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

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

Figure 1: S^2-Flow is capable of applying semantic *(top right)*, style *(bottom right)*, and joint semantic and style edits *(left)* for facial images while preserving both identity and realism.

Abstract

The high-quality images yielded by generative adversarial networks (GANs) have motivated investigations into their application for image editing. However, GANs are often limited in the control they provide for performing specific edits. One of the principal challenges is the entangled latent space of GANs, which is not directly suitable for performing independent and detailed edits. Recent editing methods allow for either controlled style edits *or* controlled semantic edits. In addition, methods that use semantic masks to edit images have difficulty preserving the identity and are unable to perform controlled style edits. We propose a method to disentangle a GAN's latent space into semantic and style spaces, enabling controlled semantic *and* style edits for face images independently within the same framework. To achieve this, we design an encoder-decoder based network architecture (S^2-Flow), which incorporates two proposed inductive biases. We show the suitability of S^2-Flow quantitatively and qualitatively by performing various semantic and style edits. Code and data are available at https://github.com/visinf/s2-flow.

1 Introduction

Powerful deep generative models of images, such as generative adversarial networks (GANs) [13] or variational autoencoders (VAEs) [26], have seen numerous applications across com-

puter vision [8, 9, 35, 38, 55]. With the advent of models based on StyleGAN [22], there has been a plethora of work focusing on controllable manipulation of the latent code for the task of image editing [9, 14, 45]. However, modifying the latent code in a controllable way, such that it leads to the desired edits in the image space, continues to be challenging.

Broadly speaking, we can divide image editing with GANs into two subgroups: *(i) Unconditional GAN-based methods* [14, 51], which find editing vectors using unsupervised learning methods like PCA [14] or activation maps [51]. They do not take user inputs into account and have a limited set of editing directions. *(ii) Conditional GAN-based methods*, on the other hand, are more cognizant to the user input. These methods generate the edited image conditioned on user inputs, such as semantic masks [29, 36, 57], attributes [15, 18], or text [30, 37, 52]. Though these methods support more varied editing operations, they lack controllability, *e. g.*, attributed-based methods provide no controllability on how a smile (wide, grinning, *etc.*) might look or if the person is wearing round or square glasses (Fig. 2(a)). Semantic-based methods, in contrast, have limited control on style editing, requiring the use of a target transfer image (Fig. 2(b)). In general, performing controlled and disentangled edits in the latent space is a very challenging task.

In this paper, we propose, to the best of our knowledge, the first approach that allows to perform controlled semantic editing (*i. e.* changes possible with a semantic mask, *e. g.*, changing smile, changing hair style, *etc.*) *and* style editing (*i. e.* changes *not* possible with a semantic mask, *e. g.*, age, hair color, *etc.*) while preserving the identity of facial images. We achieve this by disentangling the semantic and style spaces. This disentanglement is achieved by introducing two inductive biases into the network: *(1)* ***Style consistency*** – editing an image in the semantic domain should have no effect on the style properties of the image. *(2)* ***Semantic consistency*** – edits made in the semantic domain should be reflected in the semantics of the generated image. Our design of the model architecture and a novel loss formulation allow the model to incorporate the aforementioned inductive biases, helping it to make independent edits to the style and semantics of a given image. Specifically, our contributions are: *(i)* We propose a method to disentangle the latent space of a pretrained generator network into style and semantic spaces for the task of facial editing. Thereby, we are among the first to utilize normalizing flows for disentangling a GAN's latent space. *(ii)* Our method solves the problem of applying fine-grained edits to both the style *and* semantic spaces. *(iii)* We show both qualitatively and quantitatively that our model outperforms well established methods [29, 36] on two semantic editing benchmarks. *(iv)* We show our model's ability to generate high-quality identity-preserving edits for various editing tasks, see Fig. 1.

2 Related Work

Since the seminal work of Goodfellow *et al.* [13], there has been enormous progress in the area of generative modelling, from generating small-scale low-resolution images [41] to generating high-fidelity, realistic looking images [21, 22]. In this work we tackle the problem of disentangling the latent space of a pretrained GAN, focusing on facial image editing.

Disentanglement. Unsupervised disentanglement has been a long standing goal for computer vision for its usefulness in inverting the generative process. InfoGAN [8] and BetaVAE [17] tackle this problem from an information theoretic perspective. Despite their pioneering efforts, these models work only in low-resolution low-complexity dataset settings and are difficult to train. Many works [19, 32] disentangle the style and semantic codes of an image by swapping the codes with another image for the the task of image-to-image

translation. Instead of swapping style and semantic codes, we obtain a disentanglement between these spaces by incorporating transformation-based inductive biases like semantic and style consistency. Recently, [28, 53] proposed disentangled variants of StyleGAN [23] by using two separate (style and semantic) spaces and modelling the interactions between them using attention modules. In contrast, our work deals with disentangling the latent space of a pretrained GAN and, thus, requires no additional training of the GAN module, which is difficult and computationally expensive.

Editing in the latent space of GANs. There have been a plethora of interesting works for editing using the latent space of GANs. Attribute-based methods rely on learning interpretable edit directions in the latent space of a pretrained GAN using a neural network trained either on attributes [9, 24, 45, 58], paired synthetic data [49], or pseudo labels [20]. Some attribute-based methods like AttGAN [15] train a generative model based on an attribute classification loss. Text-based methods use natural language cues for editing the image; Patashnik et al. [37] apply a CLIP-based [42] loss for learning editing directions. [30, 52] learn a GAN-based model conditioned on text. All the above methods provide a high degree of control for style edits but severely lack controllability and interpretability in terms of editing semantics, *e. g.*, sunglasses *vs.* reading glasses, small smile *vs.* big smile, *etc.* (Fig. 2(a)). A fastidious user can give very detailed textual cues to obtain the desired effect, but this soon becomes untenable when finer control of semantics is required, *e. g.*, when describing the exact shape of glasses. On the other hand, methods like [9, 14, 44, 50, 51] use unsupervised approaches for finding editing directions. These methods find a set of editing directions and require copious manual effort to semantically identify them. Semantic-based editing methods [29, 36, 57] provide control over semantic editing but these models severely lack in the ability to carry out style edits. For carrying out targeted style edits, the user has to search through 1000s of images to find a suitable target (Fig. 2(b)). An overview of such work is shown in Tab. 1.

	Attribute				Semantic				Uncondit.				Text			Ours
	StyleFlow	InterfaceGAN	LatentNeuralODE	AttGAN	SPADE	SEAN	MaskGAN	EditGAN	Stylespace	GANSpace	Steerability	GANInverse.	StyleClip	TediGAN	ManiGAN	S^2-Flow
Style Edits	✓	✓	✓	✓	○	○	○	×	○	○	○	○	✓	✓	✓	✓
Structural Edits	○	○	○	○	✓	✓	✓	✓	○	○	○	○	○	○	○	✓
Mult. Edits / Model	✓	×	×	✓	✓	✓	✓	○	–	×	×	–	✓	✓	✓	✓
Pretrained GAN	✓	✓	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×	×	✓

✓: Fully, ×: None, ○: Limited (requires style image), –: N/A

Table 1: Comparison of existing editing methods. Our work sits between attribute and semantic methods, allowing for both fully controlled style *and* semantic edits.

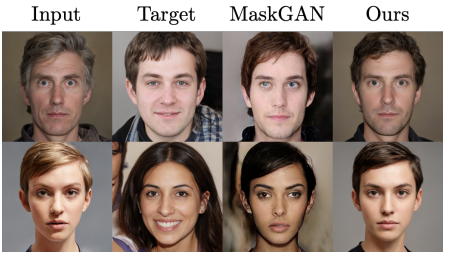
Some attribute-based methods like AttGAN [15] train a generative model based on an attribute classification loss. Text-based methods use natural language cues for editing the image; Patashnik et al. [37] apply a CLIP-based [42] loss for learning editing directions. [30, 52] learn a GAN-based model conditioned on text. All the above methods provide a high degree of control for style edits but severely lack controllability and interpretability in terms of editing semantics, *e. g.*, sunglasses *vs.* reading glasses, small smile *vs.* big smile, *etc.* (Fig. 2(a)). A fastidious user can give very detailed textual cues to obtain the desired effect, but this soon becomes untenable when finer control of semantics is required, *e. g.*, when describing the exact shape of glasses. On the other hand, methods like [9, 14, 44, 50, 51] use unsupervised approaches for finding editing directions. These methods find a set of editing directions and require copious manual effort to semantically identify them. Semantic-based editing methods [29, 36, 57] provide control over semantic editing but these models severely lack in the ability to carry out style edits. For carrying out targeted style edits, the user has to search through 1000s of images to find a suitable target (Fig. 2(b)). An overview of such work is shown in Tab. 1.

Simultaneous edits of semantic and style. Our work is able to provide controlled semantic and style editing on facial images by finding disentangled dual spaces. Unlike [23, 28, 53], we work on a pretrained GAN's latent space instead of training a new GAN model from scratch, which requires a large training set and extensive parameter tuning. Our method of disentanglement uses only 10k GAN-generated images. Moreover, it lies at the intersection of style and semantic-based methods, using semantic conditioning for enabling semantic edits and learning walks in the style space for applying highly controllable style edits.

3 Joint Semantic and Style Editing

Given *only* a pretrained GAN model and its generated images as input, our goal is to devise a method for image editing that enables semantic and style edits within the *same* framework without mutual interference between these edits. We argue that this requires disentangling

(a) Controlled semantic editing



(b) Controlled style editing

Figure 2: *(a)* **Controlled semantic editing.** Attribute-based methods, *e. g.* StyleFlow [4], do not allow for user control over how a targeted semantic edit should look like. *(b)* **Controlled style editing.** Semantic methods, *e. g.* MaskGAN [29], allow for limited controllability for style editing. As semantic-based methods rely on target images for style editing, erroneous attributes like lip makeup/skin color are also transferred when the user only aimed to change the the age (*top*) or hair color (*bottom*). In contrast, S^2-Flow does not require target images and uses interpolation in style space, enabling it to apply targeted style edits.

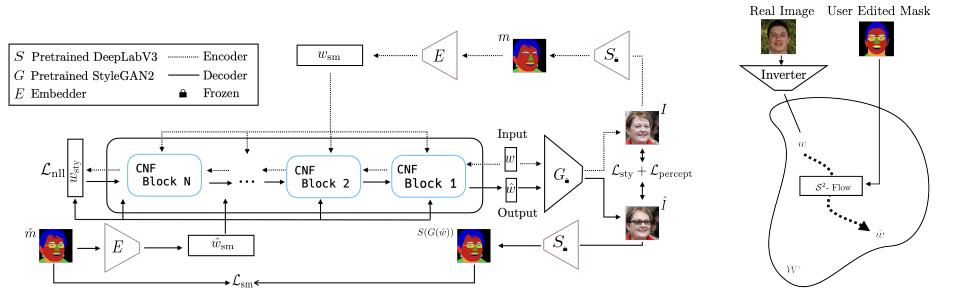
the image representation into two parts, one responsible for the semantics of the image and one for capturing its style. This disentanglement allows us, on one hand, to perform edits in multiple spaces, namely in the semantic and style space, while on the other giving more control on the generated result by ensuring the integrity of the non-edited characteristics.

The key insight of our method is that an image can be decomposed into its semantic and the style codes and edits made in one domain (semantic or style) should not affect the other. We design our image generation and editing framework as an encoder-decoder model to disentangle the aforementioned two spaces using continuous normalizing flow (CNF) [7] blocks. We chose CNFs based on two motivations: *(i)* Firstly, a CNF network is reversible by design and hence has cycle consistency, which is a crucial property to successfully disentangle semantics and style during training. *(ii)* Secondly, CNFs are much easier to train than other encoder-decoder models like VAEs [17, 26] or Transformers [43]. Before describing our architecture in Sec. 3.2, we summarize its building blocks.

3.1 Building blocks

Latent space. The latent space of deep generative models serves as a good proxy for the real image manifold. We use the latent space of StyleGAN2 [22] for our model. Given a latent sample, drawn from $\mathcal{N}(0,I)$, StyleGAN2 transforms it into an intermediate latent code using a series of nonlinear mappings. Abdal et al. [1] further extend this space by concatenating 18 different latent codes, which they term the $\mathcal{W}^+$ space. We train our network directly in this $\mathcal{W}^+$ space, since [2, 4] show that this space is better suited for editing.

Normalizing flows. A normalizing flow model [11] transforms a simple initial known distribution to a more complex one using a series of composable transformations $f_1,\ldots,f_k$. The function $F = f_1 \circ \cdots \circ f_k$ must be invertible and both F and F^{-1} should be differentiable. The function F relates the marginal densities of the two distributions using the change of variables formula, which involves computing the determinant of the Jacobian. Calculating the determinant of the Jacobian can be an expensive operation, requiring special neural network architectures for fast computation. Chen et al. [7] introduced a continuous version to alle-



(a) Training (b) Editing a real image

Figure 3: **Framework overview.** *(a)* **Training.** (1) The encoder *(dashed line)* takes as input the latent code $w \in \mathcal{W}^+$ and its corresponding inferred semantic mask $m = S(G(w))$. (2) The encoder model disentangles the style code w_{sty} from w given $w_{\text{sm}} = E(m)$. (3) The decoder *(solid line)* combines w_{sty} and the edited semantic code $\hat{w}_{\text{sm}} = E(\hat{m})$ to yield the edited latent code $\hat{w} \in \mathcal{W}^+$ (output), which is fed to the generator network to yield the edited image $\hat{I}$. *(b)* **Editing.** Given a real image, we first obtain its latent code w using an inverter network, *e. g.*, e4e [47]. Using this w and the given user-edited mask, S^2-Flow returns a new latent code $\hat{w}$, which is used to generate the edited image $G(\hat{w})$.

viate this problem, paving the way for using arbitrary neural networks for modelling these transformations. We decided to use CNFs [7] for our implementation as this allows us to use an unrestricted architecture when modelling the transformation function F, making the transformation function more flexible and expressive.

3.2 S^2-Flow model

Overview. The goal of S^2-Flow is to disentangle the latent code w into its constituent latent codes, namely, semantic w_{sm} and style w_{sty}. We use a conditional variant of CNF [7], where the forward and reverse flow[1] are used to model the decoder and encoder, respectively. In contrast to earlier works [4, 24], where the latent space of the CNF model is highly entangled, our formulation disentangles style and semantics using our novel inductive biases (Sec. 3.3). The process of encoding-decoding is visualized in Fig. 3(a) and formally described below.

Encoder. The reverse flow of our model is the encoder network and works as follows: Given a latent code w, we generate its corresponding image I and infer its semantic mask m using pretrained StyleGAN2 [22] and DeepLabV3 [6] networks, respectively. The semantic mask is then passed through a convolutional neural network called Embedder to yield the semantic code w_{sm}. The concatenation of w and w_{sm} is fed as input to each CNF block of our model. Given this semantic code w_{sm}, the encoder learns to disentangle the style code w_{sty} from the latent code w.

Decoder. Our forward flow learns to combine the latent style (w_{sty}) and semantic (w_{sm}) codes to reconstruct the original latent code w without any loss of information due to the reversibility of the CNF. During training (Fig. 3(a)), we simulate the editing behaviour by modifying the input mask m to $\hat{m}$, which results in a new $\hat{w}_{\text{sm}}$, leading to an edited latent code $\hat{w}$ and its corresponding edited image $\hat{I}=G(\hat{w})$. Editing during training is performed by

[1] CNFs define forward flow as transforming a normal distribution to a more complex one. The reverse flow transforms a complex distribution to the normal distribution.

swapping the mask between two samples based on our defined criterion, see supplemental for more details.

3.3 Objective

To disentangle style from semantics for the $\mathcal{W}^+$ space during training, we add two inductive biases to our loss function: *(i) Style consistency* ensures that editing the semantic mask of an image should have minimal impact on the style attributes. Hence at training time, the decoder should output a latent code $\hat{w}$ that is similar in style to the original w. In other words, S^2-Flow learns to extract the same style code w_{sty} for two images that only differ semantically. *(ii) Semantic consistency* addresses the fact that changes made in the semantic domain should be reflected in the generated image as well. Our overall loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{nll}}(w) + \lambda_1 \mathcal{L}_{\text{sm}}(\hat{m}, S(G(\hat{w}))) + \lambda_2 \mathcal{L}_{\text{img}}(I, \hat{I}) + \lambda_3 \mathcal{L}_{\text{percept}}(I, \hat{I}) \; . \tag{1}$$

The negative log-likelihood loss, $\mathcal{L}_{\text{nll}}$, encourages the model to learn the conditional data distribution of images given semantic masks. To ensure semantic consistency, we use the $\mathcal{L}_{\text{sm}}$ loss, which is equal to the cross-entropy loss between the edited mask $\hat{m}$ and the inferred mask of the generated image $S(G(\hat{w}))$. To ensure style consistency between the edited image $\hat{I}$ and the original image I, we use two loss functions, $\mathcal{L}_{\text{img}}$ and $\mathcal{L}_{\text{percept}}$. $\mathcal{L}_{\text{img}}$ measures the L_2 distance in the image space. We use a masked version of the L_2 distance, restricting the computation of the loss only to the edited regions. The perceptual loss, $\mathcal{L}_{\text{percept}}$, computes the L_1 distance between $\hat{I}$ and I using the intermediate features from a pretrained VGG network [46]. The detailed formulas can be found in the supplemental.

3.4 Editing and generation

After training, S^2-Flow can be used for both conditional image generation and editing.

Conditional generation. Given a semantic mask m and a style code w_{sty} from the style space of S^2-Flow, the decoder generates w and consequently a new image I=$G(w)$ consistent with the semantic mask m.

Editing. Given an edited mask for an image I (real or fake) and the latent code[2] w, the encoder disentangles w into w_{sty} and w_{sm}. For semantic editing, the decoder uses the edited mask and the original style code w_{sty} to create the edited latent code in $\mathcal{W}^+$. Fig. 3(b) shows an illustrative example of editing a real image. Style editing is performed by linearly interpolating between the source style code w_{sty} and the target style code in the style latent space. The target style code equals the mean style code of all positive samples for the given target attribute. Semantic and style edits can also be applied simultaneously, see Fig. 1.

4 Experiments

To evaluate the disentangled editing ability of our approach, we perform various facial editing experiments by applying style and semantic edits separately and in combination. We compare the results visually and quantitatively to related editing methods [4, 29, 36].

Dataset and training. The main goal of our work is to disentangle the latent space of a pretrained GAN by training our model on GAN-generated images. In particular, we use

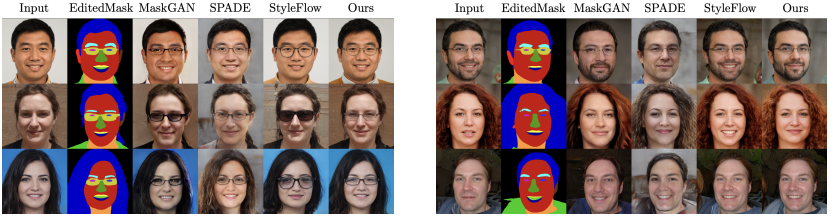
[2] For a real image, we use e4e [47] to obtain the corresponding latent code.

Method	Perceptual Quality FID ↓	LPIPS ↓	Semantic mIoU ↑	mAcc ↑	Identity ID ↓
MaskGAN [29]	40.58	0.27	0.52	0.88	0.53
SPADE [36]	60.43	0.28	0.81	0.97	0.46
S^2-Flow *(ours)*	26.65	0.14	0.77	0.95	0.15
Abs. improv.	+13.93	+0.13	-0.04	-0.02	+0.31

Table 2: Results on the smile edit benchmark from [29]

Method	Perceptual Quality FID ↓	LPIPS ↓	Semantic mIoU ↑	mAcc ↑	Identity ID ↓
MaskGAN [29]	40.79	0.29	0.58	0.82	0.50
SPADE [36]	61.02	0.30	0.89	0.92	0.46
S^2-Flow *(ours)*	26.75	0.12	0.78	0.94	0.10
Abs. improv.	+14.04	+0.17	-0.11	+0.02	+0.36

Table 3: Results on our general editing benchmark



(a) Glasses editing (b) Smile editing

Figure 4: **Comparison on semantic editing.** S^2-Flow successfully supports semantic edits while preserving the identity and yields higher quality images compared to existing semantic editing methods (MaskGAN [29], SPADE [36]). Also, S^2-Flow allows for more controlled user edits compared to existing attribute-based editing method (StyleFlow [4]).

the dataset introduced by StyleFlow [4], which consists of 10k latent codes from a StyleGAN2 [22] model trained on FFHQ [22]. For training we use the Adam [25] optimizer with a constant learning rate of $3 \cdot 10^{-4}$. Further, we rely on a curriculum learning approach where the loss function and the difficulty of the performed edits in the segmentation mask are gradually increased. We refer the reader to the supplementary for more training details.

Metrics. For measuring the structural similarity between the edited semantic mask and the semantic mask of the generated image, we use the mean IoU (mIoU) and the mean pixel-wise accuracy (mAcc). To compare the quality of generated images between different models, we use FID [16] and LPIPS [54]. We use the ArcFace [10] network to measure the identity preservation score (ID) when editing an image.

GAN inversion. For editing real images, our model makes use of GAN inversion methods [1, 2, 43, 47, 56]; specifically, we use [47] to obtain the latent code of a real image.

4.1 Quantitative experiments

S^2-Flow is a semantic-based method, which additionally allows for style editing, see Tab. 1. Thus, we compare our method against other semantic editing methods, namely MaskGAN [29] and SPADE [36], for the task of semantic editing. Both these models perform highly controllable edits conditioned on semantic masks. We do not compare our method against EditGAN [31] as it requires additional test-time optimization for each image for each editing direction; therefore, the same editing vector is not applicable to multiple images. To quantitatively measure the editing capability, we use the smile edit benchmark introduced by MaskGAN [29]. For fair comparison, we train SPADE [36] also on the StyleFlow [4] dataset using the official repository. For MaskGAN [29], we use the pretrained weights from the network trained on the CelebA-HQ dataset [21] provided by the authors. Both the CelebA-HQ and the StyleFlow dataset contain high-quality, diverse face images; hence, the test time dis-

(a) Fixed style

(b) Fixed semantics

Figure 5: **Disentanglement of style and semantics.** *(a)* **Semantic diversity.** S^2-Flow is able to generate a face with varying smiles, hairstyles, and glasses while preserving the style and identity. *(b)* **Style diversity.** S^2-Flow generates a diverse set of styles while only minimally deviating from the input semantic mask.

tribution shift should be minimal. We report the results in Tab. 2. Our model outperforms both semantic editing methods in terms of ID, LPIPS, and FID while being minimally worse in terms of mIoU and mAcc scores. We also evaluate on a more complex editing scenario in which we make diverse edits to the semantic masks. For each test image we randomly perform one of the following edits: *(1)* swap mouth, *(2)* swap nose, *(3)* swap eyebrows, *(4)* remove glasses, *(5)* swap/add glasses, and *(6)* swap hair. We refer to this as the general semantic editing benchmark; the results are shown in Tab. 3. S^2-Flow again outperforms all other baselines in terms of ID, LPIPS, and FID. Our model's slightly inferior mIoU and mAcc scores can be attributed to its behaviour of making more realistic and conservative edits to preserve the identity rather than only optimizing for the mIoU score (Fig. 4).

4.2 Qualitative results

Disentanglement of style and semantics. We show qualitatively that our model learns to disentangle style and semantics when generating or editing an image by keeping one dimension fixed while varying the other. Given a semantic mask, we randomly sample 4 different style codes from the style latent space of S^2-Flow and generate their corresponding images (Fig. 5(b)), which are diverse in style but consistent with the input semantic mask. Similarly, given an image, we obtain its style code and apply different edits to its semantic mask to generate images with the same style but different semantics (Fig. 5(a)). Both results in Fig. 5 show that our model has learned to disentangle the semantic and style codes for an image. Though we achieve a high degree of disentanglement, some factors still remain entangled between the semantic and style spaces, such as long hair and gender; this can be explained as we train with only GAN-generated images. Existing GAN-based approaches are known to not cover all modes of the underlying data distribution [12, 33, 39]. Perfect disentanglement, even with real images, remains challenging due to the inherent bias of the datasets [27] and the sample complexity [34].

Semantic editing. We evaluate the semantic editing capability of our model qualitatively against MaskGAN [29] and SPADE [36]. We also compare our method against StyleFlow [4] to show that attribute-based methods are unable to apply controlled semantic editing. We do not compare our model against unconditional and text-based models since the former lack interactive editing and the latter require very targeted text for highly controlled semantic editing. Fig. 4 clearly shows that our model is much better in terms of visual quality and identity preservation compared to previous semantic editing methods (MaskGAN [29], SPADE [36]).

(a) Joint style and semantic editing (b) Style editing

Figure 6: *(a)* **Semantic and style editing.** S^2-Flow enables diverse semantic and style edits like adding a smile (sem) & changing hair color (sty), adding glasses (sem) & changing gender (sty), *etc. (b)* **Style editing.** S^2-Flow is able to modify style attributes like hair color, age, and gender while preserving the identity and being faithful to the input semantics.

Fig. 4 also shows that attribute-based methods (StyleFlow [4]) allow for some semantic edits like smile and glasses, but without controllability over their shape.

Style editing. Fig. 6(b) shows the results of S^2-Flow for different style edits by interpolating in the style latent space. Our model can apply fine-grained style edits like changing hair color and more general edits like changing gender and age while staying truthful to the semantic mask. Since our style space itself is not disentangled by design, multiple attributes can change during interpolation. Doing similar edits with other semantic methods via style transfer requires manually searching for a target image differing in only one attribute.

Semantic and style editing. We highlight the flexibility of our model in performing edits in multiple spaces, *i. e.* style *and* semantic space. Fig. 6(a) shows that even when we edit multiple attributes in the semantic space and then perform an edit in the style space, our model is still able to preserve the identity and has excellent visual fidelity. Our work is one of the first to allow for joint editing of style and semantics with a high level of control, compared to attribute-based methods that only allow for controlled style edits (Fig. 2(a)) and semantic-based methods that only for controlled semantic edits (Fig. 2(b)).

Sequential semantic editing. We next show the capability of S^2-Flow on a more extended task that involves using edited latent vectors from the previous edit to make the next edit. This task is considerably harder since the resultant edited latent vector may not be amenable for further editing [47]. Fig. 8(c) shows sequential semantic edits like adding glasses, changing the hairstyle, removing glasses, and editing a smile. The results clearly show that S^2-Flow produces latent codes, which can be edited further. The identity and realism of the input image are preserved even in the case of long-range sequential edits.

Diverse semantic edits. Fig. 7 shows the diverse semantic editing capabilities of S^2-Flow for tasks like face frontalization, gaze change, *etc.* Even for the difficult case of randomly swapping the masks with another image, which can lead to multiple semantic changes at once, S^2-Flow preserves the identity of the input while being faithful to the edited mask.

Real image editing. Finally, we show that even though our model is trained on generated images, it can edit real images in both semantic (Fig. 8(a)) and style spaces (Fig. 8(b)). We use the e4e [47] model to embed the real images into the latent space of StyleGAN2 before editing the images using their corresponding latent vector.

Limitations. S^2-Flow is able to disentangle the style and semantics of a given image only to a certain degree. This can mainly be attributed to the fact that present GAN-based methods do not cover all the modes of the underlying data distribution [12, 33, 39]. Even when trained with real images, perfect disentanglement would require an exponential number of

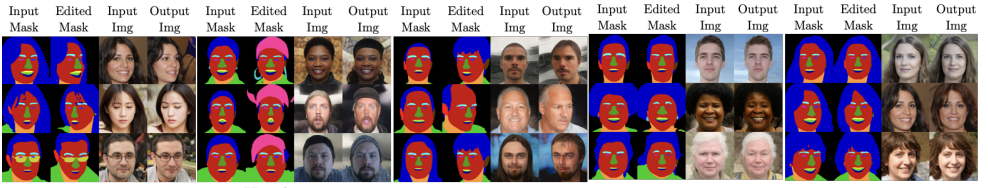
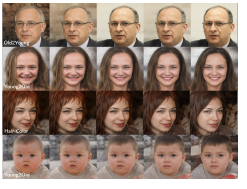
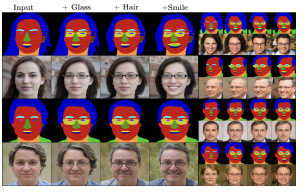


Figure 7: **Diverse semantic editing.** S^2-Flow is capable of a wide variety of semantic edits like changing gaze, changing eyebrows, adding headgear, face frontalization, and random swapping of semantic masks.

(a) Semantic editing on real images



(b) Style editing on real images



(c) Sequential semantic editing

Figure 8: *(a)* **Semantic editing on real images**. S^2-Flow is able to apply semantic edits like changing hairstyle, adding smile, and adding glasses on real images. *(b)* **Style editing on real images.** Our model allows for applying fine style edits like hair color change as well as coarse edits like changing age on real images. *(c)* **Sequential semantic editing.** S^2-Flow is able to perform sequential editing. It outputs latent editing codes, which are amenable for further edits, and the edited images are both identity preserving and realistic.

samples in the number of factors of variation [34]. Also, our model sometimes changes the background while performing style editing, which is mainly due to our style space itself not being disentangled. A simple solution is to use a post-processing optimization step like foreground-background separation or Poisson blending [40]. An exciting future direction would be to disentangle the style space itself. This would further increase controllability. Moreover, our model only allows for orthogonal changes in the semantic and style attributes. Conflicting semantic and style changes like adding a smile on the mask and then interpolating towards the sad attribute would preserve the semantic mask, only allowing for changes like squinting of eyes, which are orthogonal to semantic changes. How to better handle these conflicting cases remains an open question.

5 Conclusions

We propose S^2-Flow, a method to disentangle the latent space of a pretrained generative model to enable facial editing in multiple spaces, namely semantic and style. Our novel model design and inductive biases (*semantic* & *style* consistency) help us to achieve this disentanglement. We demonstrate visually and quantitatively that our model outperforms existing semantic-based editing methods while also adding controlled style editing capabilities to these models. Further, we illustrate the advantage of semantic editing compared to attribute-based editing.

6 Acknowledgments and disclosure of funding

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866008). The project has also been supported in part by the State of Hesse through the cluster projects "The Third Wave of Artificial Intelligence (3AI)" and "The Adaptive Mind (TAM)".

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[3] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Labels4Free: Unsupervised segmentation using StyleGAN. In *Proceedings of the Eightteenth IEEE International Conference on Computer Vision*, 2021.

[4] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics*, 40(3):1–21, 2021.

[5] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Video2StyleGAN: Disentangling local and global variations in a video. *arXiv:2205.13996 [cs.CV]*, 2022.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the 15th European Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2018.

[7] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 2016.

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[11] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[12] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *arXiv:1701.00160 [cs.LG]*, 2016.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering interpretable GAN controls. In *Advances in Neural Information Processing Systems*, 2020.

[15] 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*, 28(11):5464–5478, 2019.

[16] 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*, 2017.

[17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[18] Xianxu Hou, Xiaokang Zhang, Hanbang Liang, Linlin Shen, Zhihui Lai, and Jun Wan. GuidedStyle: Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks*, 145:209–220, 2022.

[19] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the 15th European Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2018.

[20] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *Proceedings of the Eighth International Conference on Learning Representations*, 2020.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proceedings of the Sixth International Conference on Learning Representations*, 2018.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[23] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser Nasrabadi. Style and content disentanglement in generative adversarial networks. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[24] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. Latent transformations via NeuralODEs for GAN-based image editing. In *Proceedings of the Eightteenth IEEE International Conference on Computer Vision*, 2021.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[26] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

[27] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[28] Gihyun Kwon and Jong Chul Ye. Diagonal attention and style-based GAN for content-style disentanglement in image generation and translation. In *Proceedings of the Eightteenth IEEE International Conference on Computer Vision*, 2021.

[29] 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*, 2020.

[30] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. ManiGAN: Text-guided image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. EditGAN: High-precision semantic image editing. In *Advances in Neural Information Processing Systems*, 2021.

[32] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*, 2019.

[33] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[34] F. Locatello et al. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[35] Jiteng Mu, Shalini De Mello, Zhiding Yu, Nuno Vasconcelos, Xiaolong Wang, Jan Kautz, and Sifei Liu. CoordGAN: Self-supervised dense correspondences emerge from GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[37] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the Eighteenth IEEE International Conference on Computer Vision*, 2021.

[38] William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei A Efros, and Eli Shechtman. GAN-supervised dense visual alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[39] Sen Pei, Richard Yi Da Xu, Shiming Xiang, and Gaofeng Meng. Alleviating mode collapse in GAN via diversity penalty module. *arXiv:2108.02353 [cs.CV]*, 2021.

[40] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

[41] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2016.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[43] 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*, 2021.

[44] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[45] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, 44(4):2004–2018, 2022.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[47] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics*, 40(4): 1–14, 2021.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[49] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. StyleGAN2 distillation for feed-forward image manipulation. In *Proceedings of the 16th European Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2020.

[50] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[51] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for StyleGAN image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[52] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[53] Yanbo Xu, Yueqin Yin, Liming Jiang, Qianyi Wu, Chengyao Zheng, Chen Change Loy, Bo Dai, and Wayne Wu. Transeditor: Transformer-based dual-space GAN for highly controllable facial editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[54] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[55] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[56] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *Proceedings of the 15th European Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2020.

[57] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[58] Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G. Schwing. Enjoy your editing: Controllable GANs for image editing via latent space navigation. In *Proceedings of the Eighth International Conference on Learning Representations*, 2021.