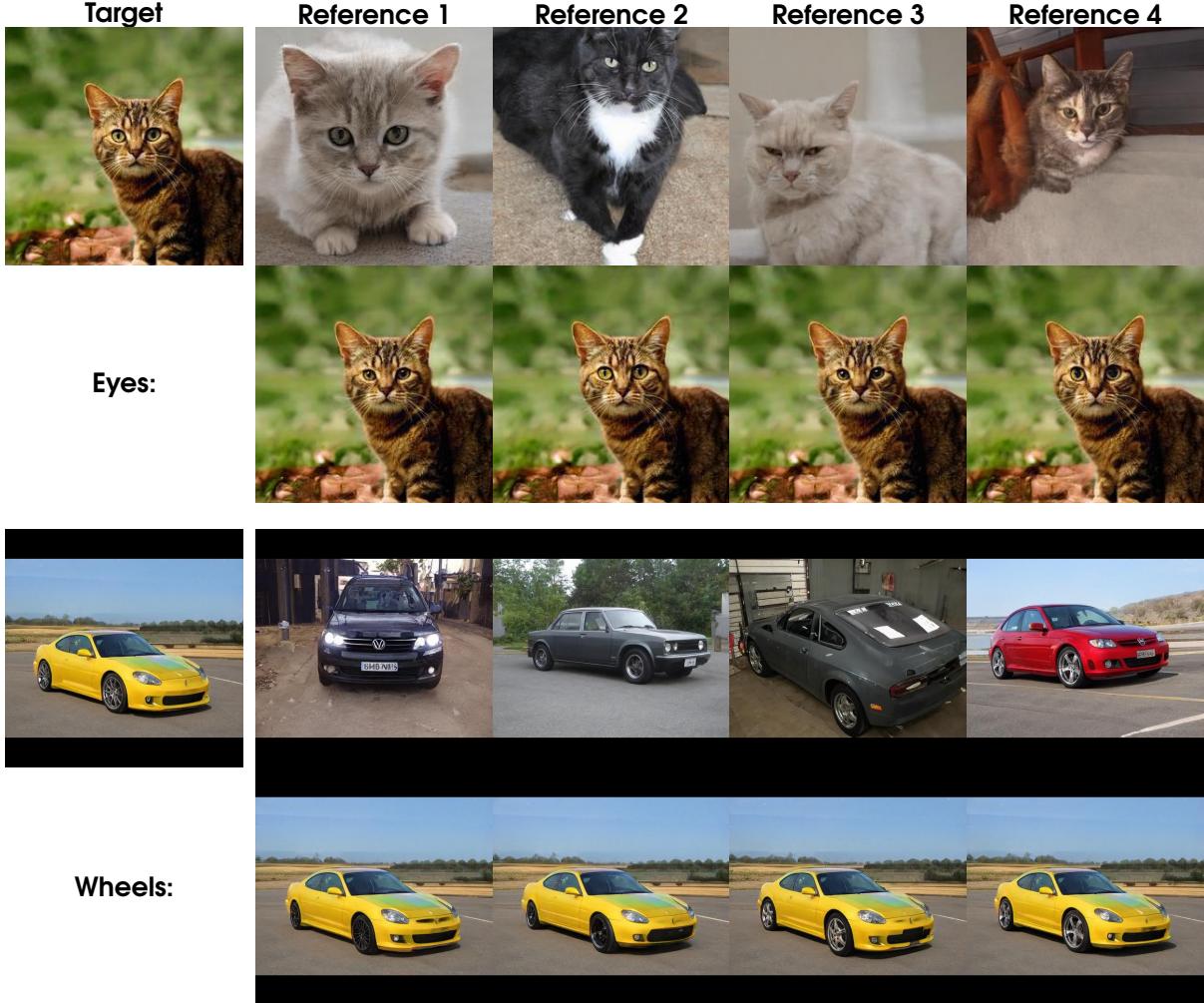


Figure 6: Our method applied to StyleGAN2 outputs. Photorealism is preserved while allowing fine control over highly-localized regions.

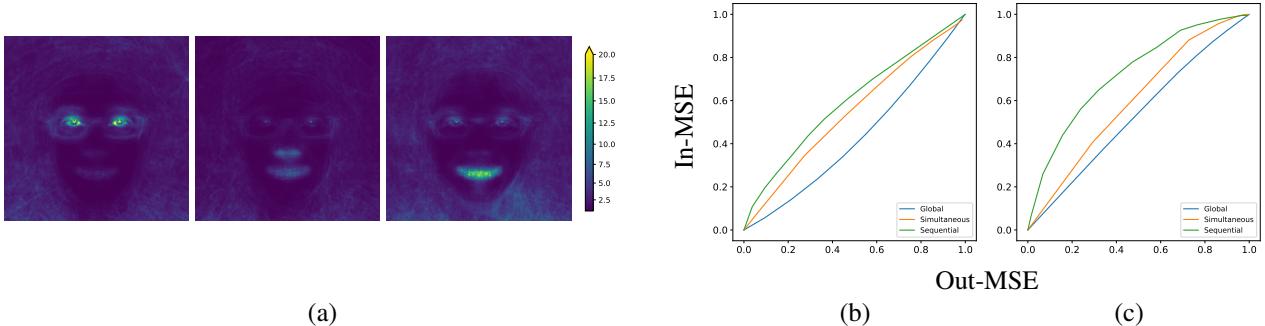


Figure 7: (a) Mean squared-error (MSE) heatmaps computed between 50K FFHQ-StyleGAN outputs and their edited counterparts for *eyes* (left), *nose* (center) and *mouth* (right). These heatmaps demonstrate that our method produces edits that are both perceptible and localized. (b) MSE **inside** the ROI vs. **outside** for various query parameters on FFHQ and (c) similarly for LSUN-Bedrooms. Our approach, *sequential*, has the best trade-off between In-MSE (high is good) and Out-MSE (low is good).

the *nose*. In contrast, our editing method primarily affects the ROI, and yet maintains the photorealism of the baseline GAN by admitting *some* necessary global changes. However, our method does not always copy the appearance of an object ‘faithfully’, as seen in the *window* row of Fig. 4.

Fig. 6 demonstrates the applicability of our method to the recent StyleGAN2 model [17] trained on LSUN-Cats and LSUN-Cars [29]. Unlike traditional blending methods, our technique is able to transfer parts between unaligned images as seen here and in Fig. 4.

4.2. Quantitative analysis

We quantitatively evaluate the results of editing on two aspects of interest: locality and photorealism.

4.3. Locality

To evaluate the locality of editing, we examine the squared-error in pixel space between target images and their edited outputs. Fig. 7 (a) shows the difference between unedited and edited images averaged over 50K FFHQ-StyleGAN samples, where at every pixel location we compute the squared distance in CIELAB color space. This figure indicates that the transfers are both perceptible and localized, and that not all object parts are equally disentangled. Compared to *eyes* and *mouth*, where edits are very localized, editing the *nose* seems to force a subtle degree of correlation with the other face parts. Such correlations trade-off control on the appearance of individual parts versus plausibility and realism of the overall output.

We further examine the localization ability of our method and variants described in Section 3.2. First, we obtain for each image the binary mask indicating the ROI, using the pre-computed spherical k-means clusters of Section 3.1. Then, we perform interpolation with various values of λ (Eqs. 2 and 4) and ϵ (Eq. 5). For each such setting we measure the (normalized) In- and Out-MSE of each target-output pair, i.e., the MSE inside the ROI and MSE outside the ROI, respectively. In Fig. 7 (b) and (c), we show that for both FFHQ and LSUN-Bedrooms, respectively, our method (*sequential*) has better localization, i.e., less change outside the ROI for the same amount of change inside the ROI.

4.4. Photorealism

Measuring photorealism is challenging, as there is not yet a direct computational way of assessing the *perceived* photorealism of an image. The Frchet Inception Distance (FID), however, [11] has been shown to correlate well with human judgement and has become a standard metric for GAN evaluation.

An aggregate statistic, FID compares the distributions of two image sets in the feature space of a deep CNN layer. In Table 1 we report the FID of 50K edited images against the original FFHQ and LSUN-Bedrooms datasets. The FID

FID	FFHQ	Bedrooms
StyleGAN[16]	4.4	2.8
Ours	5.4	4.5
Feature blending 64 × 64[26]	5.4	-

Table 1: Frchet Inception Distance between ground truth images and outputs of StyleGAN, our method, and feature blending. In the case of LSUN-Bedrooms, the non-aligned nature of the images makes feature blending inapplicable.

scores indicate that our edited images are not significantly different from the vanilla output of the baseline GAN.

However, the same result was achieved when we computed the FID of 50K FFHQ images edited with feature blending [26], although Fig. 5 shows qualitatively that these produced outputs lack photorealism. This reemphasizes the difficulty of correctly measuring photorealism in an automated way. We did not run a similar analysis with Poisson blending since the many failure cases we observed with this approach did not justify the heavy computational cost required to process a large collection of 1024×1024 images. For both feature blending and Poisson editing, we could not test the Bedrooms dataset since these methods are not suitable for unaligned image pairs.

5. Conclusion

We have demonstrated that StyleGAN’s latent representations spatially disentangle semantic objects and parts. We leverage this finding to introduce a simple method for local semantic part editing in StyleGAN images. The core idea is to let the latent object representation guide the style interpolation to produce realistic part transfers without introducing any artifacts not already inherent to StyleGAN. The locality of the result depends on the extent to which an object’s representation is disentangled from other object representations, which in the case of StyleGAN is significant. Importantly, our technique does not involve external supervision by semantic segmentation models, or complex spatial operations to define the edit region and ensure a seamless transition from edited to unedited regions.

For future investigation, our observations open the door to explicitly incorporate editing capabilities into the adversarial training itself, which we believe will improve the extent of disentanglement between semantic objects, and yield even better localization.

Finally, the method can, in principle, be extended to semantic editing of real images by leveraging the frameworks of [1], [17] to first map natural images into the latent space of StyleGAN. This opens up interesting applications in photo enhancement, augmented reality, visualization for plastic surgery, and privacy preservation.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the StyleGAN latent space? In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 8
- [2] Sarah Alotaibi and William A. P. Smith. Biofacenet: Deep biophysical face image interpretation. *arXiv preprint arXiv:1908.10578v2*, 2019. 2
- [3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding gans. *arXiv preprint arXiv:1811.10597v2*, 2018. 2
- [4] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096v2*, 2019. 2
- [5] Christian Buchta, Martin Kober, Ingo Feinerer, and Kurt Hornik. Spherical k-means clustering. *Journal of Statistical Software*, 2012. 3
- [6] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *European Conference on Computer Vision (ECCV)*, 2018. 3
- [7] Antonio Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative adversarial networks: An overview. In *IEEE Signal Processing Magazine*, pages 53–65, 2018. 2
- [8] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3
- [9] I. Goodfellow, M. Pouget-Abadie, B. Mirza, D. Xu, S. WardeFarley, A. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2
- [10] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. AttGAN: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 2019. 2
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 8
- [12] Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. 1
- [13] He Huang, Philip S. Yu, and Changhu Wang. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469v2*, 2018. 2
- [14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (CVPR)*, 2017. 3
- [15] Tero Karras, Timo Aila, Samuli Laine, and Jaakkko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 11
- [16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4, 6, 8, 11
- [17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakkko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *arXiv preprint arXiv:1912.04958*, 2019. 1, 2, 8
- [18] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3677–3685, 2017. 2
- [19] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc’Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2
- [20] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *26th International Joint Conference on Artificial Intelligence, IJCAI*, 2017. 3
- [21] Saleh Mosaddegh, Loc Simon, and Frédéric Jurie. Photorealistic face de-identification by aggregating donors face components. In *Asian Conference on Computer Vision*, 2014. 2
- [22] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Moriishim. RSGAN: face swapping and editing using face and hair representation in latent space. *arXiv preprint arXiv:1804.03447v2*, 2018. 2
- [23] Guim Perarnau, Joost Van de Weijer, and Bogdan Raducanu. Invertible conditional GANs for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 2
- [24] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 2, 5, 6
- [25] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [26] Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable image synthesis with internal representation collaging. *arXiv preprint arXiv:1811.10153*, 2018. 2, 5, 6, 8
- [27] Xiang Wang, Kai Wang, and Shiguo Lian. A survey on face data augmentation. *arXiv preprint arXiv:1904.11685*, 2019. 2
- [28] Fei Yang, Ju Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3d-aware face component transfer. *ACM Trans. Graphics*, 30(4), 2011. 2
- [29] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3, 6, 8
- [30] Xiaowu Tang, Bolei Zhou, Yujun Shen, Jinjin Gu. Interpreting the latent space of GANs for semantic face editing. *arXiv preprint arXiv:1907.10786*, 2019. 2, 3
- [31] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *European Conference on Computer Vision (ECCV)*, 2018. 2

- [32] Jun-Yan Zhu, Philipp Krhenbhl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016. [2](#)

Appendix

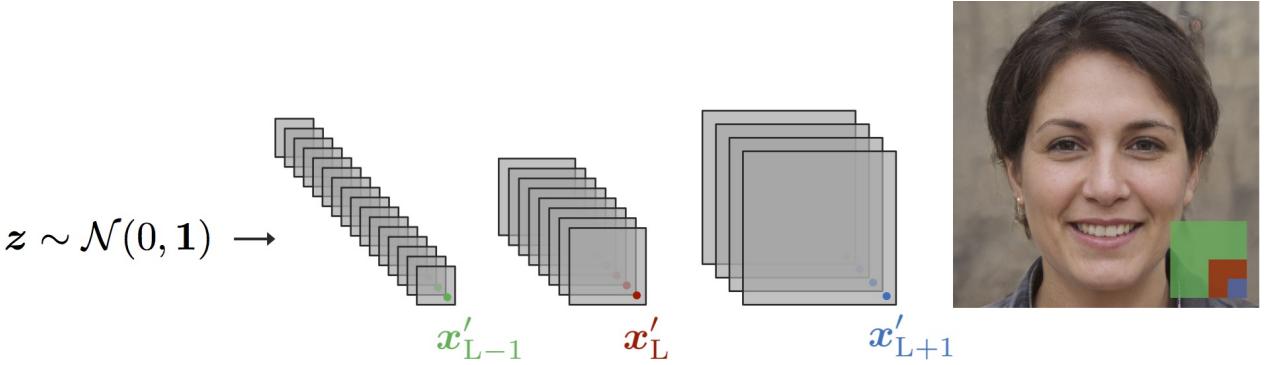


Figure 8: In convolutional generative networks, the vector x' at a single spatial position on a hidden feature map at some layer L corresponds to a whole *patch* on the RGB image. The lower the resolution of the hidden layer, the larger the patch in pixel space.

A. Spherical k-means for semantic clustering

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

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 zero-mean unit-variance activations for a batch of N images, where at each channel the feature map has spatial dimensions $H \times W$.

As shown in Fig. 8, a vector $a \in \mathbb{R}^C$ 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 $\mathbf{A} \in \mathbb{R}^{(N \cdot H \cdot W) \times C}$, i.e., to a “bag-of-patch-embeddings” representation, with no explicit encoding of spatial position or even partitioning into different samples of the batch. 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 $\mathbf{A} \approx \mathbf{U}\mathbf{V}$, where the binary matrix $\mathbf{U} \in \{0, 1\}^{(N \cdot H \cdot W) \times K}$ encodes cluster membership and the matrix $\mathbf{V} \in \mathbb{R}^{K \times C}$ encodes the unit-length centroid of each cluster.

The matrix \mathbf{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 [16], where the tensor \mathbf{U} is up-sampled and overlaid on RGB images for ease of interpretation. The color-coding in these figures indicates to which cluster a spatial position belongs. In Fig. 11 we similarly show results for ProgGAN [15] on CelebA-HQ [15].

The main observation emerging from this analysis is that

at certain layers (e.g., the 32×32 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 $N = 200$ samples and recorded all their activations. We tested several layers and rank 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.

B. 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.

C. 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 on FFHQ. Additional examples can be found on the paper’s GitHub page, linked above.

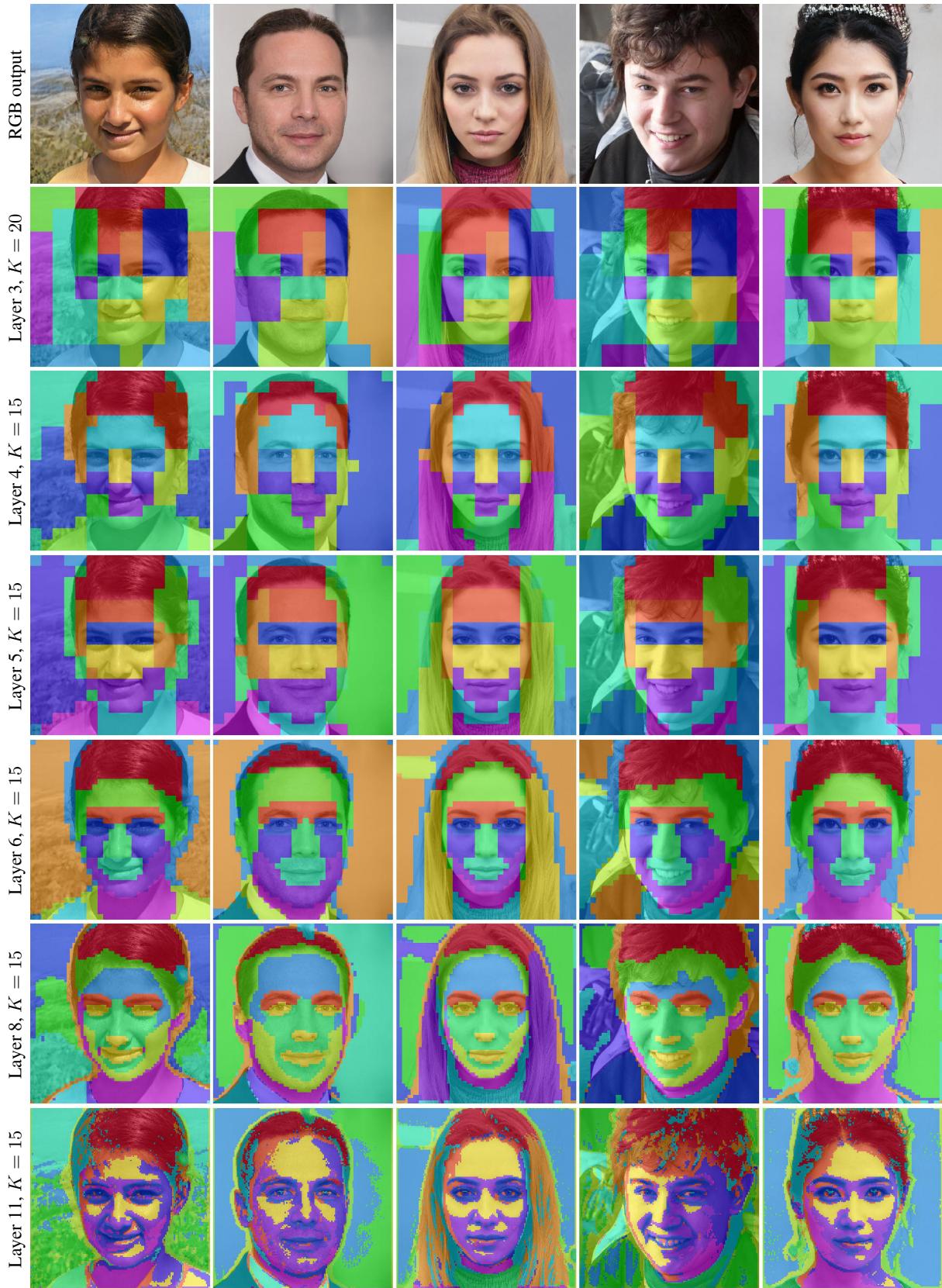


Figure 9: Spherical k-means cluster membership maps for various FFHQ-StyleGAN layers. Color-coding signifies different clusters, and is arbitrarily determined per layer.



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-ProGAN 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.

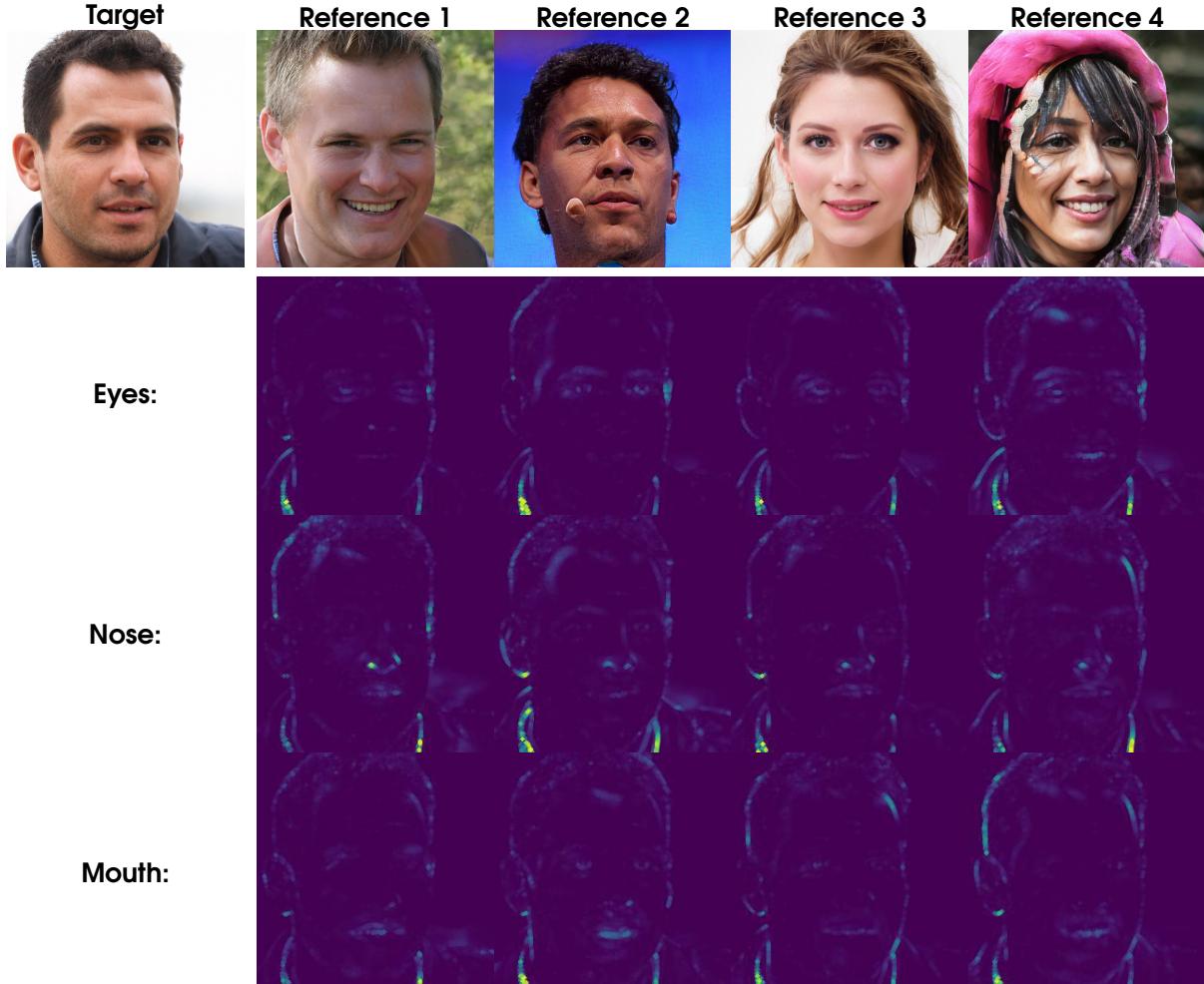


Figure 13: Mean-squared error maps between the edited outputs shown in Fig. 3 and the target image, shown in the same figure. Editing is primarily focused on the object of interest, though some subtle changes do occur elsewhere in the scene.



Figure 14: Mean-squared error maps corresponding to Fig. 4. 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.

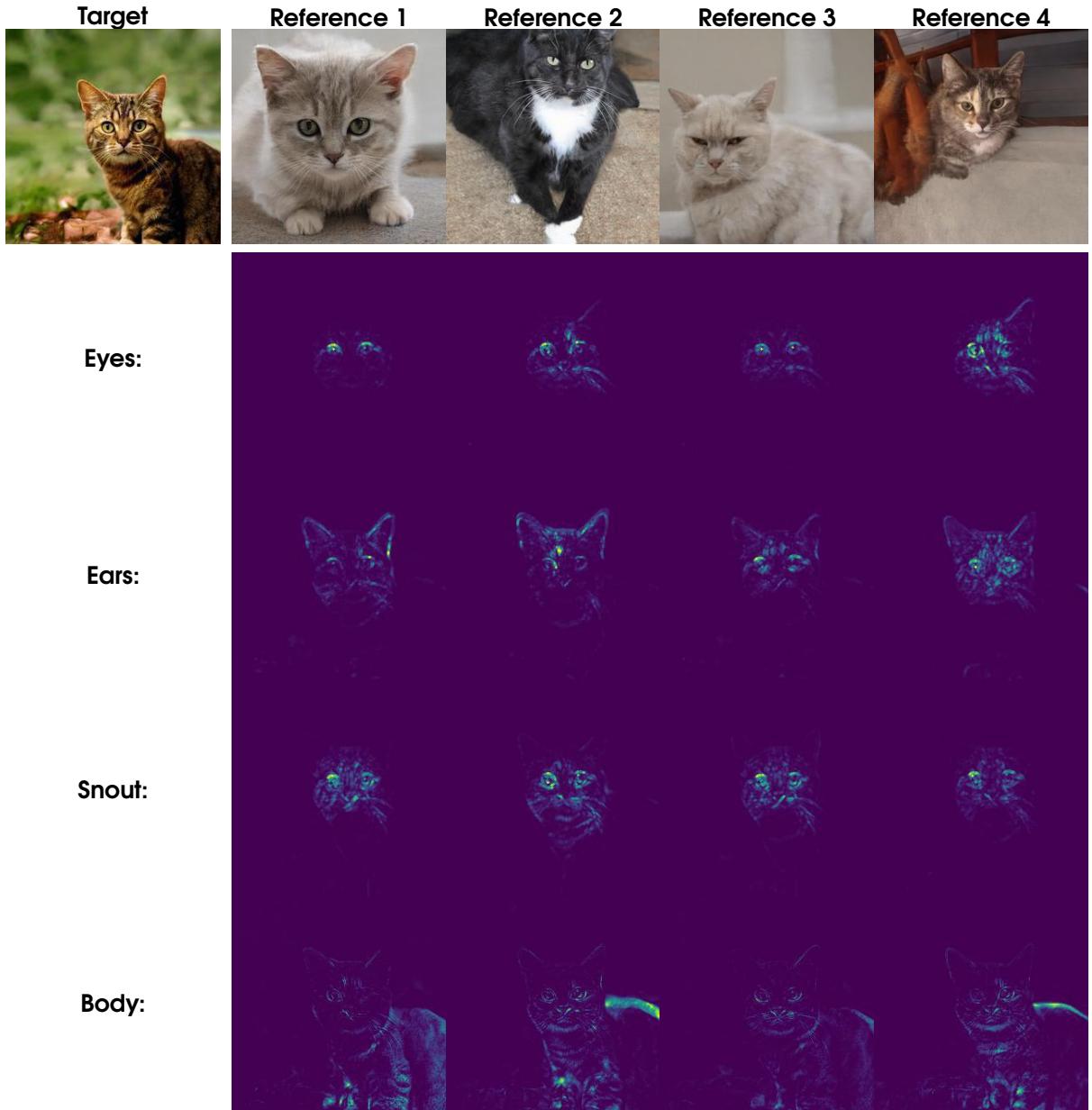


Figure 16: Diff maps corresponding to the results in Fig. 15, for StyleGAN2 trained on LSUN-Cats.

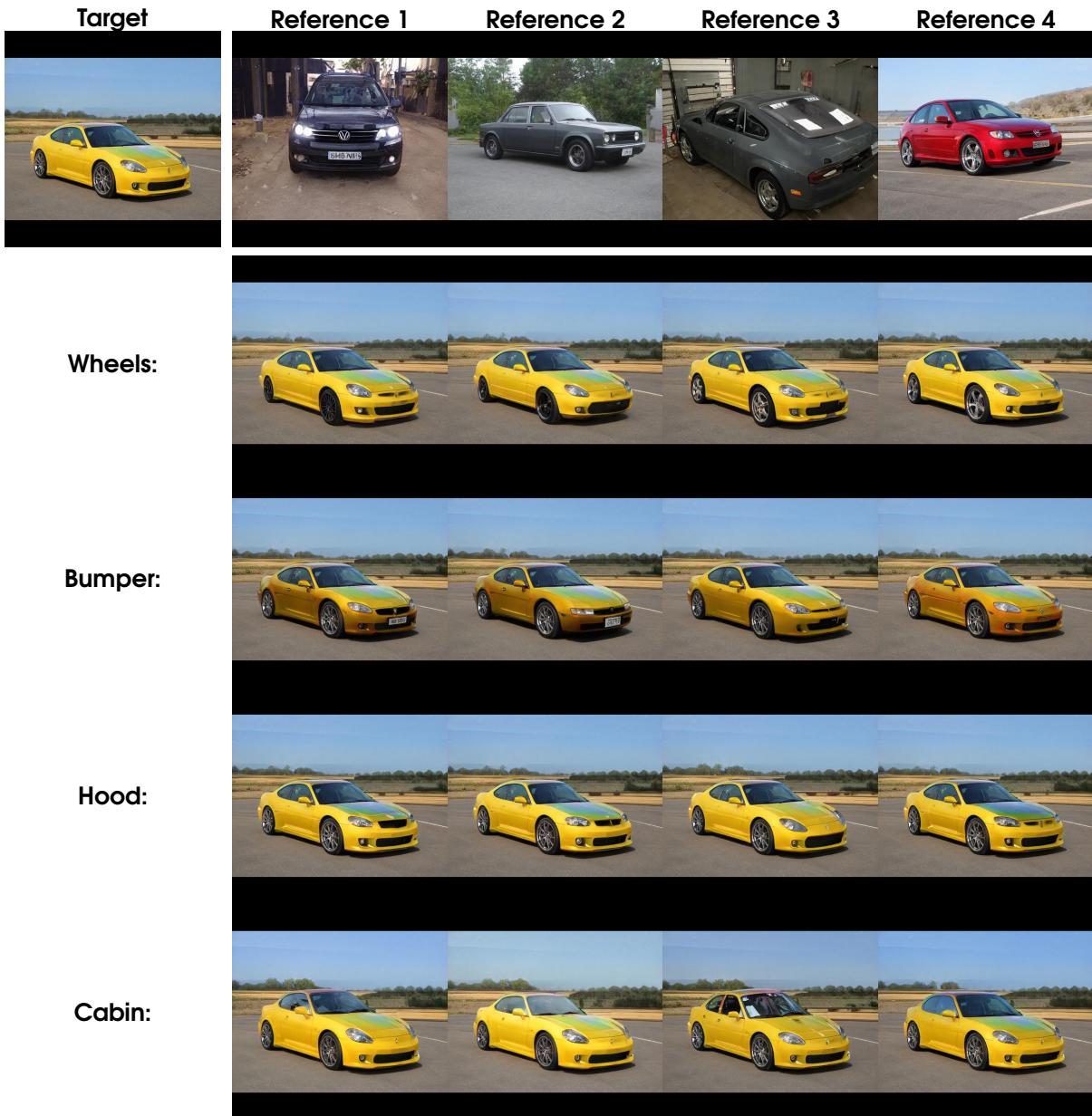


Figure 17: Our local editing method applied to StyleGAN2 trained on LSUN-Cars.

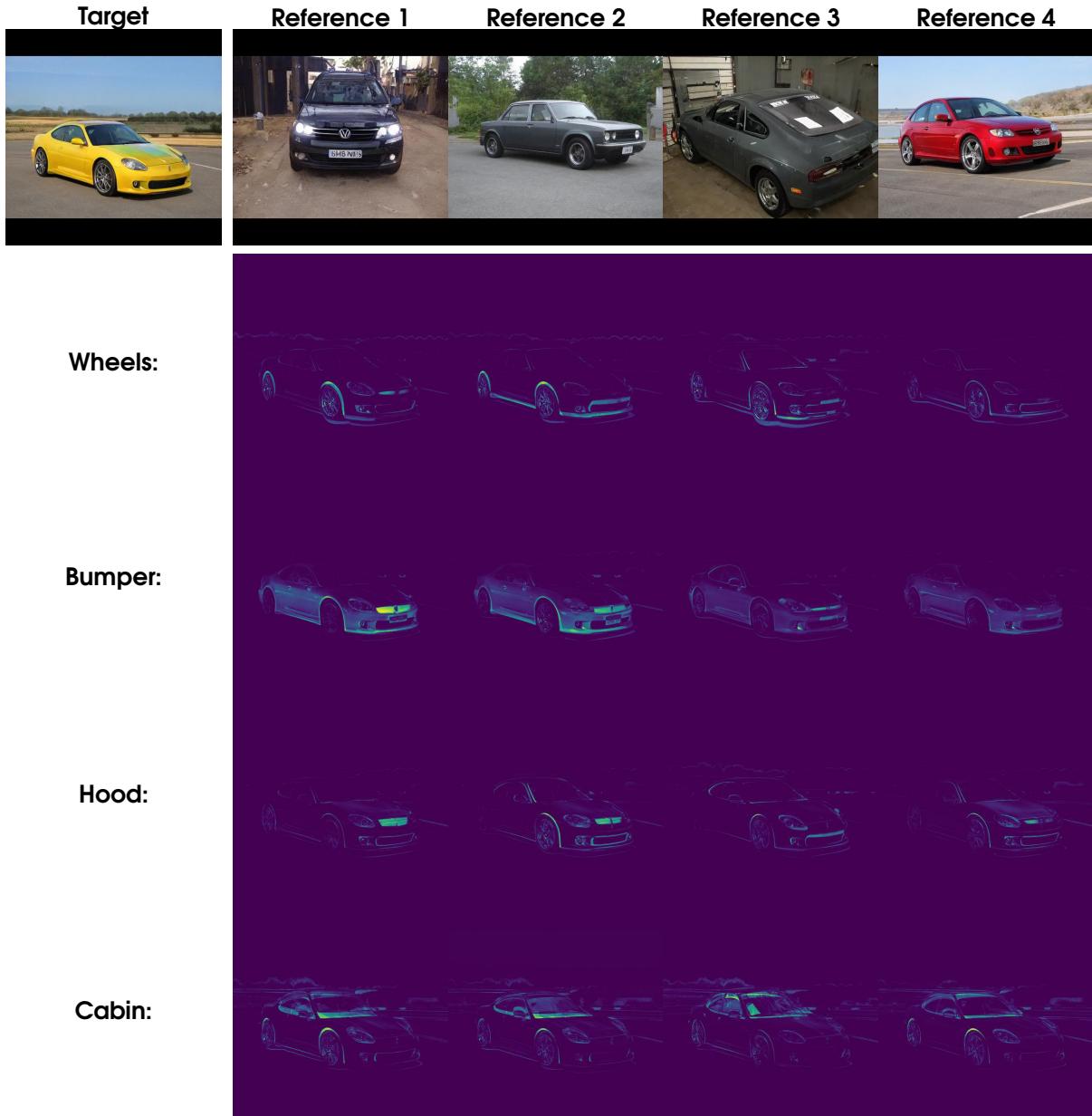


Figure 18: Diff maps corresponding to the results in Fig. 17, for StyleGAN2 trained on LSUN-Cars.

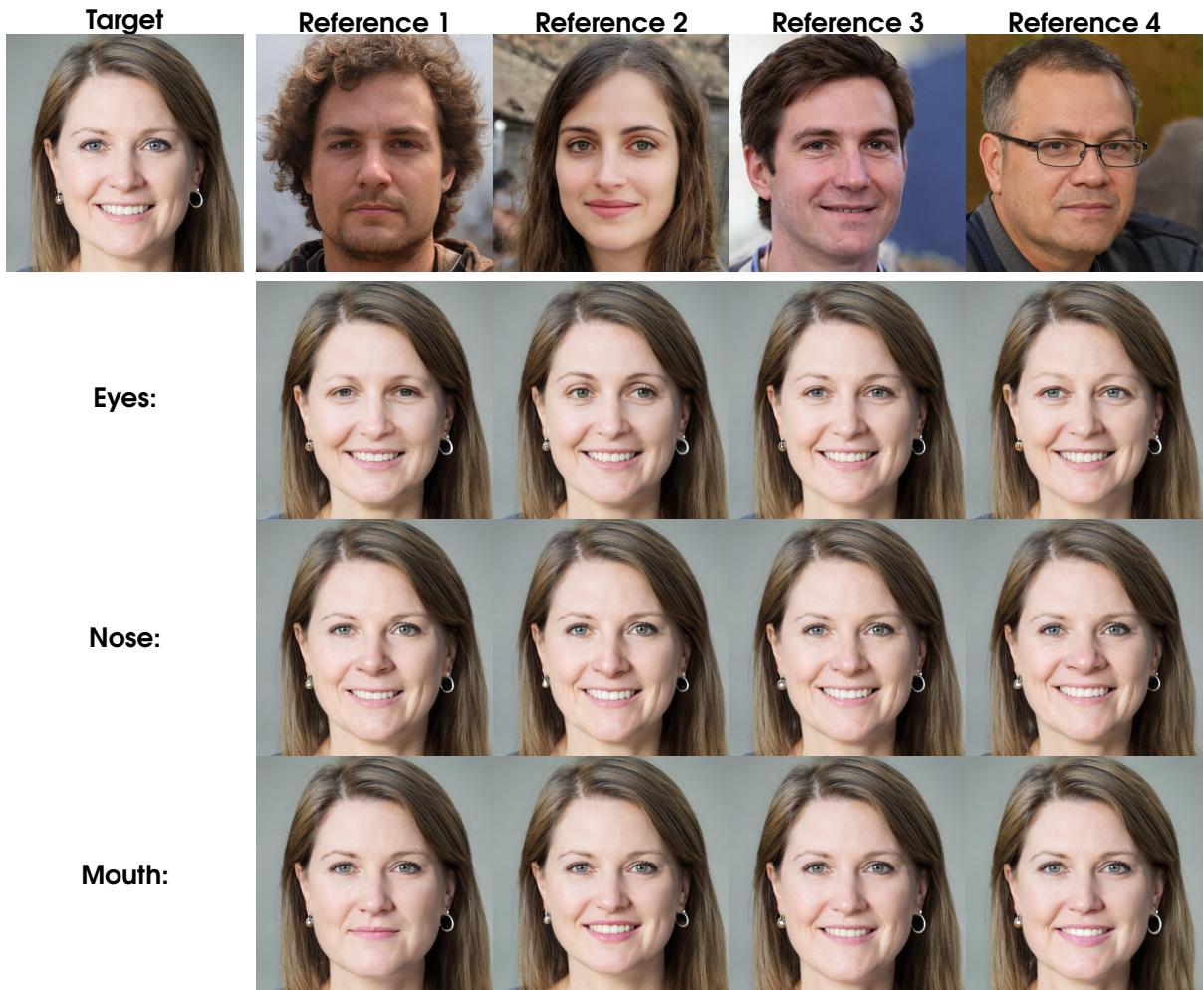


Figure 19: Our local editing method applied to StyleGAN2 trained on FFHQ.

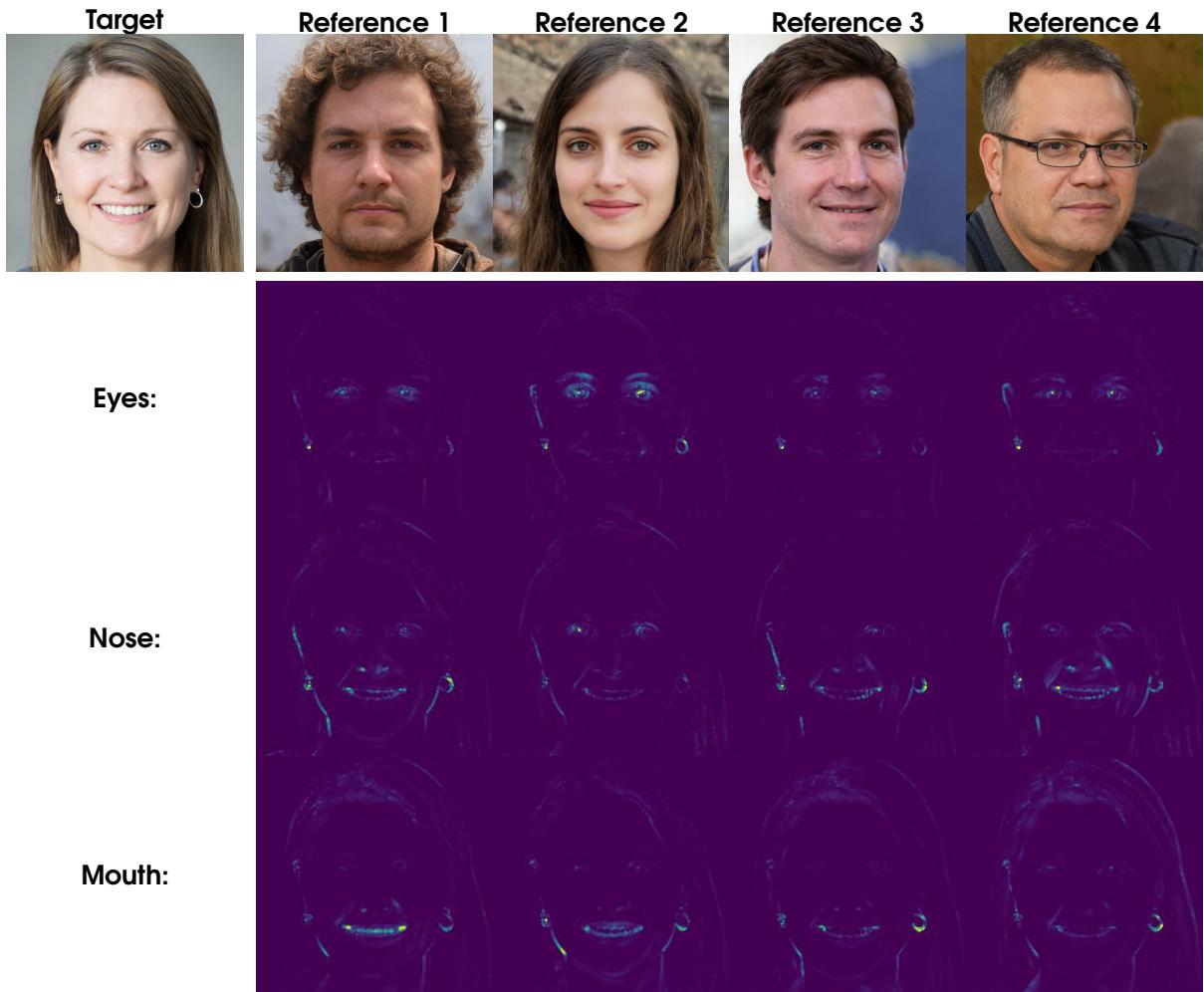


Figure 20: Diff maps corresponding to the results in Fig. 19, for StyleGAN2 trained on FFHQ.