StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery (Supplementary Material)

Or Patashnik[†*] Zongze Wu[‡*] Eli Shechtman[§] Daniel Cohen-Or[†] Dani Lischinski[‡]
[‡]Hebrew University of Jerusalem [†]Tel-Aviv University [§]Adobe Research

Input Single network Ours (three networks)

Figure 11. Comparing our mapper architecture with a simpler architecture that uses a single mapping network. The simpler mapper fails to infer multiple changes correctly. The changes in the expression and in the hair-style are not strong enough to capture the identity of the target individual. On the other hand, there are unnecessary changes in the background color in the second row when using a single network.

9. Latent Mapper – Ablation Study

In this section, we study the importance of various choices in the design of our latent mapper (Section 5).

9.1. Architecture

The architecture of the mapper is rather simple and with relatively small number of parameters. Moreover, it has negligible effect on the inference time. Yet, it is natural to compare the presented architecture, which consists of three different mapping networks, to an architecture with a single mapping network. Intuitively, using a separate network for each group of style vector entries should better enable changes at several different levels of detail in the image. Indeed, we find that with driving text that requires such changes, e.g. "Donald Trump", a single mapping network does not yield results that are as effective as those produced with three. An example is shown in Figure 11.

*Equal contribution, ordered alphabetically

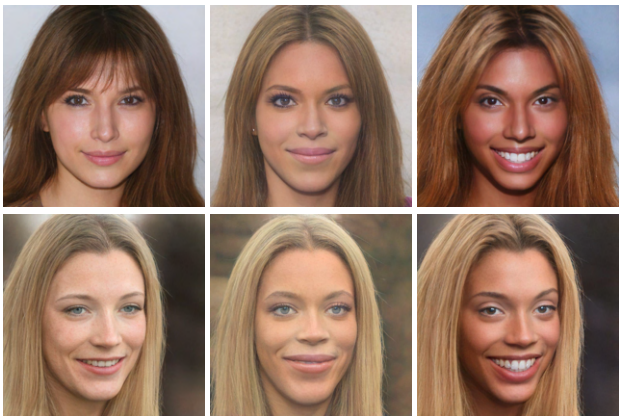
Input Curly default Curly without M^f Mohawk default Mohawk without M^f

Figure 12. Removing M^f from our full architecture for edits which do not require color scheme manipulation yields slightly better results.

Although the full, three network mapper, gives better results for some driving texts, as mentioned in Section 5, we note that not all the three are needed when the manipulation should not affect some attributes. For example, for the hairstyle edits shown in Figure 5, the manipulation should not affect the color scheme of the image. Therefore, we perform these edits when training M^c and M^m only, that is, $M_t(w) = (M_t^c(w_c), M_t^m(w_m), 0)$. We show a comparison in Figure 12. As can be seen, by removing M_f from the architecture, we get slightly better results. Therefore, for the sake of simplicity and generalization of the method, we chose to describe the method with all three networks. In the main paper, the results shown were obtained with all three networks, while here we also show results with only two (without M_f).

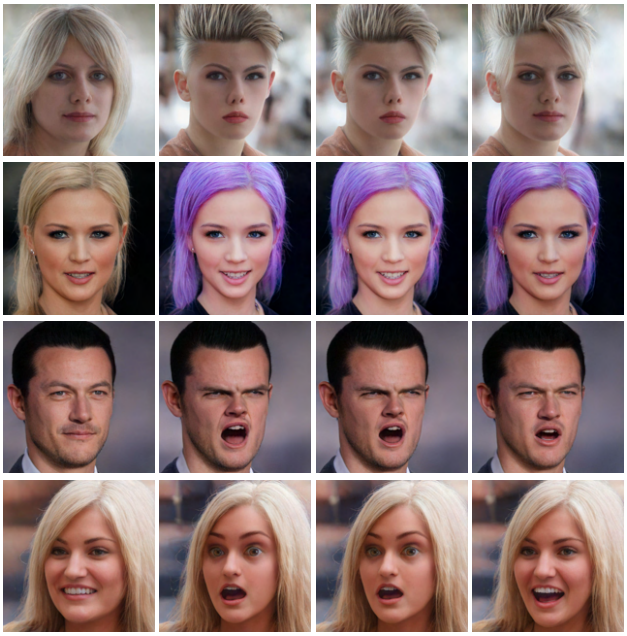
9.2. Losses

CLIP Loss To show the uniqueness of using a "celeb edit" with CLIP, we perform the following experiment. Instead of using the CLIP loss, we use the identity loss with respect to a single image of the desired celeb. Specifically, we perform this experiment by using an image of Beyonce. The results are shown in Figure 13. As can be seen, CLIP guides the mapper to perform a unique edit which cannot be achieved by simply using a facial recognition network.



Input ID Loss CLIP Loss

Figure 13. Replacing the CLIP loss with identity loss for the Beyonce edit. The identity loss is computed with respect to an image of Beyonce.



Input (0.8, 0) (1.6, 0) (0.8, 0.1)

Figure 14. Identity loss ablation study. Under each column we specify $(\lambda_{L2}, \lambda_{ID})$. In the second and the third columns we did not use the identity loss. As can be seen, the identity of individual in the input image is not preserved.

ID Loss Here we show that the identity loss is significant for preserving the identity of the person in the input image. When using the default parameter setting of $\lambda_{L2} = 0.8$ with $\lambda_{ID} = 0$ (i.e., no identity loss), we observe that the mapper fails to preserve the identity, and introduces large changes. Therefore, we also experiment with $\lambda_{L2} = 1.6$, however, this still does not preserve the original identity well enough. The results are shown in Figure 14.

10. Working in Different Latent Spaces

In this section, we compare the optimization and latent mapper methods applied in different latent spaces. Specifically, in addition to the results shown previously, where the methods were applied in $\mathcal{W}+$, here we apply these methods in the StyleSpace, $\mathcal{S}$ [10].

Optimization Here, instead of optimizing the latent vector $w \in \mathcal{W}+$, we optimize the latent vector in $\mathcal{S}$. We optimize the exact same expression, where instead of using w, we use $s \in \mathcal{S}$. We found that optimizing all of the dimensions of $\mathcal{S}$ does not yield good results. Thus, we do not optimize the style vectors which are fed into the tRGB layers. We refer the reader to Wu et al. [10] for more details about $\mathcal{S}$ space. The results of the three configurations are shown in Figure 15. The results obtained with $\mathcal{S}$ when not optimizing the tRGB style vectors, and the results obtained with $\mathcal{W}+$ are quite similar. However, we noted that the optimization in $\mathcal{S}$ is somewhat more disentangled, as can be seen in the "beard" example. However, this disentanglement comes with a cost. In $\mathcal{S}$, we find it harder to perform more global changes, as can be seen in the "Elsa" and "Trump" examples.

Latent Mapper We now turn to study the performance of the latent mapper applied in $\mathcal{S}$. As demonstrated for the optimization method, changing the style parameters that are fed into the tRGB layers does not contribute to changing semantic attributes of the image. Therefore, the mapper does not change these style vectors, and they are left intact. We also note that the different style vectors are produced by different learned affine transformations and therefore it is natural to to define a different mapper for each style vector. The latent mapper that operates in $\mathcal{S}$ is therefore defined by 16 different networks, each of which yields a displacement for a single style vector (which is not fed into the tRGB layers). In Figure 16, we compare the results obtained by the mapper that operates in $\mathcal{S}$ to those obtained by the mapper that operates in $\mathcal{W}+$. As can be seen, although $\mathcal{S}$ is more disentangled than $\mathcal{W}+$ the results are similar.

11. Additional Results

In this section we provide additional results to those presented in the paper. Specifically, we begin with a variety of image manipulations obtained using our latent mapper. All manipulated images are taken from the CelebA-HQ and were inverted by e4e [9].

1. In Figure 17 we show a large gallery of hair style manipulations.

2. In Figures 18 and 19 we show "celeb" edits, where the

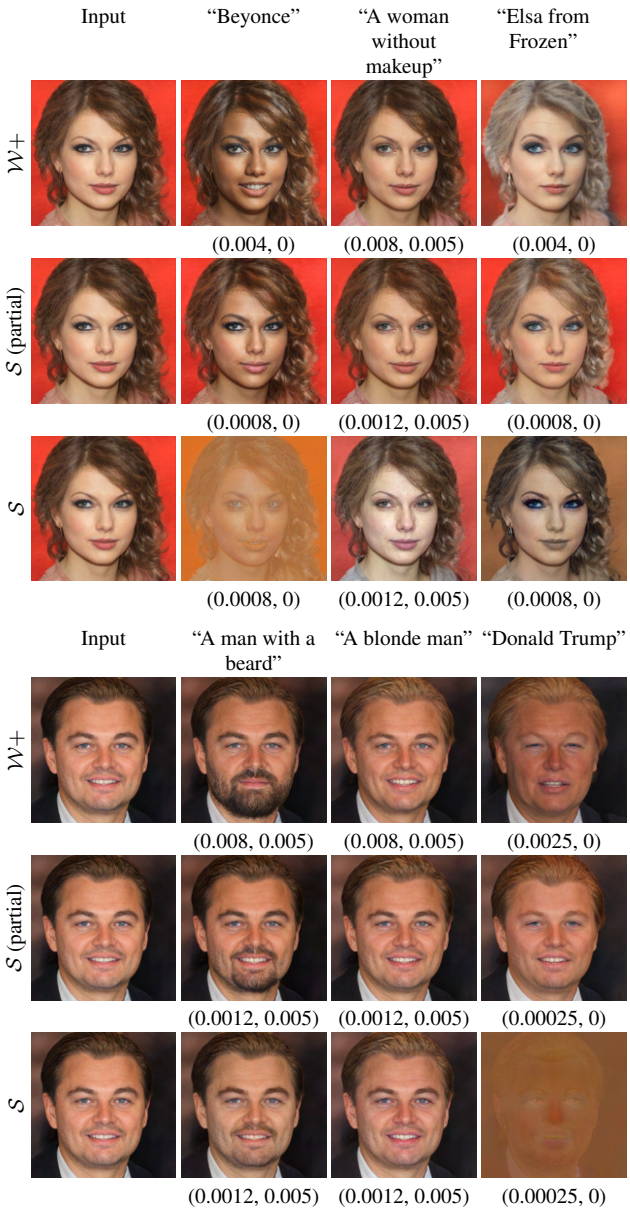


Figure 15. Comparison of optimization in different latent spaces. We show optimization in $\mathcal{W}+$, in $\mathcal{S}$, and in $\mathcal{S}$ when not using the style vectors that go to the tRGB layers, denoted by $\mathcal{S}$ (partial).

input image is manipulated to resemble a certain target celebrity.

3. In Figure 20 we show a variety of expression edits.

Next, Figure 21 shows a variety of edits on non-face datasets, performed along text-driven global latent manipulation directions (Section 6).

Figure 22 shows image manipulations driven by the prompt "a photo of a male face" for different manipulation strengths and disentanglement thresholds. Moving along the global direction, causes the facial features to become more masculine, while steps in the opposite direction yields more feminine features. The effect becomes stronger as the strength α increases. When the disentanglement threshold β is high, only the facial features are affected, and as β is lowered, additional correlated attributes, such as hair length and facial hair are affected as well.

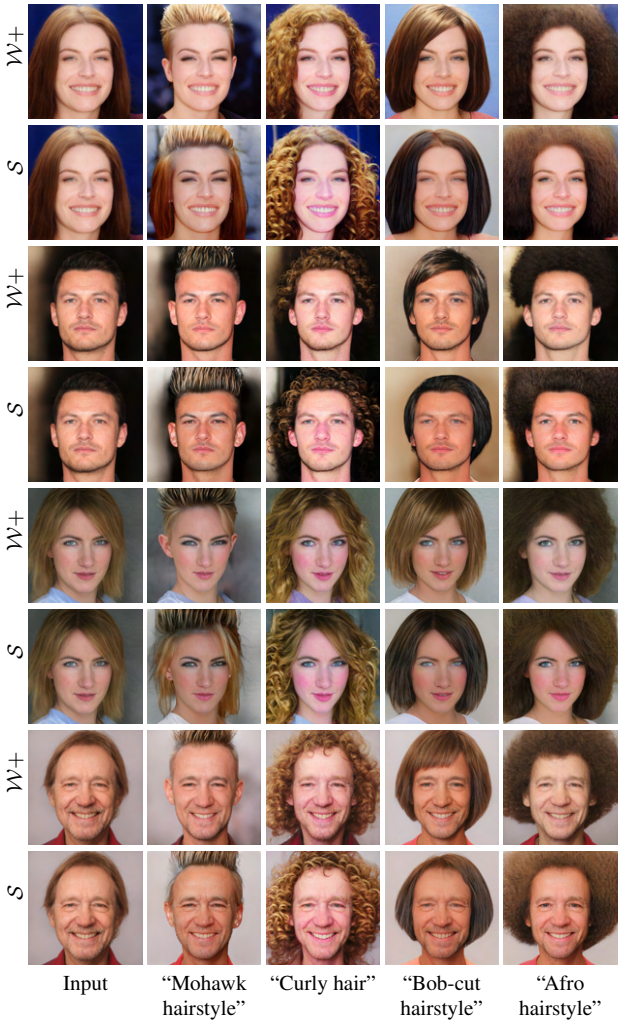


Figure 16. Comparison of the latent mapper in different latent spaces. For $\mathcal{W}+$ we trained a mapper which consists of 3 networks: M^c, M^m, M^f, and for $\mathcal{S}$ we trained a mapper which consists of 16 networks for the style vectors that are not inserted into the tRGB layers.

In Figure 23, we show another comparison between our global direction method and several state-of-the-art StyleGAN image manipulation methods: GANSpace [4], InterFaceGAN [8], and StyleSpace [10]. The comparison only examines the attributes which all of the compared methods are able to manipulate (Gender, Grey hair, and Lipstick), and thus it does not include the many novel manipulations enabled by our approach. Following Wu *et al*. [10], the manipulation step strength is chosen such that it induces

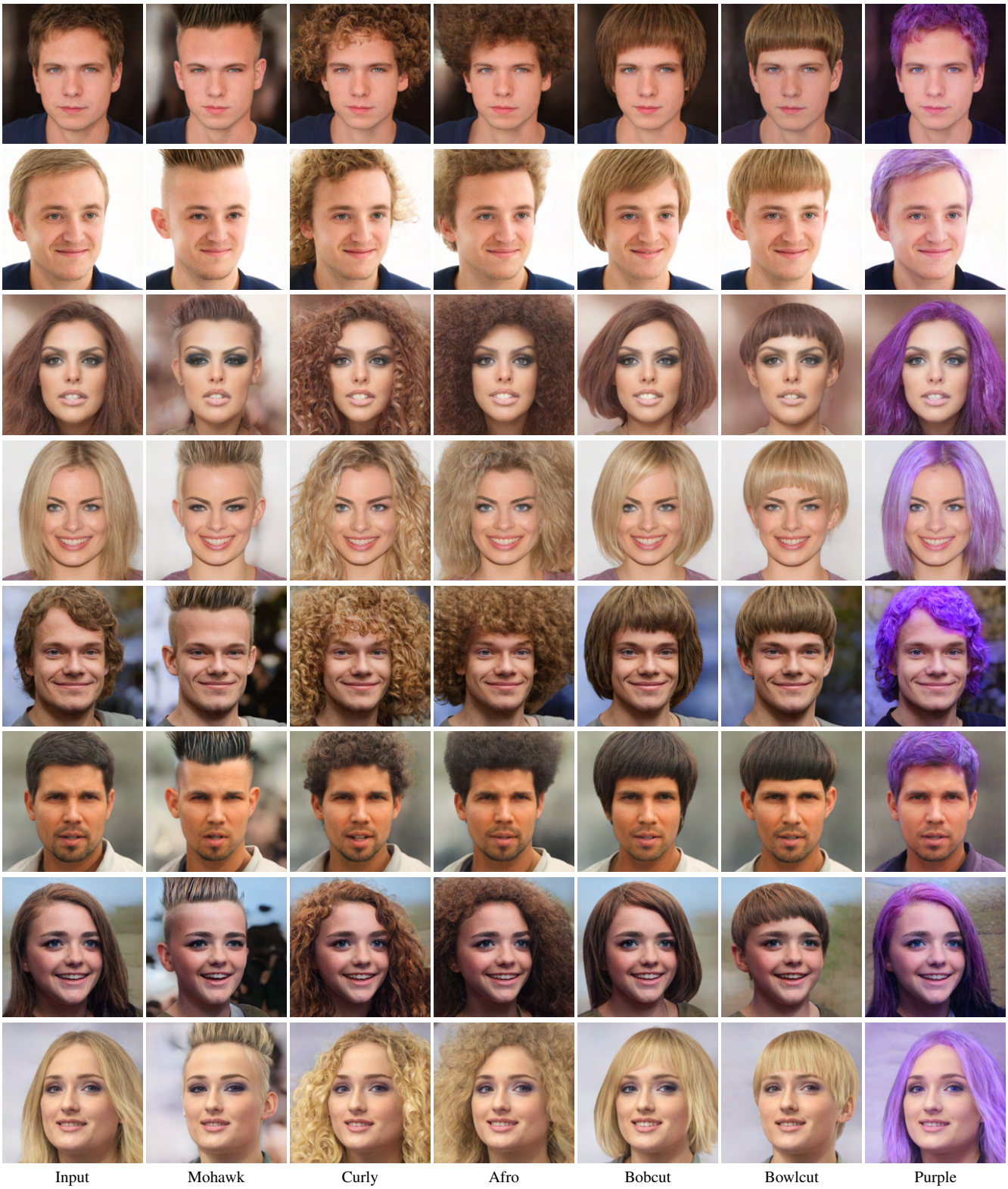
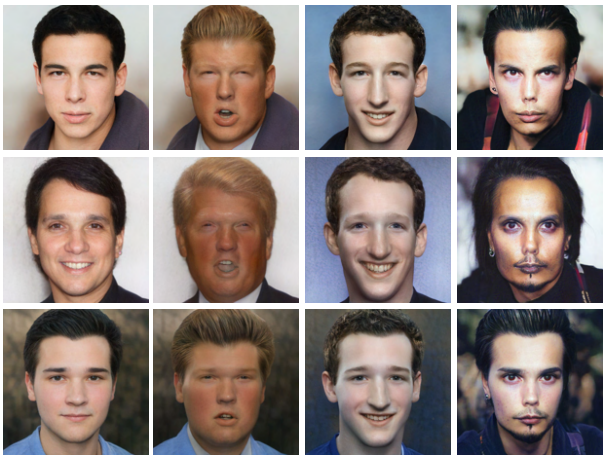


Figure 17. Hair style manipulations obtained by the latent mapper. Except for the purple hair, all mappers were trained without M^f.

Input Taylor Swift Beyonce Hillary Clinton

Figure 18. Celeb edits performed by the latent mapper.



Input Trump Mark Zuckerberg Johnny Depp

Figure 19. Celeb edits performed by the latent mapper.

the same amount of change in the logit value of the corresponding classifiers (pretrained on CelebA). It may be seen that in GANSpace [4] manipulation is entangled with skin color and lighting, while in InterFaceGAN [8] the identity may change significantly (when manipulating Lipstick). Our manipulation is very similar to StyleSpace [10], which only changes the target attribute, while all other attributes remain the same.

Figure 24 show additional edits along global text-driven manipulation directions, demonstrated on portraits of celebrities. Edits are performed using StyleGAN2 pretrained on FFHQ [6]. The inputs are real images, embedded in $\mathcal{W}+$ space using the e4e encoder [9].

Figure 25 shows a comparison between StyleFlow [1]

Input Surprised Angry

Figure 20. Expression edits performed by the latent mapper.

and our global directions method. It may be seen that our method is able to produce results of comparable visual quality, despite the fact that StyleFlow requires the simultaneous use of several attribute classifiers and regressors (from the Microsoft face API), and is thus able to manipulate a limited set of attributes. In contrast, our method required no extra supervision to produce these and all of the other manipulations demonstrated in this work.

Figure 26 shows an additional comparison between text-driven manipulation using our global directions method and our latent mapper. Our observations are similar to the ones we made regarding Figure 10 in the main paper.

Finally, Figure 27 demonstrates that drastic manipulations in visually diverse datasets are sometimes difficult to achieve using our global directions. Here we use StyleGAN-ada [5] pretrained on AFHQ wild [2], which contains wolves, lions, tigers and foxes. There is a smaller domain gap between tigers and lions, which mainly involves color and texture transformations. However, there is a larger domain gap between tigers and wolves, which, in addition to color and texture transformations, also involves more drastic shape deformations. This figure demonstrates that our global directions method is more successful in transforming tigers into lions, while failing in some cases to transform tigers to wolves.

12. Video

We show examples of interactive text-driven image manipulation in our supplementary video. We use a simple heuristic method to determine the initial disentanglement threshold (β). The threshold is chosen such that k channels will be active. For real face manipulation, we set the initial strength to $\alpha = 3$ and the disentanglement threshold so that $k = 20$. For real car manipulation, we set the initial values to $\alpha = 3$ and $k = 100$. For generated cat manipulation, we set the initial values to $\alpha = 7$ and $k = 100$.

13. Limitations

Our methods rely on a well trained disentangled StyleGAN2 [7] model. Such models shine on datasets of aligned images, where the main object is in the center of the image, such as FFHQ [6] or AHFQ [2]. But they perform less admirably on more general images, which may depict more complex scenes with multiple objects spread across the image, such as images from the LSUN bedrooms [11] or Cityscape [3] datasets. Figure 28 shows several text-driven manipulations using a StyleGAN2 [7] model pretrained on LSUN bedrooms [11] using our global directions method. It may be seen that while text-driven manipulation is possible in this model, the results are somewhat entangled.

References

[1] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. StyleFlow: attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *CoRR*, abs/2008.02401, 2020. 5, 9

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proc. CVPR*, pages 8188–8197, 2020. 6, 7, 10

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, pages 3213–3223, 2016. 6

[4] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. *arXiv preprint arXiv:2004.02546*, 2020. 3, 5

[5] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020. 6, 7, 10

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410, 2019. 5, 6, 8

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, pages 8110–8119, 2020. 6

[8] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: interpreting the disentangled face representation learned by GANs. *CoRR*, abs/2005.09635, 2020. 3, 5

[9] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *CoRR*, abs/2102.02766, 2021. 2, 5, 8

[10] Zongze Wu, Dani Lischinski, and Eli Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. *arXiv:2011.12799*, 2020. 2, 3, 5

[11] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, 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. 6, 7

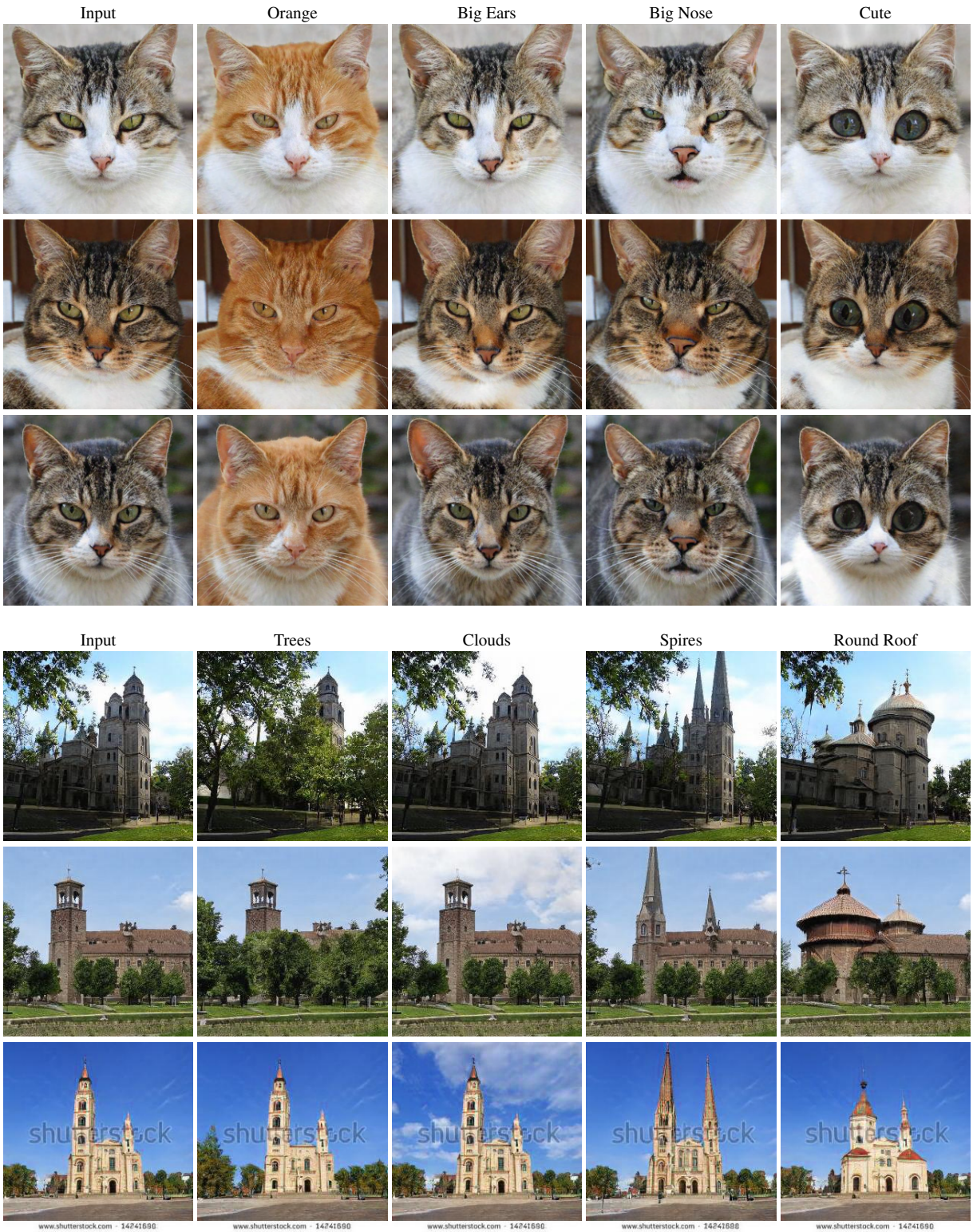


Figure 21. A variety of edits for non-face images along text-driven global latent manipulation directions. Left: using StyleGAN2-ada [5] pretrained on AFHQ cats [2]. Right: using StyleGAN2 pretrained on LSUN Church [11]. The target attribute used in the text prompt is indicated above each column.

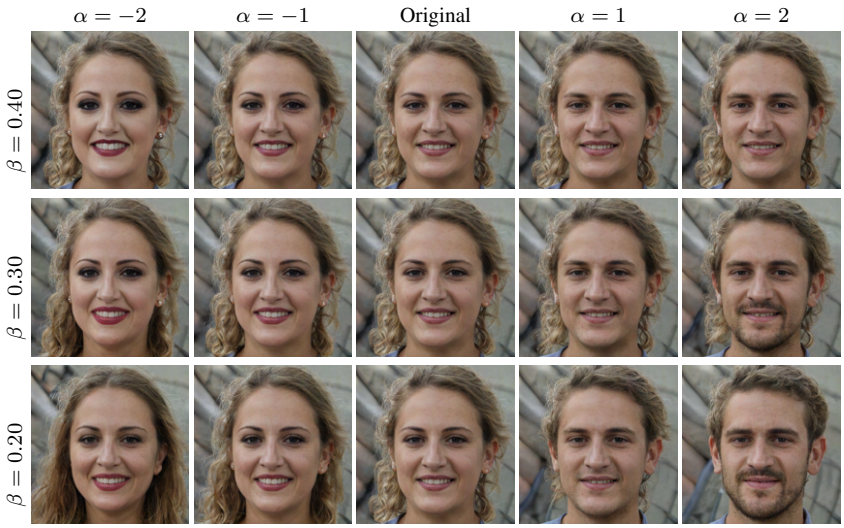


Figure 22. We demonstrate gender manipulation (driven by the prompt "a photo of a male face") for different manipulation strengths and disentanglement thresholds. Moving along the global direction, causes the facial features to become more masculine, while steps in the opposite direction yields more feminine features. The effect becomes stronger as the strength α increases. When the disentanglement threshold β is high, only the facial features are affected, and as β is lowered, additional correlated attributes, such as hair length and facial hair are affected as well.

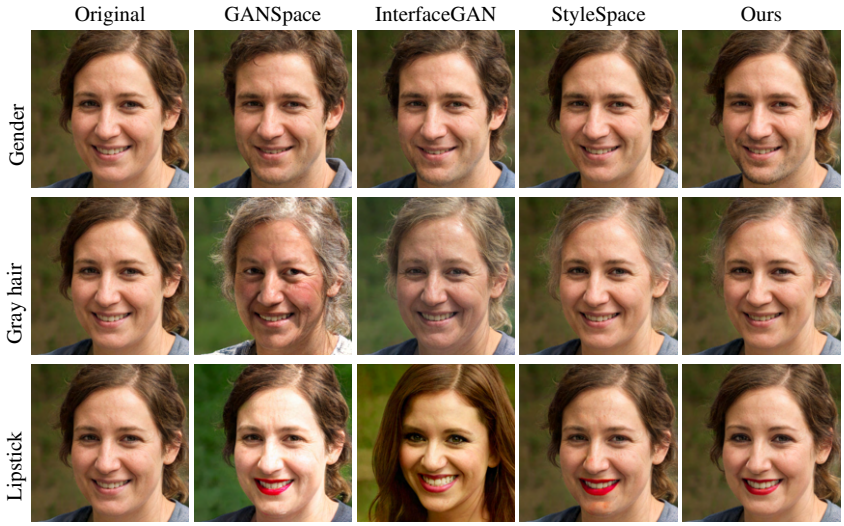


Figure 23. Comparison with state-of-the-art methods using the same amount of manipulation according to a pretrained attribute classifier.

Figure 24. A variety of edits along global text-driven manipulation directions, demonstrated on portraits of celebrities. Edits are performed using StyleGAN2 pretrained on FFHQ [6]. The inputs are real images, embedded in $\mathcal{W}+$ space using the e4e encoder [9]. The target driving text is indicated above each column.

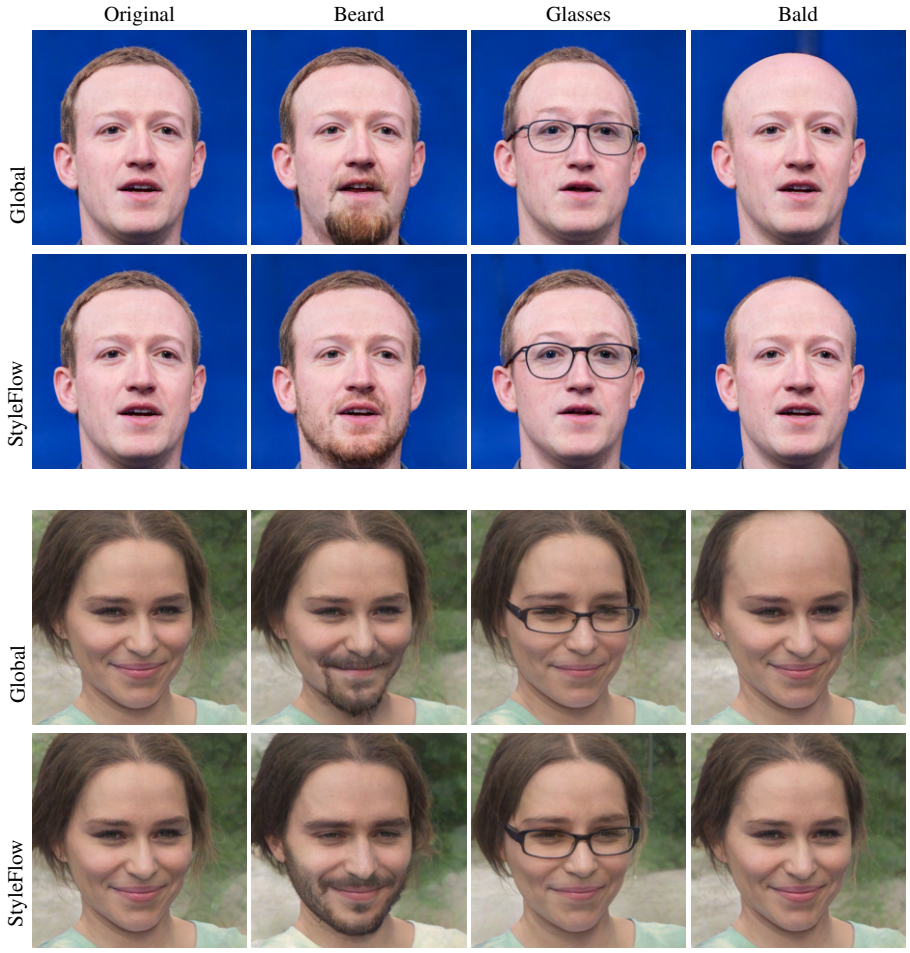


Figure 25. Comparison between StyleFlow [1] and our global directions. Our method produces results of similar quality, despite the fact that StyleFlow simultaneously uses several attribute classifiers and regressors (from the Microsoft face API), and is thus able to manipulate a limited set of attributes. In contrast, our method requires no extra supervision.

Figure 26. We compare our global directions with our latent mapper using three different kinds of attributes.

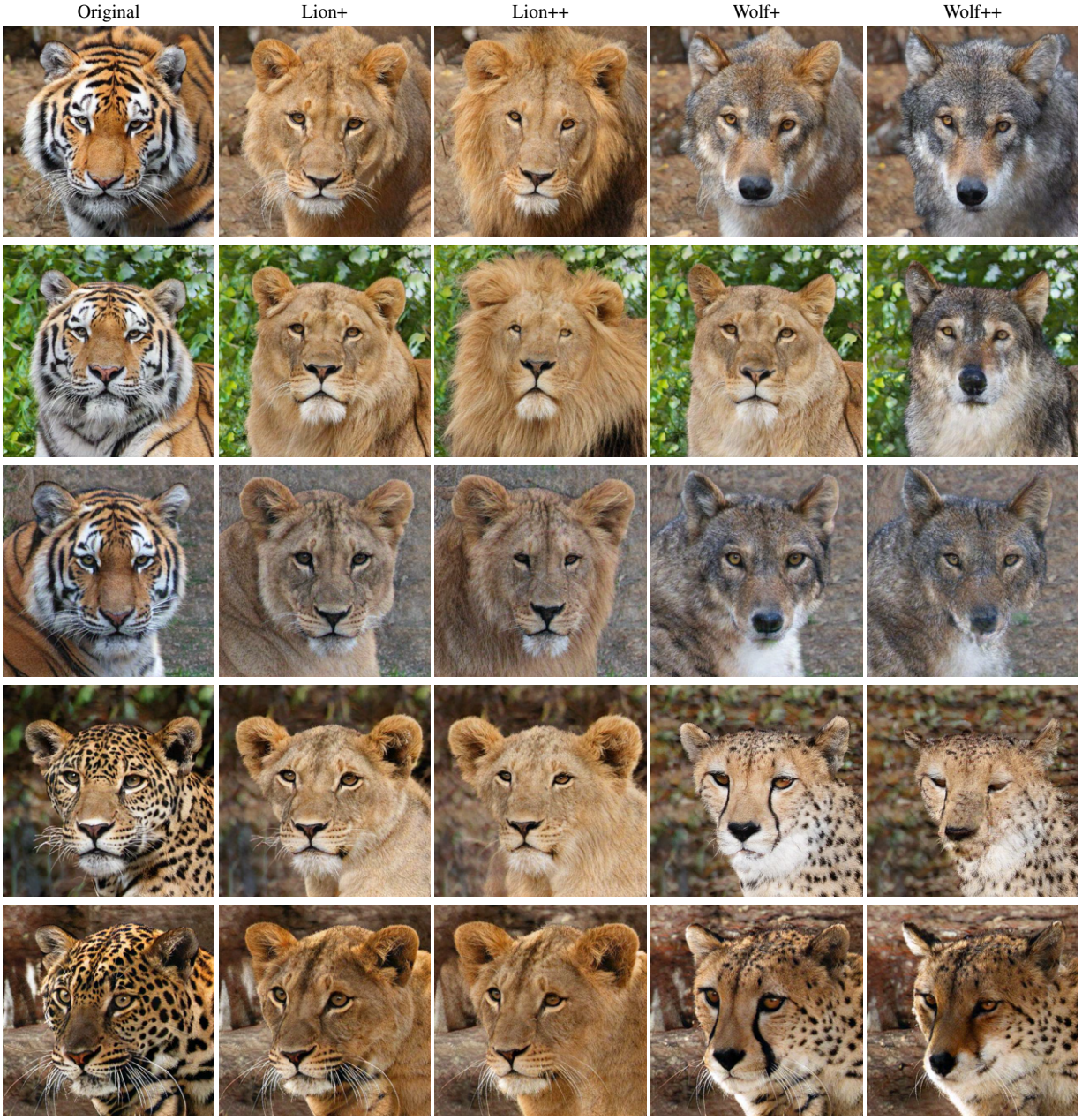


Figure 27. Drastic manipulations in visually diverse datasets are sometimes difficult to achieve using our global directions. Here we use StyleGAN-ada [5] pretrained on AFHQ wild [2], which contains wolves, lions, tigers and foxes. There is a smaller domain gap between tigers and lions, which mainly involves color and texture transformations. However, there is a larger domain gap between tigers and wolves, which, in addition to color and texture transformations, also involves more drastic shape deformations. This figure demonstrates that our global directions method is more successful in transforming tigers into lions, while failing in some cases to transform tigers to wolves. The "+" and "++" indicate medium and strong manipulation strength, respectively.

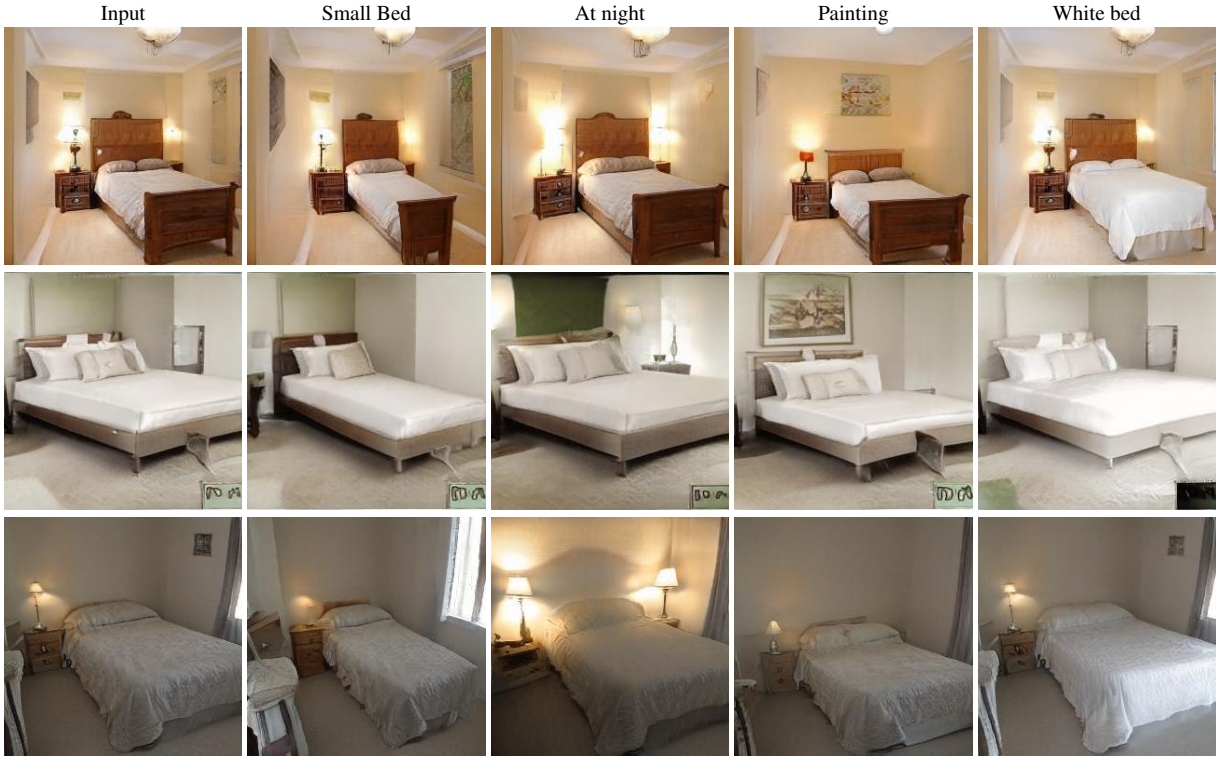


Figure 28. The LSUN Bedroom dataset contains various viewpoints, distances to camera, and bedroom styles. The StyleGAN trained on this dataset lends itself to manipulation using our global directions method, but the manipulations exhibit some entanglement.