S^2-Flow: Joint Semantic and Style Editing of Images

Supplemental Material

Krishnakant Singh [1]
krishnakant.singh@visinf.tu-darmstadt.de
Simone Schaub-Meyer[1, 2]
simone.schaub@visinf.tu-darmstadt.de
Stefan Roth[1, 2]
stefan.roth@visinf.tu-darmstadt.de

[1] Department of Computer Science, TU Darmstadt
[2] hessian.AI

A Training Details

Loss function. We describe each part of our loss function (Eq. (1) in the main paper) in detail. The negative log likelihood loss, $\mathcal{L}_{\text{nll}}$, encourages the network to learn the conditional data distribution of images over semantic masks. It is formally given by

$$\mathcal{L}_{\text{nll}}(w) = \log p(w|m) \ . \tag{2}$$

The semantic consistency loss, $\mathcal{L}_{\text{sm}}$, encourages the network to output an edited latent code $\hat{w}$ such that the mask of the generated image, $S(G(\hat{w}))$, is equal to the edited mask $\hat{m}$ provided by the user. It is formally written as

$$\mathcal{L}_{\text{sm}}(\hat{m},\ S(G(\hat{w}))) = \text{CrossEntropy}(\hat{m},\ S(G(\hat{w}))) \ . \tag{3}$$

The pixel-wise image loss, $\mathcal{L}_{\text{img}}$, encourages the network to make the edited area appear similar in style to the original image, *e. g.*, an edited hair/nose texture should be equal to the original hair/nose texture. It formally denoted as

$$\mathcal{L}_{\text{img}} = \|\hat{I} - I\|_2 \otimes M \ , \tag{4}$$

where $\otimes$ is a pixel-wise multiplication with mask M, which restricts the computation to the edited areas. The mask M is computed as the XOR product between the original mask m and the edited mask $\hat{m}$. We also tried the XNOR (not XOR) operator and found the XOR operator to perform slightly better in our experiments.

The perceptual loss, $\mathcal{L}_{\text{percept}}$, encourages the generated image $\hat{I} = \text{G}(\hat{w})$ and the original image I to be perceptually similar. We use the LPIPS [54] loss for this, which is formally given by

$$L_{\text{percept}} = \|\phi(\hat{I}) - \phi(I)\|_1 \ , \tag{5}$$

where ϕ denotes the ImageNet[59]-trained VGG [46] features.

Curriculum learning. For training, we rely on a curriculum learning approach where the loss function and the difficulty of the performed edits are increased gradually. Compared to training with the full loss right away, this stabilizes the convergence of our model. Concretely, the first 30 epochs are trained only using the negative log-likelihood loss, making the network learn the conditional data distribution. For the remaining epochs, the full loss in Eq. (1) is used; all components of the loss function are equally weighted (*i. e.* $\lambda_i = 1$).

Semantic edits are simulated during training by swapping the ground-truth semantic mask with another one from the training dataset. For a given sample and its (source) semantic mask, we categorize the difficulty of swapping it with every other (target) semantic mask in the dataset. We categorize the difficulty as a combination of two criteria, as specified in Tab. 4. First, we measure the pixel-wise accuracy between two semantic masks (mAcc). Second, we compute the landmark distance using [61], which is the distance between the facial landmarks in the corresponding images. Both these criteria help us identify which (target) semantic masks are well aligned with the given (source) sample's semantic mask. Swapping with a well-aligned (target) semantic mask creates a non-noisy semantic mask. After the initial 30 epochs, we simulate editing during training by swapping the (source) semantic mask with a (target) semantic mask from the easy category. This is done for a duration of 30 epochs. After this, we start swapping with a semantic mask from the medium category for the next 20 epochs. For the remaining 20 epochs, we swap with a semantic mask from the hard category. This strategy helps to stabilize the training process.

Category	mAcc	Landmark distance $\mathcal{D}_\text{L}$ [61]
easy	mAcc ≥ 0.8	$\mathcal{D}_\text{L} \leq 100$
medium	$0.7 <$ mAcc < 0.8	$100 < \mathcal{D}_\text{L} < 150$
hard	mAcc ≤ 0.7	$\mathcal{D}_\text{L} \geq 150$

Table 4: Difficulty classification criteria for segmentation edits during training used for curriculum training. See text for details.

B Architecture Details and Ablation

Here, we give additional details and ablation experiments, complementing the results shown in the main paper. Specifically, in Appendix B.1 we explain the architectural details of our Embedder and CNF blocks used in the proposed network architecture (Fig. 3(a)). In Appendix B.2 we give full ablation results on the importance of each component of our loss functions.

B.1 Network architecture

Embedder. The embedder is composed of 3 blocks. Each block consists of a ConvolutionBatchNormalization layer followed by a 2D Max-Pool Layer. This is followed by an MLP layer, which outputs a vector of size 19×1. We use ReLU activations after each CNN-BatchNormalization block.

CNF block. Each CNF block is made of gate-bias modulation functions called Concat-Squash functions [60]. The gate-bias modulation block consists of 3 linear layers, which modulate the output of the CNF blocks both on the input and the conditioning variable. The CNF works by solving an ODE through time, hence the CNF blocks receive time as an input. For adding the conditional input to the CNF blocks, we follow [4] and broadcast the time

Method	Perceptual Quality		Semantic		Identity
	FID (↓)	LPIPS (↓)	mIoU (↑)	mAcc (↑)	ID (↓)
$\mathcal{L}_{\text{nll}}$	26.50	0.17	0.80	0.92	0.17
$\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{sm}}$	**26.47**	0.18	**0.84**	0.93	0.19
$\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{sm}} + \mathcal{L}_{\text{img}}$	26.60	0.16	0.79	0.91	0.16
$\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{sm}} + \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{percept}}$ *(ours)*	26.75	**0.12**	0.78	**0.94**	**0.10**

Table 5: Impact of different losses on performance metrics for general edit benchmark.

dimension so that it is of the same size as the conditional input. The broadcasted time variable and conditional input are then concatenated channel-wise before being fed to the CNF blocks.

B.2 Loss ablation

From Tab. 5 we can observe that the perceptual loss function ($\mathcal{L}_{\text{percept}}$) is most helpful in terms of identity preservation and the LPIPS [54]. $\mathcal{L}_{\text{sm}}$ improves the mIoU and mAcc scores compared to $\mathcal{L}_{\text{nll}}$, indicating that the model learns to be more faithful to the edited semantic mask. Surprisingly, we find $\mathcal{L}_{\text{img}}$ helps only marginally over $\mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{sm}}$ in terms of the LPIPS and ID preservation. A possible reason for this can be the use of masking the $\mathcal{L}_{\text{img}}$ loss to only the edited regions. Our final loss function is much better on three key metrics of LPIPS, ID, and mAcc, while being slightly worse in terms of mIoU and FID scores compared to others.

C Additional Qualitative Results

We provide numerous additional visual results to show the editing capabilities of our method. In Appendix C.1, we show visual results for the task of conditional generation. Appendices C.2 to C.4 show additional qualitative results for the task of disentanglement, semantic and style editing.

C.1 Conditional generation

Fig. 9 shows that S^2-Flow is able to generate realistic-looking images that are pixel precise to the given input mask. This is in contrast to other attributed-based methods, *e. g.* StyleFlow [4], and text-based methods, *e. g.* ManiGAN [30] or TediGAN [52], which provide little control over how the generated images should look (Fig. 2). Though semantic-based methods like MaskGAN [29] and SPADE [36] are also able to perform controlled conditional generation. Fig. 11 shows results in addition to Fig. 4 for these methods, which showcase that semantic-based methods lack realism when generating images, which was also seen quantitatively in Tabs. 2 and 3.

C.2 Disentanglement

One of the key ideas of S^2-Flow is to disentangle the style and semantic spaces to provide controlled edits in both spaces. Fig. 10 shows additional results to Fig. 5, verifying the disentanglement of S^2-Flow. Fig. 10(a) shows that the S^2-Flow can generate images with diverse style while being faithful to the semantic mask. On the other hand, Fig. 10(b) shows

the results for the case when the style code is fixed and the semantic mask is changed. This is equivalent to semantically editing an image. S^2-Flow is able to generate images that are faithful to the input mask while preserving the identity and style of the input image (see Fig. 10(b)).

C.3 Semantic editing

Since S^2-Flow is primarily a semantic-based editing method, we provide several additional results on the task of semantic editing. First, Fig. 11 provides results in addition to Fig. 4 for comparing S^2-Flow against its peers, namely MaskGAN [29] and SPADE [36]. We also include a comparison against an attribute-based method, StyleFlow [4], to further show the superiority and need of semantic-based methods for controlled semantic editing. Fig. 11 shows that S^2-Flow is superior in terms of visual quality to MaskGAN and SPADE while being more faithful to the edited input mask than StyleFlow, leading to more controlled editing; this was also supported quantitatively in Tabs. 2 and 3.

We then show results for semantic editing of fake images (Fig. 12), and additional results to Fig. 8(a) for editing real images (Fig. 13) on various semantic edits like editing a smile, changing glasses, and hairstyle. We want to point out that sometimes when the edited semantic mask is noisy[1] (Fig. 12 row [6,7]-left and Fig. 13 row [3-left]), it can lead to slight changes in the background or other style attributes. However, the edited image by S^2-Flow still preserves the identity of the original image to a very high degree.

C.4 Style editing

Figs. 14 and 15 show additional results to Figs. 6(b) and 8(b) for style editing of fake and real images, respectively. S^2-Flow uses simple linear interpolation in its style space for applying style edits. It is able to apply fine edits like changing hair color and broad edits like modifying age. While making style edits, our method preserves the person's identity and only minimally changes the semantics of a given image. We want to point out that since S^2-Flow's style space is not disentangled, broad style edits like age can cause multiple attributes to change (Fig. 15 row 5 and Fig. 14 row 6). We can alleviate this issue by disentangling the style space or using non-linear interpolation methods like [24]. In Fig. 16, we show results for a style editing comparison with StyleFlow. Though S^2-Flow is never explicitly trained on style attributes, it still has comparable results with StyleFlow, a model explicitly trained with style attributes. Moreover, S^2-Flow allows for additional edits like hair color (Fig. 6(b)), which are not possible with StyleFlow. The results show that our method performs comparably to models that are designed for style editing.

D Limitations

In this section we provide visual results showing the limitations of S^2-Flow. As discussed previously, perfect disentanglement with a limited number of GAN-generated data is impossible [12, 33, 34, 39]. Fig. 17(a) shows that S^2-Flow is unable to disentangle semantic attributes like long hair with style attributes like eye and lip makeup. This can be attributed to the model being trained on a GAN-generated dataset [9], which contains no or limited examples of men with long hair, making the model associate long hair with the female gender

[1]Noisy semantic masks happen due to swapping semantic parts between images that are not well aligned.

erroneously. Many models suffer from dataset bias, requiring us to be mindful of the dataset with which we train our model. The style space of S^2-Flow is entangled and when coupled with linear interpolation, it causes multiple style attributes to change. Fig. 17(b) shows results where multiple style attributes change when performing age editing (old2young) in the style space of S^2-Flow. As stated previously, these issues can be resolved by either disentangling the style space of S^2-Flow or by using non-linear interpolation methods [24].

References

[59] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[60] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.

[61] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 2009.

Figure 9: **Conditional generation.** Given a semantic mask m, S^2-Flow can generate highly realistic-looking images, which are pixel precise to the input mask m. In comparison, this high level of controllability is not possible with attribute-based, *e. g.* StyleFlow [2], and text-based methods, *e. g.* [30, 52] (see also Fig. 2(a)).

(a) Fixed Mask (b) Fixed Style

Figure 10: **Disentanglement of semantics and style.** *(a)* **Style diversity.** S^2-Flow generates a diverse set of styles while only minimally deviating from the semantic mask. *(row [1-2])* shows age variations with the same semantic mask and *(row [3-5])* shows hair color variations while keeping the semantic mask fixed. *(b)* **Semantic diversity.** S^2-Flow is able to generate a face with varying smiles *(row [1-2])*, glasses *(row [3-4])*, and hairstyle *(row [5-6])* while keeping the style fixed and preserving identity.

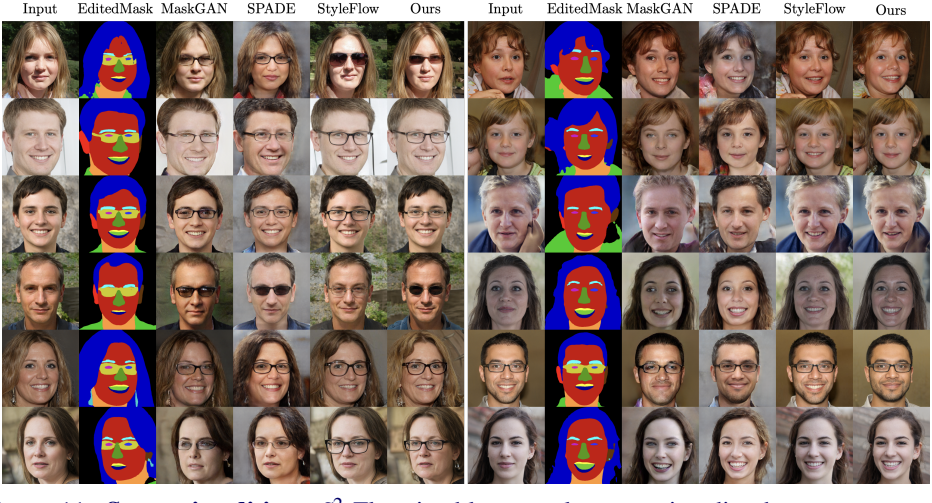

Figure 11: **Semantic editing.** S^2-Flow is able to apply semantic edits that are more controllable (this is particularly visible for adding glasses) than attributed-based methods, *e. g.* StyleFlow [4], and has higher identity preservation and realism compared to semantic-based methods, *e. g.* MaskGAN [29] and SPADE [36].

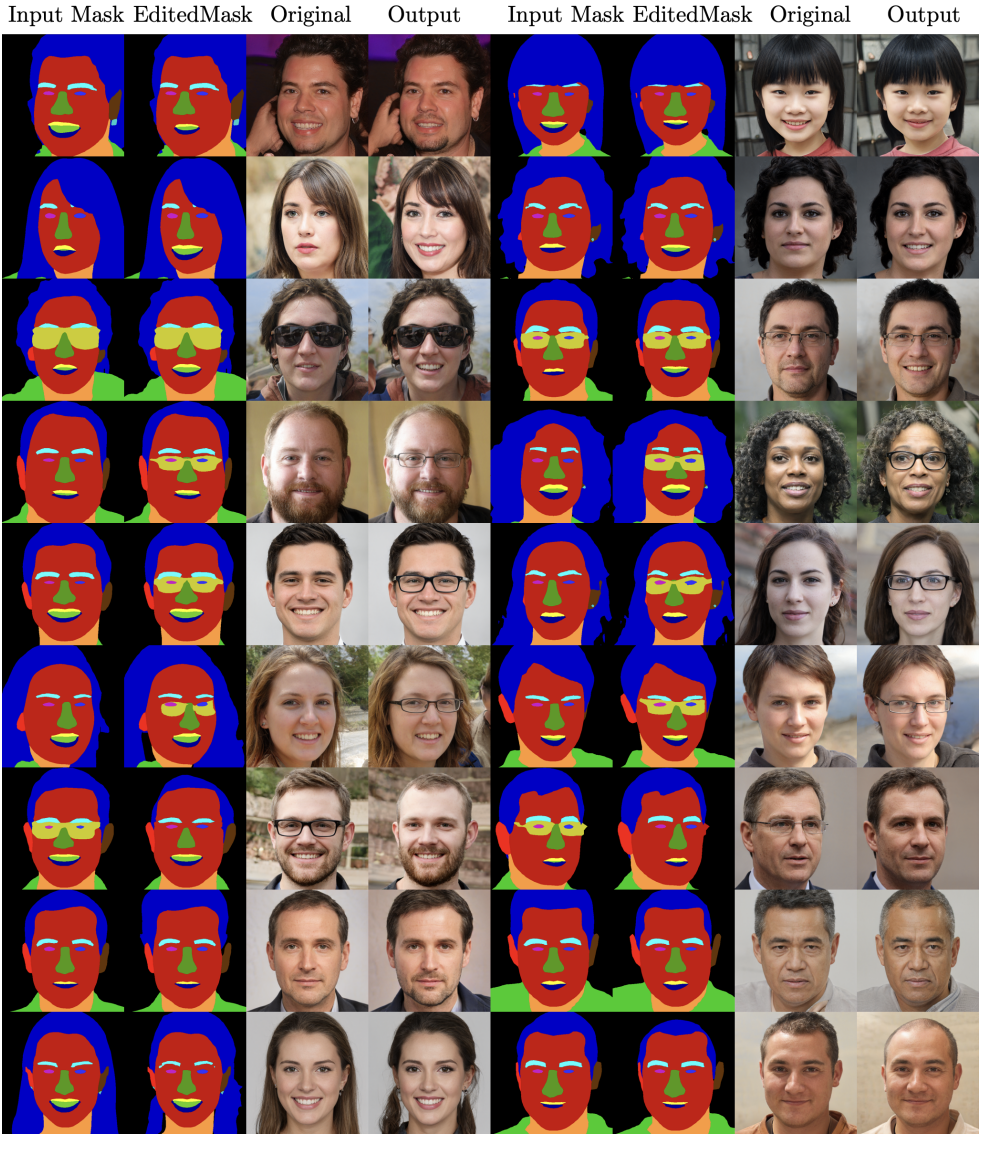

Figure 12: **Semantic editing on generated images.** Our model is able to provide a wide variety of semantic edits like smile change *(row [1-3])*, adding glasses *(row [4-5, 6-left])*, removing glasses *(row [6-right, 7])*, and changing hairstyle *(row [8-9])* on GAN-generated images. All edits are identity preserving and have a high visual realism. Even for a noisy edited mask *(row [6, 7]-left)* with multiple changes, S^2-Flow is able to make high-quality edits that are faithful to the input mask.

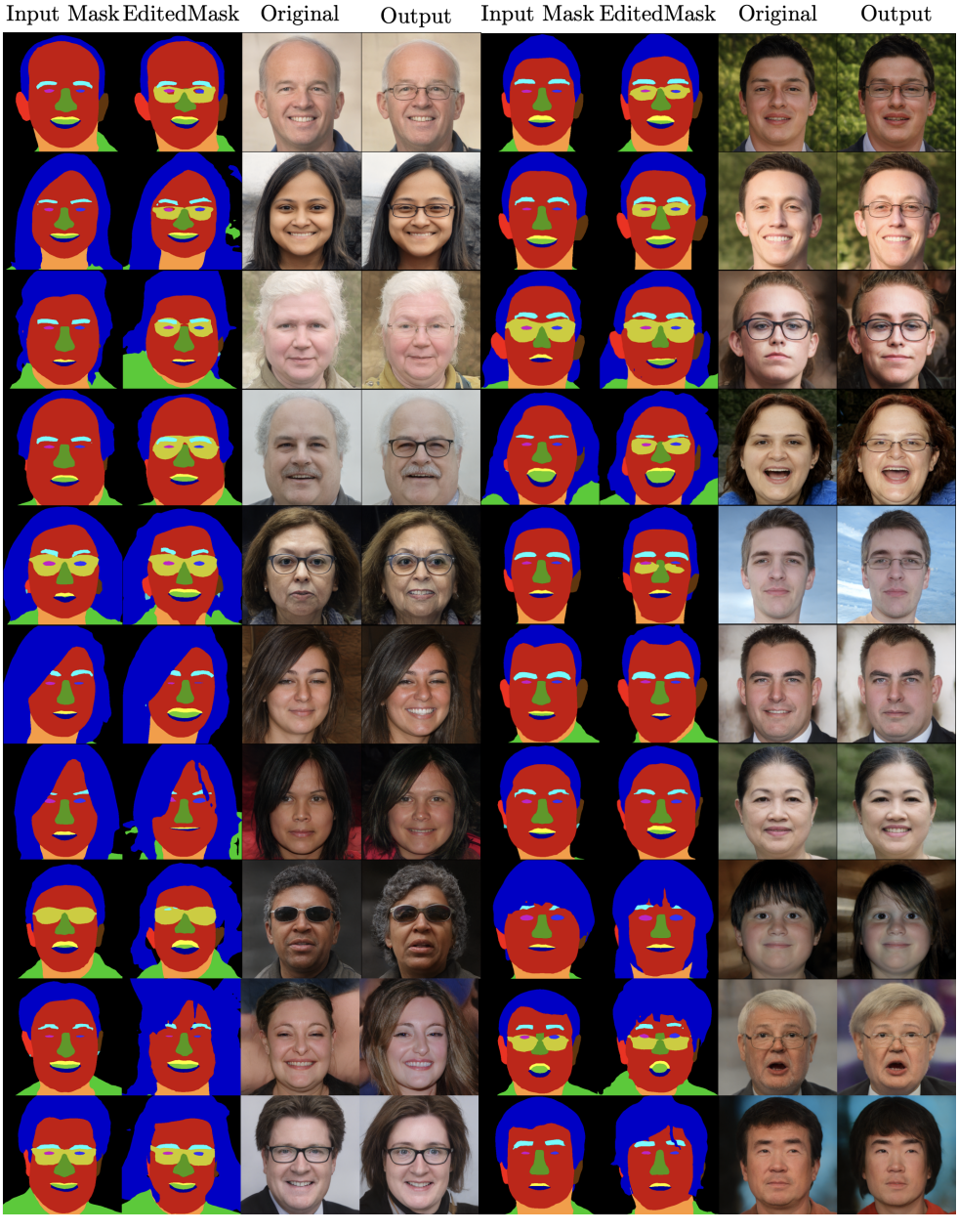

Figure 13: **Semantic editing on real images.** S^2-Flow can make a wide variety of semantic edits like adding glasses *(row [1-4, 5-right])*, smile editing *(row [5-left, 6-7])*, and hairstyle editing *(row [8-10])* on real images, even when the model is never trained on real images. Even with a noisy edited mask *(row [3-left, 5-right])*, the model is able to preserve identity while only slightly changing some style attributes (background color).

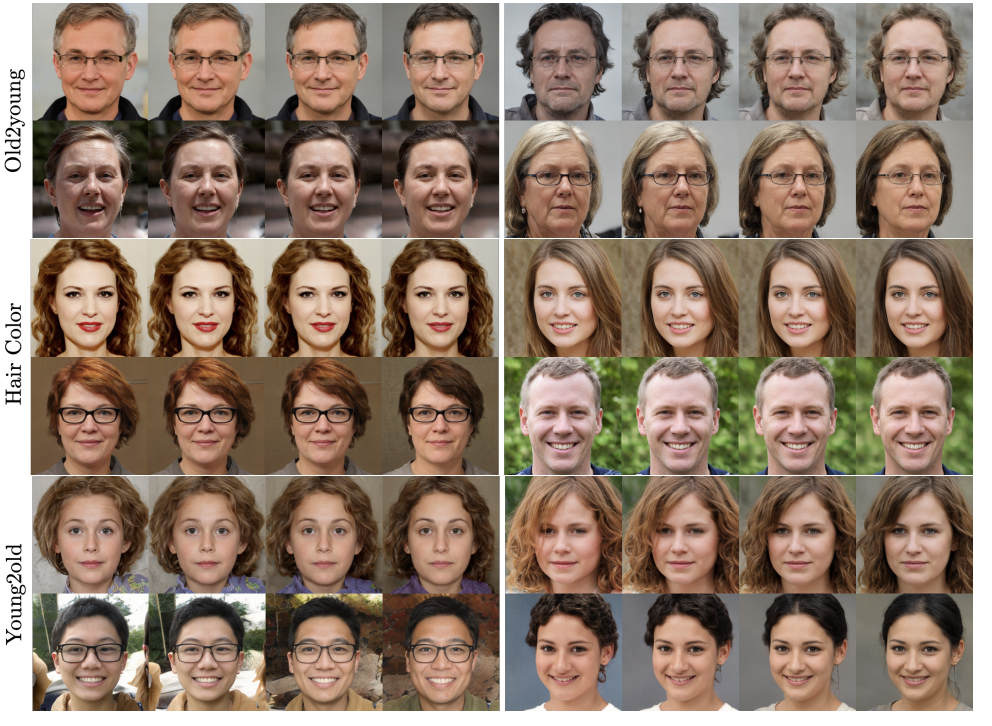

Figure 14: **Style editing of generated images.** Our model is able to apply fine style edits like hair color, as well as broad style edits like age (young2old and old2young), with high degree of identity preservation and realism. Some erroneous attributes, *e. g.* background, can change when applying style edits because the style space of S^2-Flow is not disentangled itself.

Figure 15: **Style editing of real images.** Our model is able to apply fine style edits like changing hair color, as well as broad style edits like age edits (young2old and old2young) on real images with a high degree of identity preservation, even though the model is never trained on real images.

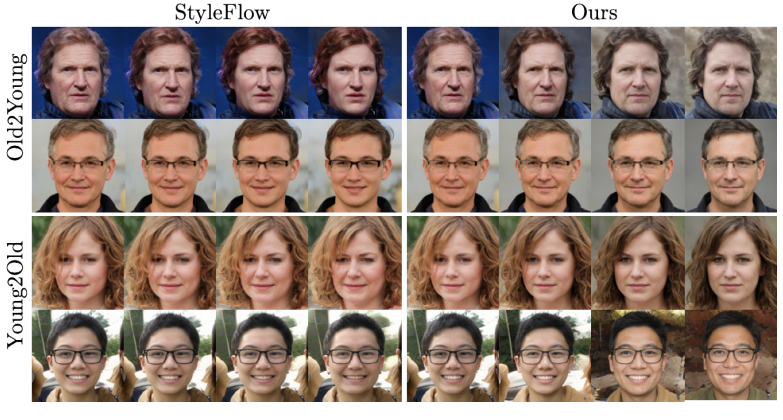

Figure 16: **Style editing comparison.** Age style editing comparison between StyleFlow and S^2-Flow. Though S^2-Flow is never trained explicity with style attributes, it is still is comparable to StyleFlow, a model specifically trained for style editing.

(a) Semantic editing – hair style change (b) Style editing – old2young

Figure 17: **Limitations.** *(a)* **Semantic and style entanglement.** S^2-Flow is unable to perfectly disentangle long hair (semantic) from attributes like lip and eye makeup (style). This can be attributed to the dataset bias as there exists no or limited samples of men having long hair in the StyleFlow [4] dataset with which S^2-Flow is trained. *(b)* **Style space entanglement.** The style space of S^2-Flow is entangled, and, when coupled with linear interpolation for performing style edits, it can cause multiple attributes like background, shirt color, *etc.* to change when applying age editing (old2young).