StyleFusion: Disentangling Spatial Segments in StyleGAN-Generated Images

OMER KAFRI, Tel-Aviv University,
OR PATASHNIK, Tel-Aviv University,
YUVAL ALALUF, Tel-Aviv University,
DANIEL COHEN-OR, Tel-Aviv University,

We present *StyleFusion*, a new mapping architecture for StyleGAN, which takes as input a number of latent codes and fuses them into a single style code. Inserting the resulting style code into a pre-trained StyleGAN generator results in a single *harmonized* image in which each semantic region is controlled by one of the input latent codes. Effectively, StyleFusion yields a disentangled representation of the image, providing fine-grained control over each region of the generated image. Moreover, to help facilitate global control over the generated image, a special input latent code is incorporated into the fused representation. StyleFusion operates in a hierarchical manner, where each level is tasked with learning to disentangle a pair of image regions (e.g., the car body and wheels). The resulting learned disentanglement allows one to modify both local, fine-grained semantics (e.g., facial features) as well as more global features (e.g., pose and background), providing improved flexibility in the synthesis process. As a natural extension, StyleFusion allows one to perform semantically-aware cross-image mixing of regions that are not necessarily aligned. Finally, we demonstrate how StyleFusion can be paired with existing editing techniques to more faithfully constrain the edit to the user's region of interest. Code is available at: https://github.com/OmerKafri/StyleFusion.

CCS Concepts: • **Computing methodologies** → **Learning latent representations**; *Image manipulation*; Neural networks.

Additional Key Words and Phrases: Generative Adversarial Network, Image Generation, Disentangled Representation

1 INTRODUCTION

Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] are nowadays considered a strong and common method for synthesizing images of phenomenal realism. In particular, StyleGAN [Karras et al. 2020a, 2019, 2020b] has been established as the state-of-the-art synthesis network, producing images of high visual quality and fidelity. The recent surge of interest in StyleGAN should also be attributed to its learned latent space $\mathcal{W}$, which has been shown to encode valuable semantic information [Abdal et al. 2019, 2020a] and intriguing disentanglement properties [Collins et al. 2020; Shen et al. 2020; Wu et al. 2020; Yang et al. 2020].

Disentangling various attributes is of utmost importance as it helps facilitate better and easier use of the generator and control over the synthesized images. Therefore, different latent spaces [Abdal et al. 2019, 2020a; Wu et al. 2020; Zhu et al. 2021b] and architecture variants [Gal et al. 2021; Kim et al. 2021; Kwon and Ye 2021; Lewis et al. 2021] of StyleGAN have been explored, aiming to construct a more disentangled latent representation. To leverage the disentanglement of StyleGAN's latent space for manipulating specific segments of an image, recent works have demonstrated that specific channels of the latent code control spatially local attributes of the generated image [Collins et al. 2020; Wu et al. 2020]. However, while there exist diverse editing techniques, they

Authors' addresses: Omer Kafri, Tel-Aviv University, , omer934@gmail.com; Or Patashnik, Tel-Aviv University, ; Yuval Alaluf, Tel-Aviv University, ; Daniel Cohen-Or, Tel-Aviv University,

0730-0301/2022/4-ART $15.00
https://doi.org/10.1145/3527168

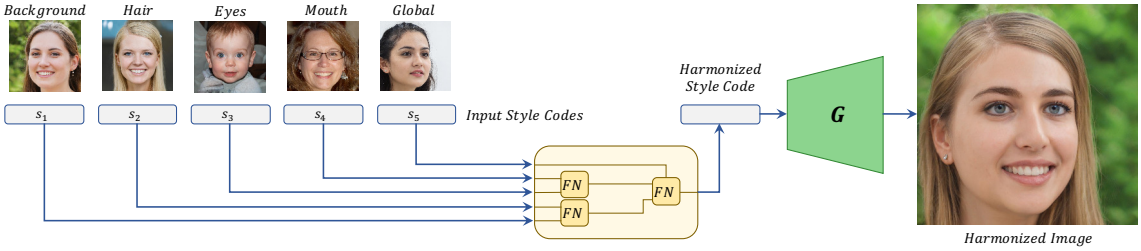


Fig. 1. ***StyleFusion Overview.*** Given a set of style codes, StyleFusion generates a single *harmonized* image in which each semantic region is controlled by one of the input latent codes. This harmonized image is obtained via a pre-trained generator (e.g., StyleGAN) by *learning* to fuse each of the input latent codes into a single unified style code. In a sense, this learned fusion results in a semantically-aware, disentangled representation of the image. We additionally introduce a latent code tasked with controlling global aspects of the generated image (e.g., the individual's pose). Observe that StyleFusion operates in a hierarchical manner and learns to disentangle the image regions in a coarse-to-fine fashion, providing users control over the learned disentanglement.

oftentimes struggle in achieving fine-grained local control over the generation as the single input latent code influences the synthesized image in its entirety. As such, attaining accurate control requires precise manipulation over the single latent representation.

In this work, we leverage the expressiveness of the StyleGAN latent space to provide direct and intuitive control over local, possibly semantic, segments of the generated image. Specifically, we introduce a mapping mechanism that learns to generate a single *harmonized* image from a *set* of latent codes, each controlling a specific segment of the synthesized image. This mechanism results in a disentangled latent representation corresponding to disjoint semantic image regions. As shall be shown, by learning a semantically-aware disentanglement, our proposed approach, *StyleFusion*, facilitates easier control over local, fine-grained features (e.g., facial attributes) and global image features, see Figure **??**. Specifically, StyleFusion provides an intuitive "plug-and-play" interface for users to compose a single harmonized image formed by fusing multiple latent codes, see Figure 1. As shown, StyleFusion operates in a hierarchical fashion, where each level of the hierarchy, a *FusionNet*, learns to disentangle a pair of image regions.

Observe that constructing a harmonized image from a set of latent codes is not trivial. For example, composing a single facial image where the background, identity, and hair are extracted from three different images is especially challenging when the three images are unaligned (e.g., of different poses) or contain different global features (e.g., lighting). To address this, we introduce an additional latent code tasked with aligning these global features of the source images before fusing the input latent codes. As a result, this learned alignment results in a more precise composition of semantic regions in the synthesized image.

We perform an extensive analysis of our method to show the advantages of the learned disentangled representation StyleFusion offers on multiple domains. To demonstrate this, we show how StyleFusion allows one to generate images with explicit control over a semantic target image region. Moreover, we show the role of the global latent codes in the synthesis process. Finally, we illustrate how our method complements other editing techniques. Specifically, as shown in Figure 2, our intuitive disentangled representations allow one to easily pair existing latent space editing techniques with our proposed representation, facilitating improved control over the manipulation of *real* images.

2 BACKGROUND AND RELATED WORKS

2.1 Learning a Disentangled Representation

Learning a disentangled representation is of utmost importance across numerous computer graphics and computer vision tasks. Such a representation can be viewed as one that encodes data using independent factors of variation, allowing one to control a single factor without affecting others [Locatello et al. 2019].

A key challenge in learning a disentangled representation is reducing the supervision needed to learn the desired disentanglement. For example, manually collecting paired data of the same scene in different styles is tedious and even impractical at times. To address this, diverse weakly-supervised and unsupervised methods have recently been explored. InfoGAN [Chen et al. 2016] learns a disentangled representation in an unsupervised manner by maximizing the mutual information between the latent variables and the observation. Lee *et al.* [2020] pair a Variational Autoencoder (VAE) and a GAN for learning a more meaningful, disentangled latent representation. Others have explored additional means for learning disentangled representations including, but not limited to, semi-supervised learning [Nie et al. 2020], adversarial training [Mathieu et al. 2016], cycle-consistency [Jha et al. 2018], and mixing sub-parts of different latent representations [Gal et al. 2020; Hong et al. 2020; Hu et al. 2018; Karras et al. 2019].

Following the seminal work of Isola *et al.* [2017], numerous works [Kotovenko et al. 2019; Lee et al. 2018; Wang et al. 2018; Zhu et al. 2017] learn to disentangle style and content for performing image-to-image translation between two domains. More recently, Park *et al.* [2020] train an autoencoder to encode real images into two independent components representing structure and texture in an unsupervised manner. Kazemi *et al.* [2018] and Kwon *et al.* [2021] modify the generator's architecture to explicitly learn a disentanglement of content and style.

Our work differs from these works in several key aspects. First, many works rely on human supervision such as collecting image pairs with the same style or appearance. Second, previous methods are typically more global in nature, mainly transferring texture, and may therefore struggle to model semantic-based local changes in the desired image. In contrast, our StyleFusion approach enables both local and global control via a learned disentanglement and fusion of multiple style codes.

2.2 GAN-Based Image Editing

While StyleGAN is able to synthesize images of phenomenal realism and diversity, one is often more interested in utilizing the trained GAN for performing image manipulations.

Latent Space Editing. Most commonly, recent works perform a latent space traversal of the GAN's learned manifold for controlling a specific attribute of interest such as age, gender, and expression [Abdal et al. 2020b; Goetschalckx et al. 2019; Härkönen et al. 2020; Jahanian et al. 2020; Shen et al. 2020; Shen and Zhou 2020; Voynov and Babenko 2020; Wu et al. 2020]. There have also been numerous works exploring more diverse methods for manipulating latent representations. Tewari *et al.* [2020a] learn semantic face edits by employing pre-trained 3DMM models. Shen *et al.* [2020] perform eigenvalue decomposition on the affine transformation layers of StyleGAN2 generators [Karras et al. 2020b] to learn versatile manipulation directions. Xia *et al.* [2021a] and Patashnik and Wu *et al.* [2021] manipulate images using a human-understandable text prompt providing a more intuitive image editing interface.

Other works propose end-to-end approaches for learning a direct disentanglement and editing of real images over a specific attribute such as hair style [Tan et al. 2020; Zhu et al. 2021a], facial identity [Li et al. 2020; Nitzan et al. 2020], and other facial attributes [Alaluf et al. 2021a; He et al. 2018; Hou et al. 2020; Lample et al. 2018].

While these works allow for extensive manipulations of images, most rely on the existence of a well-behaved, disentangled latent space, which is still difficult to achieve in practice. For example, it is not necessarily trivial to modify an individual's hairstyle by manipulating the image's latent representation without altering other

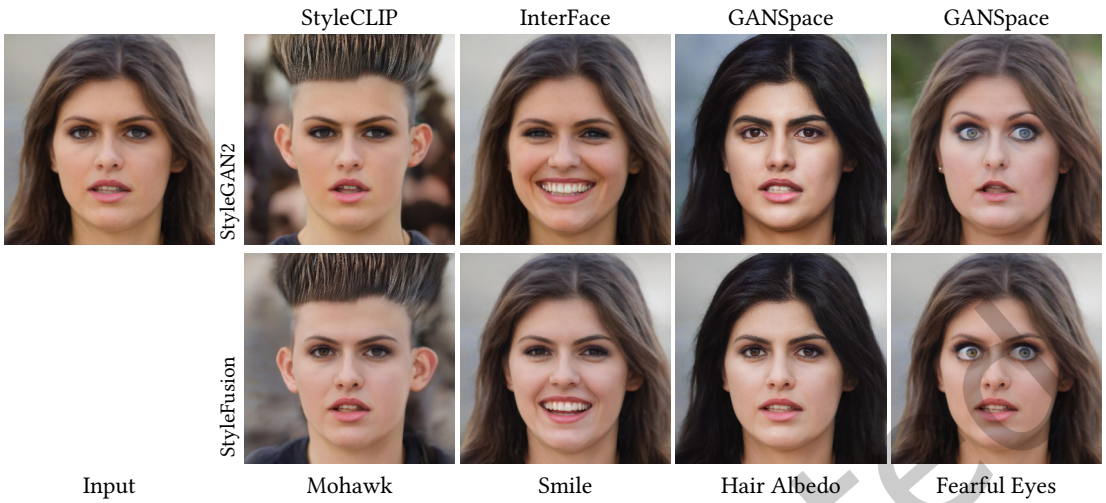


Fig. 2. ***Latent traversal editing techniques performed with StyleFusion.*** As shall by demonstrated, pairing StyleFusion with existing latent editing techniques (e.g., InterFaceGAN [Shen et al. 2020], StyleCLIP [Patashnik et al. 2021]) results in more precise image manipulations. The input image is a reconstruction obtained with an e4e encoder [Tov et al. 2021].

facial features since these works operate over a single latent code controlling the entire image. Therefore, a disentangled representation is of great importance for the success of these methods.

It is important to emphasize that our work is not designed to compete with the aforementioned editing techniques. Instead, one may view our work as complementing these existing approaches. For example, as shall be shown, pairing StyleFusion with GANSpace [Härkönen et al. 2020] or StyleCLIP [Patashnik et al. 2021] leverages their diverse manipulations while ensuring that the resulting edits alter only the desired semantic regions.

Editing Local Regions. Some works propose methods specifically aimed at editing local regions of the image. Alhabri *et al.* [2020] achieve a grid-based spatial disentanglement by injecting structured noise into a GAN. Chai *et al.* [2021] design a latent regressor capable of fusing multiple image patches into a single coherent image. Yet, their approach requires manual region selection from each input image. Bau *et al.* [2021] combine an input text and a mask to edit specific image regions. These works suffer from the same drawbacks in that require a manual per-image marking of the region to be altered. Conversely, our approach requires only specifying a semantic region from a pre-defined set, providing easier and more intuitive user control. Collins *et al.* [2020] and a very recent work from Chong [2021] blend two style codes to perform semantic editing guided by a target spatial segment of the image. Finally, similarly to our approach, Zhu *et al.* [2020a] encode images using a set of style codes, each controlling a specific image region. However, their method requires collecting pairs of corresponding images and segmentation maps for learning this disentanglement.

A particular task that has achieved increased attention is that of *hair transfer* where we wish to transfer the hairstyle from one individual to another. Due to the entanglement of the hair region with other image attributes (e.g., pose, lighting), solving this task remains challenging. Tan *et al.* [2020] design a conditional GAN for learning to disentangle hair into structure and appearance. Saha *et al.* [2021] perform latent vector optimization to explicitly disentangle hair attributes. More recently, Zhu *et al.* [2021a] perform mixing between the latent representations of a reference and target image, guided by a target segmentation map. However, these works rely on external supervision (e.g., segmentation masks) or a specialized inpainting mechanism to guide the learned disentanglement and region transfer. Using our learned disentangled representation, we demonstrate StyleFusion's ability to faithfully transfer the hair region, even though it was not explicitly trained to do so.

3 PRELIMINARIES

In recent years, StyleGAN [Karras et al. 2020a, 2019, 2020b] has become regarded as the state-of-the-art image generator. In addition to the standard Gaussian latent space $\mathcal{Z}$, StyleGAN has a learned latent space $\mathcal{W}$ that is derived from the initial latent space $\mathcal{Z}$ via a fully-connected mapping network. Many works have demonstrated that $\mathcal{W}$ is able to better encode semantic properties which provide control over the generated image [Collins et al. 2020; Karras et al. 2019; Shen et al. 2020]. In the standard StyleGAN architecture, a latent code $w \in \mathcal{W}$ is additionally transformed via a different learned affine transformation at each input layer of the generator. The outputs of these affine transformations are then inserted into the generator's synthesis network through style modulation layers at different resolutions to generate the output image.

The space spanned by the output of the affine transformations, referred to as the StyleSpace or the $\mathcal{S}$ space, has recently been explored [Liu et al. 2020; Wu et al. 2020]. Assuming a generator that outputs images at a 1024×1024 resolution, the $\mathcal{S}$ space has $9,088$ dimensions in total, where $6,048$ dimensions are applied over the feature maps, and $3,040$ additional dimensions are applied over the tRGB blocks. We refer the reader to [Wu et al. 2020] for a more detailed description of the $\mathcal{S}$ space. Wu *et al.* [2020] demonstrate that $\mathcal{S}$ is more disentangled than $\mathcal{W}$ and is therefore better suited for providing fine-grained control over the generated images.

Collins *et al.* [2020], demonstrate that a per-channel linear interpolation between two style codes allows one to transfer local regions between the two corresponding images. In this work, we refer to this per-channel interpolation as *fusion*. We employ two ideas presented in Collins *et al.* [2020]. First, given an image generated by StyleGAN, it is possible to segment the image into semantic regions by clustering the activation vectors at a given layer. Second, given a partitioning of an image into K semantic regions, Collins *et al.* [2020] define a matrix that measures the correspondence between a style code channel c and a semantic region k of the image.

This matrix has been shown to be useful for various editing methods [Collins et al. 2020; Lewis et al. 2021], and is defined as follows:

$$M_{k,c} = \frac{1}{NWH} \sum_{n,h,w} A^2_{n,c,h,w} \cdot U_{n,k,h,w}. \tag{1}$$

Here, N is the number of images over which the matrix is calculated, $A \in \mathbb{R}^{N \times C \times H \times W}$ is the normalized activation tensor at a given layer of StyleGAN, and $U \in \{0,1\}^{N \times K \times H \times W}$ specifies the region membership of each spatial location to a segmented region. Note that both $A^2_{n,c,h,w}$ and $U_{n,k,h,w}$ are scalars and that $M_{k,c}$ is normalized such that $M_{k,c} = 1$ denotes that the influence of channel c is fully contained in region k. Similarly, $M_{k,c} = 0.5$ means that c has equal influence over both k and its complement. Finally, $M_{k,c} = 0$ indicates that c has no influence over the image region k.

4 METHOD

We start by presenting a high-level overview of our approach for learning a disentangled latent representation of semantic image segments. Given a pre-trained StyleGAN generator, our goal is to generate an image by fusing a set of K latent codes each of which controls a *single* spatial, semantic segment (or region) of the generated image. As the generated image is defined by a learned fusion of multiple style codes, we name our approach *StyleFusion*. Intuitively, such a scheme would allow one to seamlessly control specific regions of the image without altering other regions.

We illustrate this idea in Figure 1 where five latent codes are randomly sampled from the $\mathcal{Z}$ latent space. These latent codes are first passed through StyleGAN's mapping network and learned affine transformation layers and encoded into the generator's $\mathcal{S}$ space [Wu et al. 2020]. The resulting generated image of each input style code is shown above as a reference to the reader.

Having obtained the five $\mathcal{S}$-space representations of the input codes, our StyleFusion mechanism operates in a hierarchical fashion. Each component in this hierarchy, a *FusionNet*, learns a spatial disentanglement of a pair of

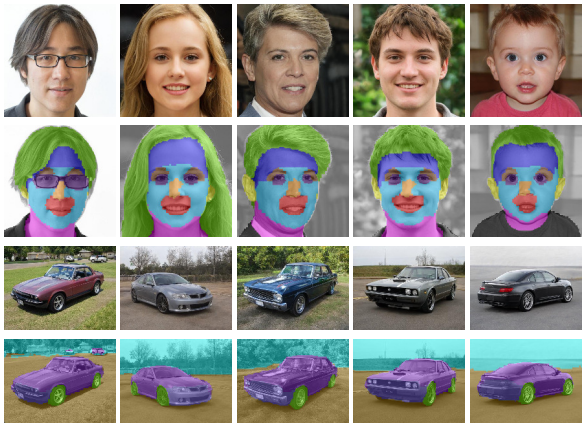


Fig. 3. ***The semantic image regions of StyleFusion.*** Here, we illustrate the different semantic image segments computed by StyleFusion to learn a disentangled representation on the human facial and cars domains.

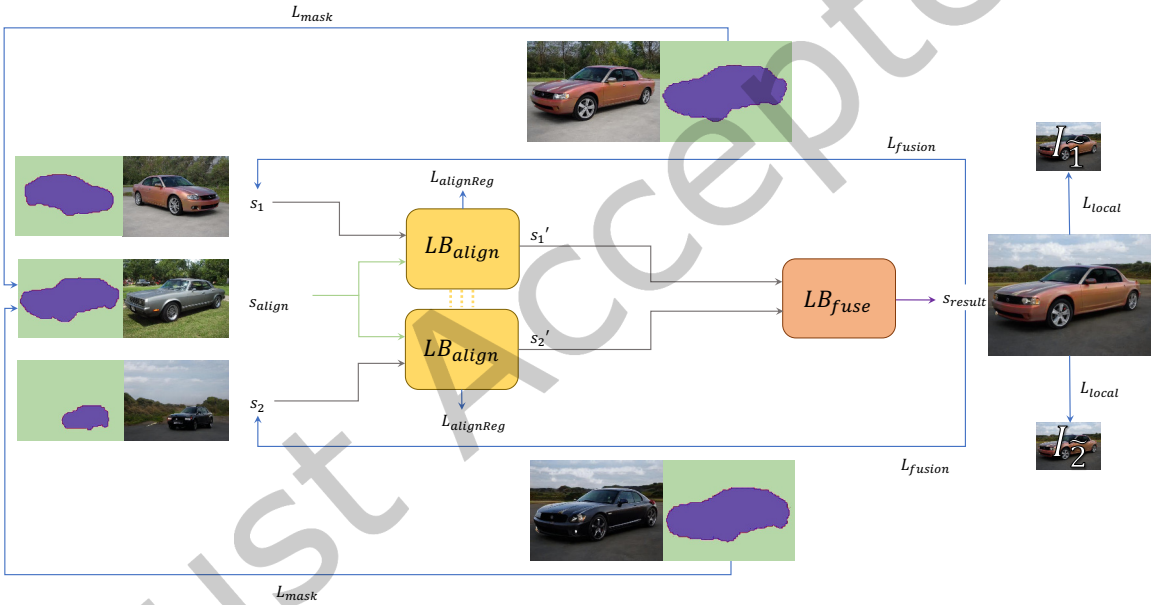


Fig. 4. ***The FusionNet architecture.*** The FusionNet learns to disentangle between two specified semantic image regions (e.g., the car body and image background). The FusionNet, composed of a pair of building blocks named *Latent Blender* (LB), receives as input three input latent codes $s_1, s_2, s_{align} \in \mathcal{S}$. The first block, LB_{align} aligns the images of s_1 and s_2 to match the spatial location of the image regions of s_{align}. Observe how the images of the resulting latent codes s_1' and s_2' match the spatial layout of s_{align}, namely the car pose and background layout, as illustrated by the segmentation mask shown to the right of each image. Observe that while LB_{align} operates on s_1 and s_2 independently, its network weights are shared between the two. Having aligned the two images, the second block, LB_{fuse}, is then used to fuse the two semantic regions into a unified image. Specifically, notice how the car body in the resulting image is taken from s_1, the background is taken from s_2, and the spatial layout and pose are controlled by s_{align}.

image regions by learning to disentangle and interpolate between their corresponding $\mathcal{S}$-space representations. The output of the StyleFusion hierarchy is a single harmonized style code constructed from the five input style codes. Finally, passing this style code to the pre-trained StyleGAN generator returns the final harmonized image, shown to the right. This harmonization is, in part, achieved by leveraging the prior of the well-trained StyleGAN network for generating realistic images. Observe that in the harmonized image, each semantic region is controlled by a single input latent code, demonstrating StyleFusion's learned disentanglement of the regions.

4.1 Discovering the Semantic Image Regions

To learn a disentanglement of the input latent codes into different semantic regions of interest, we must first segment the generated image into those segments. Recently, numerous works have demonstrated the ability to perform segmentation by employing StyleGAN's learned features [Abdal et al. 2021; Collins et al. 2020; Pakhomov et al. 2021; Zhang et al. 2021]. In our work, we build on the segmentation approach introduced in Collins *et al.* [2020]. Consider all the activation tensors along StyleGAN's synthesis network. We first up-sample the tensors to match the size of the largest activation tensor, and then concatenate the activations to a single tensor composed of $C = 6,048$ channels, corresponding to the $6,048$ channels of the $\mathcal{S}$ space.

Having obtained the joint representation, we then apply Ward's hierarchical clustering [Ward 1963] where each spatial location is assigned a cluster membership. Finally, we review the resulting cluster and label each cluster as corresponding to a certain semantic object. In the case that multiple clusters correspond to a single attribute, we merge their regions. Notice that this process occurs one time per domain, requiring minimal human supervision (e.g., several minutes). That is, given a newly generated image, the image's semantic regions can be computed without additional supervision by assigning each pixel to its closest semantic cluster. We refer the reader to the supplementary materials for additional details on the clustering procedure for computing the semantic region masks.

Figure 3 shows examples of the resulting clusters generated for the human facial and cars domains. As can be seen, the clustering process results in clusters that encode semantically-meaningful attributes (e.g., hair, eyes, mouth for the human facial domain and wheels, body, and background for the cars domain).

4.2 FusionNet

The FusionNet component is tasked with learning a spatial disentangled representation over two semantic regions R_1 and R_2 (e.g., hair and face). The FusionNet receives as input three randomly sampled style codes $s_1, s_2, s_{align} \in \mathcal{S}$ where s_1 and s_2 are designed to control the semantics associated with regions R_1 and R_2. Additionally, s_{align} is designed to align the spatial location of R_1 and R_2 in the resulting fused image. Specifically, the spatial locations of the two image regions in the final fused image is determined by the segmentation of the image corresponding to s_{align}. In a sense, the image corresponding to s_{align} acts as the target canvas or template for the spatial layout of the final image.

Given these three latent codes, the FusionNet is designed to output a single unified style code $s_{result} \in \mathcal{S}$ with the following properties:

(1) Every semantic attribute in R_1 is determined by s_1.
(2) Every semantic attribute in R_2 is determined by s_2.
(3) The spatial location of each region and image features common to both input images are determined by s_{align}.

Below, we describe the FusionNet architecture, followed by the training scheme and loss objectives employed for achieving these properties.

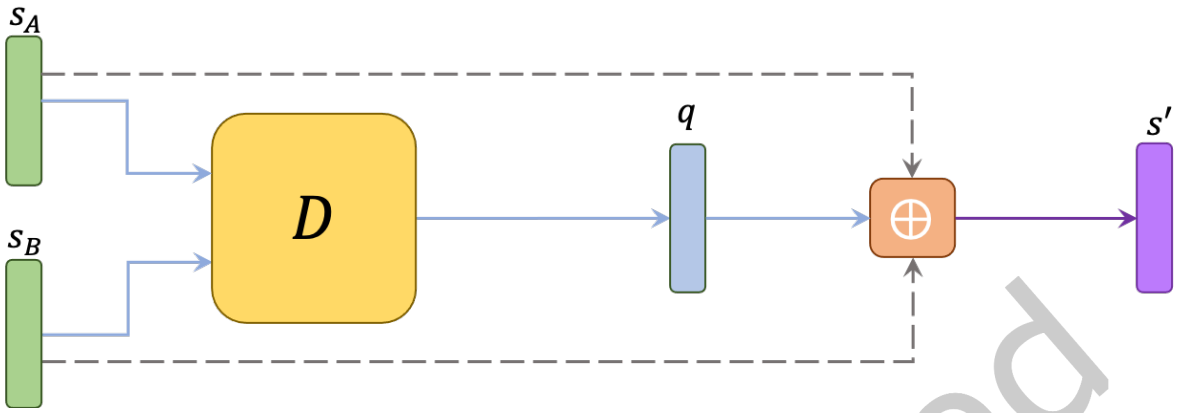


Fig. 5. ***The Latent Blender.*** Given two input latent codes $s_A, s_B \in \mathcal{S}$, the Latent Blender, composed of a fully-connected network D, learns a single fusion coefficient $q \in [0,1]^{\mathcal{S}}$ used to blend between the two input codes via a per-channel interpolation, resulting in a fused latent code s'.

Architecture. To generate a harmonized image from two latent codes s_1 and s_2, we introduce the FusionNet constructed from a pair of building blocks named *Latent Blender*. The first blender, denoted LB_{align} is trained to align s_1 and s_2 to a common spatial layout determined by the third latent code, s_{align}.

Given the aligned latent codes, the second blender, LB_{fuse} is tasked with learning to fuse the semantic regions of the two input latent codes. This idea is illustrated in Figure 4 where s_1 controls the car body and s_2 controls the background of the generated image, with the spatial layout (i.e., car pose) determined by s_{align}. Note that while LB_{align} and LB_{fuse} share the same architecture (described below), they naturally differ in their role.

Latent Blender. As mentioned above, the core component of the FusionNet is the *Latent Blender*, illustrated in Figure 5. Consider two input latent codes $s_A, s_B \in \mathcal{S}$, which are concatenated and passed to a simple fully-connected network, denoted D, followed by a sigmoid activation function. The output of this network is a single vector $q \in [0,1]^{\mathcal{S}}$, which we label the fusion coefficient vector.

This coefficient vector and the input style codes are then passed to the *fusion* procedure which performs a per-channel interpolation between the two style codes, weighted by the fusion coefficient. Specifically, we compute,

$$s' = q \odot s_A + (1 - q) \odot s_B, \tag{2}$$

where $\odot$ denotes a per-element multiplication. That is, this operation blends s_A and s_B according to q and outputs a unified code s'.

Observe that, in practice, the size of the fully-connected network described above will be large — its input will be of size $2|\mathcal{S}|$ and its output will be of dimension $|\mathcal{S}| = 9,088$. To address this, we propose to use a layer encoding allowing the Latent Blender to consider each of the generator's input layers independently. We describe this design in detail in the supplementary materials.

4.3 Training

At each training iteration, we begin by randomly sampling four latent codes: $z_1, z_2, z_{align}, z_{rnd} \in \mathcal{Z}$. We then pass the latent codes to StyleGAN's mapping function and affine transformation layers resulting in their corresponding style codes $s_1, s_2, s_{align}, s_{rnd} \in \mathcal{S}$.

Given the style codes s_1, s_2, s_{align}, the FusionNet blends the three codes into a single style code s_{result}. Passing the code through StyleGAN's synthesis network, we obtain the corresponding image I_{result}. Finally, using the clustering procedure described in Section 4.1, we compute the $M^1_{k,c}, M^2_{k,c}$ correlation matrices of the images of s_1, s_2, respectively.

Given the input $\mathcal{S}$-space representations and the resulting fused style code s_{result}, the FusionNet is trained using a weighted combination of several loss objectives described below. Note that during training the pre-trained generator remains fixed.

Mask Loss. The mask loss is tasked with aligning the two images corresponding to s_1 and s_2. This is done by demanding that the spatial regions in the fused output image will be aligned with the spatial regions of the image I_{align} corresponding to s_{align}. Assume we have obtained the latent code s_{result} corresponding to the generated image I_{result}. We additionally define $m_1(I_{result}), m_2(I_{result})$ to denote the masks of image regions R_1 and R_2 in the resulting image I_{result}, computed using the clustering approach described in Section 4.1. Specifically, $m_i(I_{result})$ takes a value of 1 in all pixels inside image region R_i and 0 elsewhere.

The mask loss is then defined as,

$$\begin{aligned} d_1 &= \left\| m_1(I_{result}) - m_1(I_{align}) \right\|_1 \\ d_2 &= \left\| m_2(I_{result}) - m_2(I_{align}) \right\|_1 \\ \mathcal{L}_{mask} &= \frac{d_1 + d_2}{\left\| m_1(I_{align}) + m_2(I_{align}) \right\|_1}. \end{aligned} \tag{3}$$

Here, we align the masks corresponding to regions R_1, R_2 in the fused image I_{result} according to the regions' spatial locations in the image I_{align}. Consider a motivating example of fusing the face and hair regions. This loss encourages the FusionNet to align the face and hair segments of the two input latent codes according to the spatial location of the hair and face regions in the image I_{align}.

Observe that in order to compute this loss, one must calculate the masks in a differentiable manner. Details on doing so are provided in the supplementary materials.

Localization Loss. The core idea behind the disentanglement process is to demand that the latent code s_1 does *not* affect the semantic region R_2 of the generated image, and vice-versa. To encourage the FusionNet to only interpolate between style channels affecting the region of interest, we introduce a localization loss as follows. Given the styles s_1 and s_2, we insert small perturbations into the style codes according to s_{rnd}:

$$\begin{aligned} \widetilde{s_1} &= s_1 + \varepsilon \cdot (s_{rnd} - s_1), \\ \widetilde{s_2} &= s_2 + \varepsilon \cdot (s_{rnd} - s_2), \end{aligned} \tag{4}$$

where ε is a scalar denoting the perturbation strength. Given $\widetilde{s_1}, \widetilde{s_2}$, we then generate two images in addition to I_{result} introduced above:

(1) $I_{\widetilde{1}}$ by fusing the style codes $\widetilde{s_1}, s_2, s_{align}$
(2) $I_{\widetilde{2}}$ by fusing the style codes $s_1, \widetilde{s_2}, s_{align}$.

We additionally generate the two binary masks $m_1(I_{result}), m_2(I_{result})$ as defined in the mask loss above. The localization loss is then computed as:

$$\begin{aligned} \mathcal{L}_{local} = &\frac{\left\| m_2(I_{result}) \cdot I_{result} - m_2(I_{result}) \cdot I_{\widetilde{1}} \right\|_2}{\left\| m_2(I_{result}) \right\|_1} + \\ &\frac{\left\| m_1(I_{result}) \cdot I_{result} - m_1(I_{result}) \cdot I_{\widetilde{2}} \right\|_2}{\left\| m_1(I_{result}) \right\|_1}. \end{aligned} \tag{5}$$

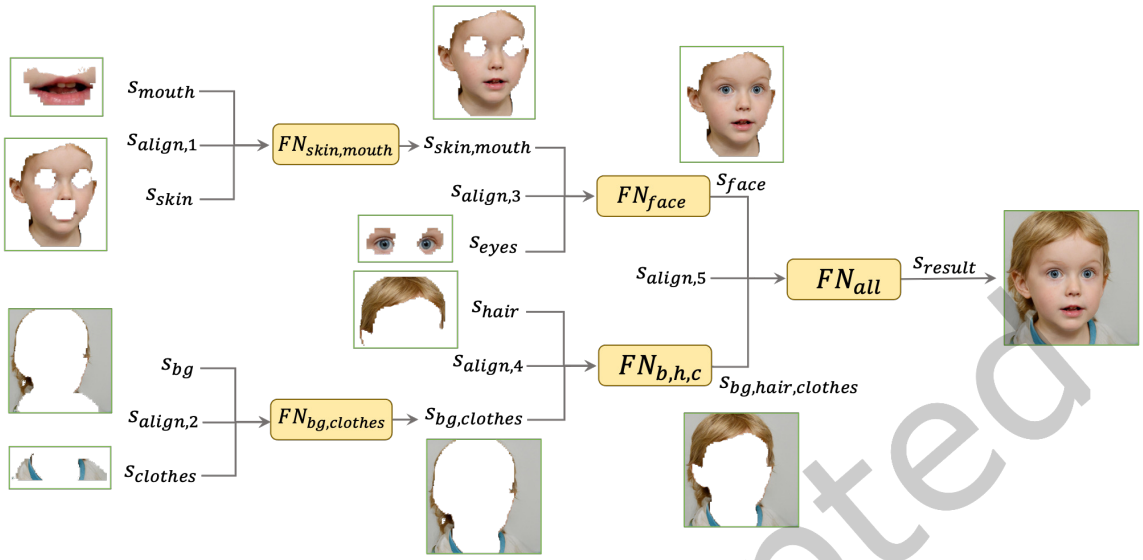


Fig. 6. ***Hierarchical Multi-Fusion.*** By constructing a tree of FusionNets (FN), StyleFusion is able to disentangle multiple image regions in a coarse-to-fine fashion. Given a set of input codes, each FusionNet learns to disentangle then fuse two image regions which are passed as input to the next FusionNet in the hierarchy. The final output is a single image composed from all input codes. The hierarchies for all domains are provided in the supplementary materials.

Observe that we divide the losses of the two regions by their spatial size to normalize the loss. Consider our motivating example of disentangling between the face region (R_1) and the hair region (R_2). This loss ensures that making small changes to the face style code s_1 does not change the hair region of the resulting image (i.e., $m_2(I_{result}) = m_2(I_{\overline{1}})$). Likewise, making small changes to s_2 should not result in changes to the image's face region.

Common Regularization Loss. Consider the first Latent Blender, LB_{align}, within the FusionNet that is tasked with aligning the two input latent codes s_1, s_2 according to the alignment code s_{align}. Consider a scenario in which LB_{align} outputs the all-ones vector for the fusion coefficient q_{align}. In such a case, the fusion procedure will then return $s'_1 = s'_2 = s_{align}$ for any input. As a result, the intermediate images I'_1 and I'_2 will both be equal to I_{align} and the final fused image will also be equal to I_{align}.

To avoid this trivial solution to the fusion procedure, we constrain the magnitude of the fusion coefficient of LB_{align},

$$\mathcal{L}_{alignReg} = \left\| q_{align} \right\|_1 . \tag{6}$$

Fusion Loss. To faithfully transfer each image region R_i from its source image to the final fused output image, we define weight vectors $w_{R_1}, w_{R_2} \in [0,1]^{|S|}$. Specifically, for a given style channel c, $w_{R_i}(c)$ represents the contribution of channel c to the region R_i. Using the weight vectors, we define the fusion loss as,

$$\begin{aligned} \mathcal{L}_{fusion} = & \left\| w_{R_1} \odot (s_{result} - s_1) \right\|_2 + \\ & \left\| w_{R_2} \odot (s_{result} - s_2) \right\|_2 \end{aligned} \tag{7}$$

where we define $w_{R_i} = [w_{R_i}(c_0), w_{R_i}(c_1), ..., w_{R_i}(c_{|S|})]$. Back to our example of disentangling between the face (R_1) and hair (R_2), this loss is designed to ensure that channels in s_1 that influence the face region are transferred to the fused code s_{result} while channels in s_2 that control the hair region are transferred to s_{result}.

We now turn to describe how each of the weight vectors w_{R_i} is computed for region R_i and corresponding code s_i. As described above, the correlation between a style code channel c and an image region k is encoded in a matrix $M_{k,c}$. Specifically, $M_{k,c}$ measures what portion of the activations of channel c are located in the semantic cluster corresponding to the image region k.

Consider the image of the latent code s_i. We compute the image's corresponding correlation matrix for region R_i (i.e., $M_{R_i,c}$ for all channels c). The weight vector for region R_i is then defined by,

$$w_{R_i}(c) = 2 \cdot \max(M_{R_i,c}, 0.5) - 1. \tag{8}$$

Note that these weights rank the contribution of each channel c to R_i. In words, if c has a strong contribution over R_i (i.e., $M_{R_i,c} > 0.5$), we set $w_{R_i}(c) > 0$. Conversely, if c does not contribute to the image region R_i, we set $w_{R_i}(c) = 0$.

Total Loss. In summary, the complete loss objective is given by:

$$\begin{aligned}\mathcal{L} = \lambda_{mask}\mathcal{L}_{mask} + \lambda_{local}\mathcal{L}_{local} + \\ \lambda_{alignReg}\mathcal{L}_{alignReg} + \lambda_{fusion}\mathcal{L}_{fusion},\end{aligned} \tag{9}$$

where $\lambda_{mask}, \lambda_{local}, \lambda_{alignReg}, \lambda_{fusion}$ denote the loss weights. Additional implementation details and hyper-parameters are provided in the supplementary materials.

Multi-Step Training. To improve the results and provide a more stable training process, we use a multi-step training procedure. Specifically, we split the training process into three stages as follows:

(1) Only the first Latent Blender component, LB_{align}, which is tasked in aligning the two input latent codes, is trained. In this stage, the FusionNet does not use the LB_{fuse} component. We instead employ only $\mathcal{L}_{mask}$, $\mathcal{L}_{alignReg}$, and $\mathcal{L}_{fusion}$ on the intermediate outputs s_1' and s_2' and their corresponding generated images.
(2) Here, only the LB_{fuse} component is trained. This stage focuses on identifying the attributes we wish to disentangle, given the aligned latent codes. To achieve this, we apply only L_{fusion} on the intermediate latent codes s_1' and s_2'.
(3) In the final stage, both components are trained simultaneously using $\mathcal{L}_{local}$, $\mathcal{L}_{alignReg}$, and $\mathcal{L}_{fusion}$.

By splitting the training into multiple steps, we allow the network to focus on first aligning the images and then disentangling the desired attributes, rather than having to perform the two simultaneously. We find that doing so results better convergence with less sensitivity to the choice of hyperparameters. We validate the effectiveness of the multi-step training process in Section 7 where we compare the multi-step training of StyleFusion to a model trained in an end-to-end fashion. We additionally explore the contribution of each loss term described above. Finally, we provide the training duration for each of the above steps in the supplementary materials.

4.4 Hierarchical Multi-Fusion

To obtain a disentanglement of multiple image regions, we build a hierarchy of FusionNets. Specifically, we model the spatial regions of the image as a binary tree, in which each node corresponds to a single region of the image. The two children of each node correspond to the partition of the region into two sub-regions (e.g., the face to the eyes and mouth). Note that the root of the tree corresponds to the entire image.

As such, we build a set of FusionNets, one for each parent node in the above tree. We then combine the FusionNets into a hierarchy as illustrated in Figure 6. Observe that the training of the hierarchy is done sequentially, starting with the root FusionNet. Once the parent node is trained, we begin training the child nodes with the

Table 1. ***Local control impact over FID score.*** We calculate the FID of StyleFusion's local control by generating two latent codes, one for the controlled region and the second for the rest of the regions.

	StyleGAN2	Hair	Mouth	Background	Eyes
FID	3.06	13.17	3.36	9.97	4.32

parent node remaining fixed. Observe further that the tree need not be balanced, providing users the ability to control the granularity of the learned disentanglement into disjoint image regions. In a sense, this hierarchy provides users a plug-and-play interface for generating a unified image composed of multiple semantic regions.

For example, consider the human facial domain, shown in Figure 6. The first FusionNet, FN_{all}, is trained to disentangle between the face and the remaining image features (background, hair, and clothes) using input style codes s_{face} and $s_{bg,hair,clothes}$. Once trained, FN_{all} becomes fixed. We can then move to disentangle the face into multiple facial regions by introducing two additional codes s_{eyes} and $s_{skin,mouth}$ and training an additional FusionNet, FN_{face}. Similarly, we can disentangle the image background and hair into two sub-parts using a third FusionNet, $FN_{b,h,c}$. We can continue this process and move down the constructed tree with each level resulting in an additional learned disentanglement.

After training is complete, at inference time, we utilize a set of input latent codes for controlling each of the semantic regions (mouth, skin, eyes, hair, etc.) and additional latent codes (i.e., the global codes) for controlling the spatial layout of each region in the resulting facial image. Observe that each stage of the hierarchy may receive a different global code or the same global code.

Note that users may also control the granularity of the composition post-training. For example, users may take the mouth and skin from the same image by bypassing the FusionNet labeled $FN_{skin,mouth}$ and passing the desired input code directly to FN_{face}.

5 RESULTS

To gain a stronger understanding of StyleFusion and attain key insights into its functionality, we now turn to analyze the various capabilities of our proposed system. Specifically, we begin by analyzing the locality of StyleFusion's control (e.g., altering the eyes does not affect other regions of the generated image). We then illustrate the role of the "global" latent code in controlling global attributes such as pose and structure and show its usefulness in more challenging domains with large variability. Additional results can be found in the supplementary materials.

Local Semantic Control. The first natural evaluation of the learned disentanglement of StyleFusion can be done by generating multiple images sharing the same characteristics with the exception of a single semantic image region (e.g., images sharing the same face, hair, eyes, but all with different mouths). In other words, we would like to provide users with local control over the image attributes. In Figure 7 we analyze StyleFusion's ability to provide such local control in the human facial domain. In the Figure, each row corresponds to a single semantic region which we wish to modify in the target image. Observe, for example, how StyleFusion is able to faithfully alter the hair in the top row and the mouth in the third row while preserving the other attributes and the individual's identity. Similarly, in Figure 8, notice StyleFusion's ability to faithfully manipulate more global features such as the image background as well as smaller details such as the wheels of the car.

To further validate the locality of StyleFusion's control, we refer the reader to Figures 9 and 10. First, in Figure 9, we demonstrate the area controlled by a target semantic region R. To do so, we sample one latent code, s_{R^C}, to control all of the regions other then R, and sample two latent codes, s_{R_1}, s_{R_2} for controlling region R. We then measure the per-pixel difference between the image generated by (s_{R_1}, s_{R^C}) and the image generated by (s_{R_2}, s_{R^C}). We then average this difference over 500 randomly chosen samples for R_1, R_2, and R_C. As can be seen, altering a specific target attribute results in variations specific to the desired region with very small changes to other facial attributes.

Fig. 7. ***Local semantic control on the facial domain.*** StyleFusion faithfully controls specific facial attributes without altering other attributes. Observe our ability to control both finer details (e.g., eyes), as well as more global attributes (e.g., hair) showing the versatility of StyleFusion's control.

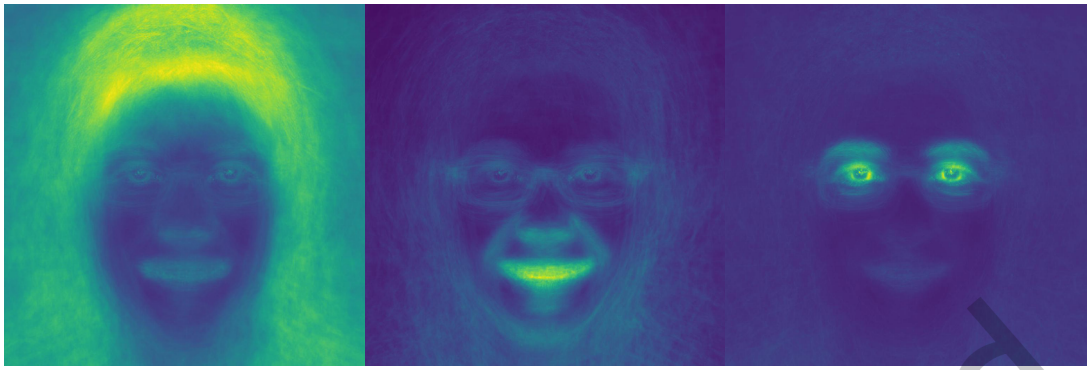
The above visualization is useful for the human facial domain since the semantic regions generally remain in the same spatial location across all samples. For domains with higher variability (e.g., cars) this assumption cannot be made. We therefore refer the reader to Figure 10. There, we illustrate the image regions controlled by a given semantic attribute, relative to a constant latent code for all regions other than R. In other words, we repeat the same process as above, but keep s_{R^C} fixed over all of the samples. Observe that even in the more complex

Fig. 8. ***Local semantic control on the cars domain.*** StyleFusion faithfully controls specific attributes of the car image without altering other attributes. Similar to the facial domain, observe the control over finer details (e.g., wheels), as well as more global attributes (e.g., background).

cars domain, StyleFusion is still able to faithfully alter the specified region, demonstrating the accuracy of the learned disentanglement.

In Table 1, we provide a quantitative evaluation of altering local regions in images generated with StyleFusion. For each region, we randomly generate 50,000 images where two different latent codes are used for each image: (i) one for controlling the region of interest and (ii) one for all other regions. We then compute the FID [Heusel et al. 2018] of the generated images. As can be seen, the FID scores of the images generated by StyleFusion remain in-line with that of images generated with StyleGAN2.

Coarse Manipulation. We now turn to analyze the role of the global latent code when learning to disentangle the different semantic image regions. Recall that the global latent code is tasked with aligning the location of semantic regions of one image with those of a second image. Here, we demonstrate how StyleFusion provides users with control over coarse attributes such as head pose, car pose, and church structure by altering this global code. Specifically, consider Figure 11 where we demonstrate control over such features. Here, different



Hair Mouth Eyes

Fig. 9. ***Visualizing local changes.*** For each attribute, we randomly sample 500 images and alter the specified attribute. We then compute the per-pixel differences between the original and altered images and average across all 500 images to obtained the displayed heatmaps. Observe that the changes are easily perceptible while mainly altering the specified region.

global variations are created by randomly sampling different global latent codes as input to StyleFusion (with the remaining input latent codes remaining fixed).

Observe StyleFusion's ability to change the pose of the car while faithfully preserving its car body and small details such as the wheels. Interestingly, StyleFusion is able to complete missing details when altering pose. This demonstrates the versatility of StyleFusion in that it facilitates control over both local and global aspects of the generated image, even in the absence of specific details. Additionally, altering the global code allows one to generate church buildings of the same style but different structure, as shown in Figure 12.

6 APPLICATIONS

In this section, we explore various applications of StyleFusion. We refer the reader to the supplementary materials for additional results and analysis.

6.1 Real Image Editing

In the previous section, we demonstrated how StyleFusion's learned disentanglement provides users with both local and global control over the generated images. In this section, we extend this and illustrate how the representation learned by StyleFusion can be paired with existing latent space editing techniques to achieve improved editing control over *real* images.

To manipulate real images, it is first necessary to invert them into their latent code representations. This is typically done via per-image optimization [Abdal et al. 2019, 2020a; Bau et al. 2019; Creswell and Bharath 2018; Karras et al. 2020b; Lipton and Tripathi 2017; Roich et al. 2021; Tewari et al. 2020b; Zhu et al. 2016] or by training an encoder to learn a direct mapping from a given image to its corresponding latent code [Alaluf et al. 2021b; Chai et al. 2021; Perarnau et al. 2016; Pidhorskyi et al. 2020; Richardson et al. 2020; Tov et al. 2021; Zhu et al. 2020b, 2016]. For a comprehensive survey on GAN inversion, we refer the reader to Xia *et al.* [2021b]. In this work, we adopt the e4e encoder from Tov *et al.* [2021] as it is designed to facilitate improved editing on the resulting inversions.

Many methods resort to manipulating images by traversing the latent space. A main challenge in doing so is editing a specific region by modifying a latent code that controls the entire image. As shown above, StyleFusion allows one to manipulate a latent code controlling specific target regions of the image. It is therefore natural to

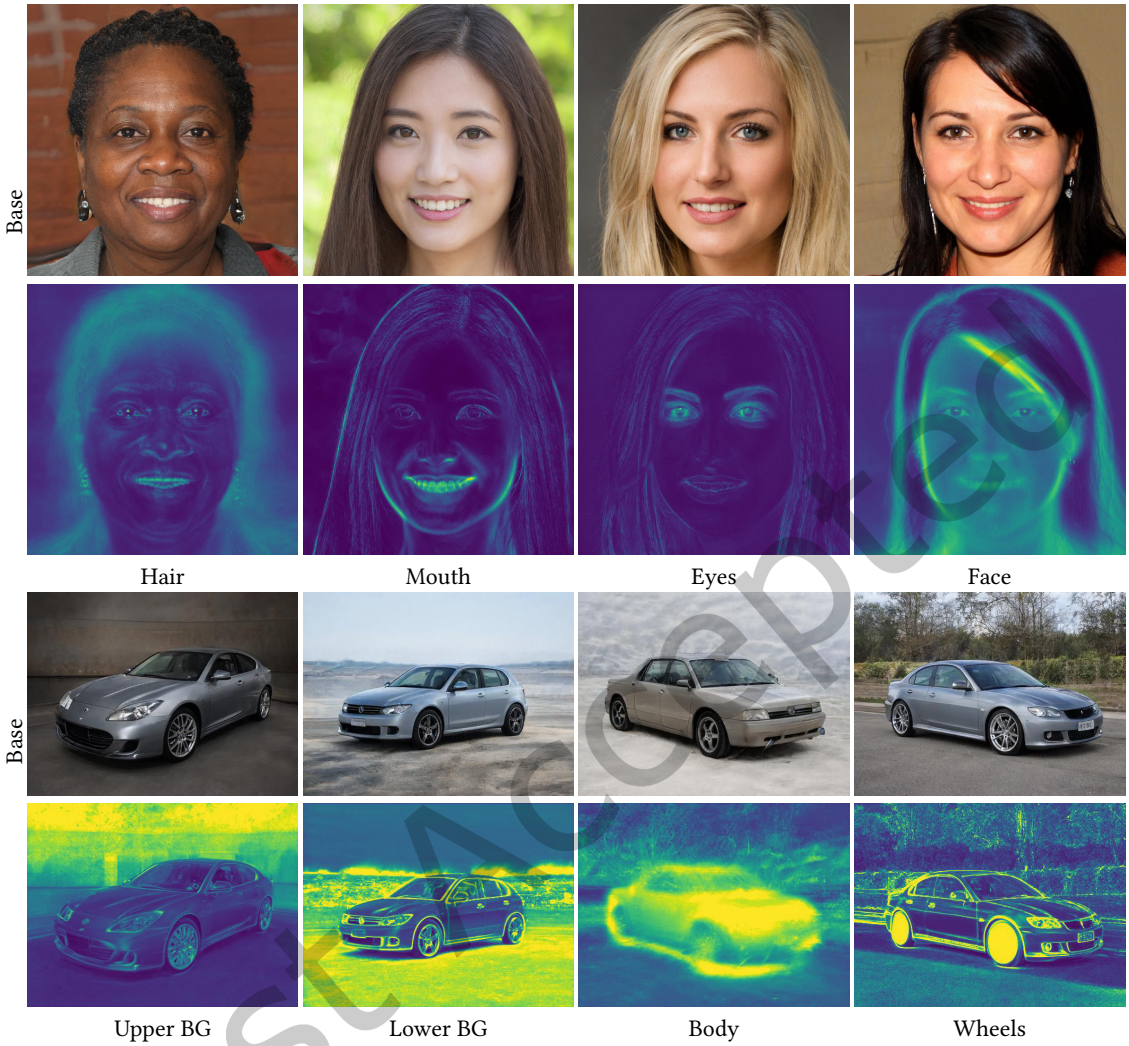


Fig. 10. ***Visualizing local changes.*** Similar to Figure 9 where instead of averaging over 500 randomly sampled images, we average over 500 random modifications of the specified attribute relative to a constant base image (shown above each corresponding heatmap)

.

pair existing latent traversal methods with StyleFusion for manipulating only the latent code controlling the desired region.

Given an input image, we first invert the image into its $\mathcal{W}+$ latent representation, which we denote by w. We then manipulate the resulting code to obtain the edited representation w_{edit} (e.g., via a traversal along a latent path learned by InterFaceGAN or GANSpace). We then pass to StyleFusion's hierarchy the $\mathcal{S}$-representation of w_{edit} in all regions relevant to the desired edit with w being passed as input to the remaining FusionNets. For example, when performing the "fearful eyes" edit we pass w_{edit} as input to FN_{face} while the original code w is passed as input to the remaining networks.

Fig. 11. ***Coarse control using StyleFusion.*** We demonstrate the control gained over more coarse-like image attributes such as pose, hair style, and church structure by altering the global style code controlling the specified target attribute. Observe, for example, that altering these features does not affect other details (e.g., identity or car wheels).

In Figure 13 we show the advantage of using StyleFusion's disentangled representation when editing images using three latent traversal editing methods: InterFaceGAN [Shen et al. 2020], GANSpace [Härkönen et al. 2020], and StyleCLIP [Patashnik et al. 2021]. For InterFaceGAN and GANSpace we use their official implementation and latent directions while for StyleCLIP we adapt the official implementation to train a latent mapper which manipulates only the desired image region.

As can be seen, pairing these editing methods with StyleFusion's disentangled representation leads to more accurate, local edits that more faithfully alter the desired image regions. For example, when StyleFusion is not

Fig. 12. ***Coarse control using StyleFusion on the churches domain***. Similar to Figure 11, we demonstrate that StyleFusion is able to generate multiple church buildings of the same style, but different structure. Notably, observe the similarity in the church layout for each of the columns shown.

used, observe the change in the face shape in the "smile" edit across all examples or the wrinkles that are removed in the "mohawk" edit of the second example. Additionally, observe the change in skin color and eye color when editing the hair albedo in the fourth example. Overall, when StyleFusion is paired with existing editing techniques, the resulting edits are able to better preserve the original head shape and facial identity.

6.2 Local Feature Transfer

A natural extension of the local and global editing presented in Section 5 is allowing one to faithfully transfer a semantic region from a reference image to a given target image. Observe that doing so is especially challenging when the reference and target images are unaligned (e.g., have different poses).

We compare StyleFusion's performance on reference-guided local editing on real images to several recent approaches: StyleMapGAN from [Kim et al. 2021], Retrieve in Style (RIS) from [Chong et al. 2021], and the latent regressor of Chai *et al.* [Chai et al. 2021]. For StyleFusion and RIS, we perform editing in the latent space of the pre-trained StyleGAN generator. For both, we invert the given real image using the e4e encoder from [Tov et al. 2021]. For StyleMapGAN and Chai [2021], we use their official implementations and provided editing interfaces for performing the edits.

Observe that while StyleMapGAN and Chai [2021] require a manual selection of the image region to be transferred, StyleFusion and RIS require no such per-image selection. We refer the reader to Figure 14 for a comparison of the four approaches. First, observe that StyleMapGAN struggles to accurately transfer and realistically blend the source image with the selected reference image region, especially when the images are not aligned (e.g., when transferring the hair in both examples). Second, note that Chai *et al.* [2021] struggle in accurately capturing the desired edit (e.g., the mouth edit in the first example). Compared to these other approaches, RIS which builds on the Editing in Style (EIS) approach from Collins *et al.* [2020] and is a concurrent work of ours, provides improved editing and disentanglement. Yet, their results may lead to some undesirable side-effects in other semantic regions. For example, observe the skewed head shape in the rightmost column of the second row or the unrealistic hair composition in the sixth row when transferring the hairstyle between the

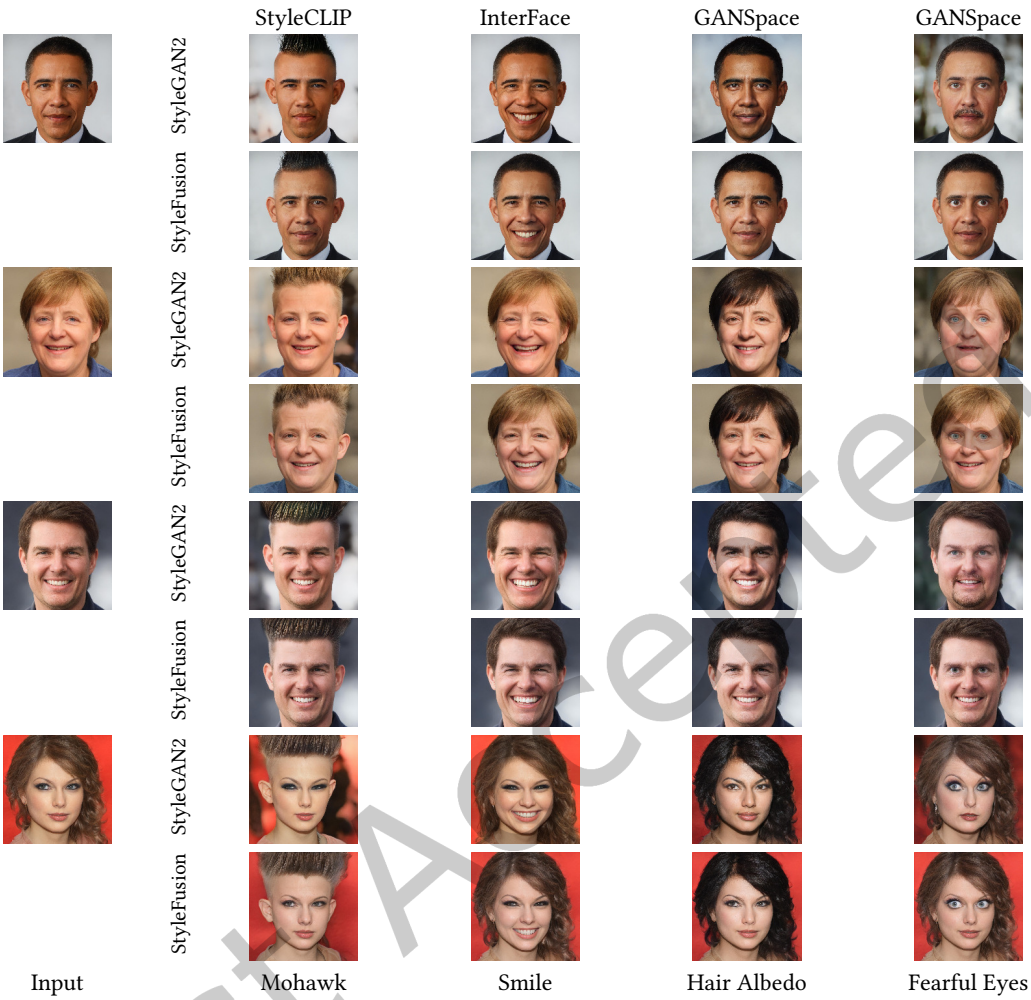


Fig. 13. ***Latent editing techniques performed with and without StyleFusion.*** With StyleFusion, the techniques are applied only to the latent code that controls the relevant region(s). As shown, pairing a latent editing technique with StyleFusion leads to a more disentangled, precise edit. All input images are reconstructions obtained with e4e [Tov et al. 2021]

two unaligned images. Overall, StyleFusion provides improved disentanglement and image composition without requiring any manual supervision at inference time.

Finally, observe that StyleFusion provides several additional advantages over EIS and RIS. First, StyleFusion is faster as it is designed to be part of the forward pass of StyleGAN's synthesis process and does not require any per-image calculations. Additionally, StyleFusion is a generic disentanglement framework that can more naturally be integrated with other editing methods whereas EIS and RIS are dedicated techniques for semantic feature transfer.

6.3 Hair Transfer

We now extend the local feature transfer explored above and focus on the task of *hair transfer*, which has recently received particular attention due to its complexity [Tan et al. 2020; Zhu et al. 2021a]. Here, we wish to transfer

Fig. 14. ***Local feature transfer.*** We compare StyleFusion with three alternative methods for local image editing. For each pair of source image and reference image, we perform three edits: eyes, mouth, and hair. For StyleMapGAN [Kim et al. 2021] and Chai *et al.* [2021], we provide the selected regions for performing the edits as a reference to the reader. StyleFusion and RIS [Chong et al. 2021] receive as input a latent code while StyleMapGAN and Chai *et al.* [2021] receive images. Therefore, for StyleFusion and RIS we pass as input the latent code corresponding to the input image inverted with e4e [Tov et al. 2021] and show the reconstructed image as the source image. For a fair comparison, we pass to StyleMapGAN and Chai *et al.* [2021] the same reconstructed image.

the hair from a reference image onto a given source image. Doing so becomes more challenging when the images are of different poses, genders, and ages.

In Figure 15, we demonstrate StyleFusion's ability to perform such a transfer even in the presence of such variations between the source and reference image. Here, each row depicts the same source individual at different poses with each column representing a different hairstyle to transfer. Observe how StyleFusion is able to faithfully transfer the hairstyle to the different poses (i.e., the hairstyle remains fixed at each column). Notably, observe that the transfer remains accurate even when the reference image is of individuals of different ages and genders.

Fig. 15. ***Hair transfer using StyleFusion.*** Given the source individual (generated, shown to the left) at different poses, StyleFusion is able to transfer the hair style from multiple references (generated, shown in the top row). Observe the hair style consistency along each column and the pose consistency along each row.
Notice also that the identity of the source is well-preserved, showing the disentanglement between the hair and face image regions.

7 ABLATION STUDY

In this section, we provide an ablation study to validate the design choices of StyleFusion. Specifically, we first demonstrate the importance of the global style code tasked with aligning the two input images. We additionally demonstrate the contribution of the multi-step training scheme presented in Section 4.3. For our ablation study, we focus on the cars domains and train a FusionNet tasked with disentangling the car body and image background. We compare our complete StyleFusion model with five variants:

(a) Preserving Car Body & Altering Background (b) Preserving Background & Altering Car Body

Fig. 16. ***Ablation Study.*** We train a single FusionNet to disentangle between the car body and the image background. In each column of 16a we alter the image background, while keeping the car body and pose fixed. Conversely, in 16b, we alter the car body, while keeping the background and pose fixed.

(1) *No Global*: Here, we do not use the global code used for aligning the two input images. As such, the FusionNet is composed only of the LB_{fuse} Latent Blender component (see Figure 4).
(2) *No Stage 1*: Training is performed without the first stage. Observe, that in this setting, the *mask loss* is not used for training the FusionNet.
(3) *No Stage 2*: Training is performed without the second stage where only LB_{fused} is trained.
(4) *No Stage 3*: Training is performed without the third stage where both LB_{fuse} and LB_{align} are trained simultaneously. Observe that L_{local} is used only in this stage.
(5) *End to End*: StyleFusion trained without the multi-step training procedure in an end-to-end fashion (i.e., all losses are optimized simultaneously).

We begin our analysis by referring the reader to Figure 16a. Here, the latent code controlling the car body and the global latent code remains fixed while the latent code controlling the image background is changed across each column. That is, each image along the columns should depict the same car model at the same pose, but different backgrounds. Observe how when the global latent code is not used (in the top row), the car body varies greatly across the four outputs. Comparing our full model to the other variants, notice how StyleFusion is able to better preserve coarse details (e.g., the car build varies in row 4) while better preserving smaller details along the wheels (e.g., the car wheels vary in row 3). Comparing the results shown in the row 2 illustrates the effectiveness of the first stage of training in improving the preservation of small details.

Fig. 17. ***Limitations.*** Although StyleFusion is effective in aligning semantic regions, when given images of extreme poses, the resulting control may result in unwanted artifacts. For example, in the first row we control hair with a latent code that is strongly unaligned with the base code. In the second row, altering the pose of the car may result in skewed car shapes.

In Figure 16b we perform a similar process, but here, the background and global latent codes remain fixed while the latent code controlling the car body is varied. Here, notice the importance of the fusion and localization losses utilized in the second and third stages of training. Specifically, without these losses, the results suffer from a slight mode collapse, mainly with respect to the car's color. In both cases, observe how StyleFusion without the multi-step training process fails to preserve the car body (in Figure 16a) and the background (in Figure 16b).

8 DISCUSSION AND CONCLUSIONS

We have presented a novel approach in which an image is generated from a set of disentangled latent codes, each controlling a single semantic region of the image. We showed that such a generator can be built by pairing a pre-trained generator with a hierarchical mapping network that fuses a set of latent codes into a single one.

While we have demonstrated the competence of StyleFusion in learning a semantically-aware disentanglement of images, several limitations should be considered. First, while the global input code is effective in aligning images to the same spatial layout when the global code represents an image with an extreme pose or under-represented layout, the alignment stage may result in unwanted artifacts, see Figure 17.

While we have demonstrated the locality of StyleFusion's control over semantic regions in Section 5, there may still remain some level of entanglement between different image attributes. This is most prominent when editing more global features such as hairstyle, which may oftentimes be entangled with facial identity.

Our network is trained hierarchically, providing control over the granularity of the learned disentanglement. This implies a sequential top-down training, where the disentanglement of finer features is dependent on the fidelity of disentanglement of coarser semantics.

It should be noted that many of the results presented in the paper can possibly be achieved using previous editing techniques. Yet, our StyleFusion mechanism itself is not an editing method, but a mapping network that facilitates performing local editing over semantic regions by refining existing latent space editing techniques.

The contributions of our work are two-fold. First, we introduce a new approach for designing generators. Instead of feeding the synthesis network a single latent code that controls the image in its entirety, we feed a latent code for each semantic region of the image. The resulting fused, disentangled code facilitates better flexibility over the synthesis process. Second, we propose implementing this design by training a new mapping function that takes a set of latent codes into the latent space of a pre-trained synthesis network.

It should be noted that using a pre-trained generator inherently bounds the disentanglement that can be achieved by our method. A future research direction is to re-design the channels of the generator itself to achieve better disentanglement and control over semantic regions, or to train such a generator that receives several latent codes in an end-to-end fashion.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2stylegan: How to embed images into the stylegan latent space?. In *Proceedings of the IEEE international conference on computer vision*. 4432–4441.

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020a. Image2StyleGAN++: How to Edit the Embedded Images?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8296–8305.

Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. 2020b. StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. arXiv:2008.02401 [cs.CV]

Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. 2021. Labels4Free: Unsupervised Segmentation using StyleGAN. *arXiv preprint arXiv:2103.14968* (2021).

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021a. Only a Matter of Style: Age Transformation Using a Style-Based Regression Model. arXiv:2102.02754 [cs.CV]

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021b. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. arXiv:2104.02699 [cs.CV]

Yazeed Alharbi and Peter Wonka. 2020. Disentangled Image Generation Through Structured Noise Injection. arXiv:2004.12411 [cs.CV]

David Bau, Alex Andonian, Audrey Cui, YeonHwan Park, Ali Jahanian, Aude Oliva, and Antonio Torralba. 2021. Paint by Word. arXiv:2103.10951 [cs.CV]

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. 2019. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics* 38, 4 (Jul 2019), 1–11. https://doi.org/10.1145/3306346.3323023

Lucy Chai, Jonas Wulff, and Phillip Isola. 2021. Using latent space regression to analyze and leverage compositionality in GANs.. In *International Conference on Learning Representations*.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. arXiv:1606.03657 [cs.LG]

Min Jin Chong, Wen-Sheng Chu, Abhishek Kumar, and David Forsyth. 2021. Retrieve in Style: Unsupervised Facial Feature Transfer and Retrieval. arXiv:2107.06256 [cs.CV]

Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. 2020. Editing in Style: Uncovering the Local Semantics of GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5771–5780.

Antonia Creswell and Anil Anthony Bharath. 2018. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems* 30, 7 (2018), 1967–1974.

Rinon Gal, Amit Bermano, Hao Zhang, and Daniel Cohen-Or. 2020. MRGAN: Multi-Rooted 3D Shape Generation with Unsupervised Part Disentanglement. arXiv:2007.12944 [cs.CV]

Rinon Gal, Dana Cohen, Amit Bermano, and Daniel Cohen-Or. 2021. SWAGAN: A Style-based Wavelet-driven Generative Model. arXiv:2102.06108 [cs.CV]

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. 2019. GANalyze: Toward Visual Definitions of Cognitive Image Properties. arXiv:1906.10112 [cs.CV]

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 2672–2680.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. *arXiv preprint arXiv:2004.02546* (2020).

Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2018. AttGAN: Facial Attribute Editing by Only Changing What You Want. arXiv:1711.10678 [cs.CV]

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2018. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. arXiv:1706.08500 [cs.LG]

Sarah Jane Hong, Martin Arjovsky, Darryl Barnhart, and Ian Thompson. 2020. Low Distortion Block-Resampling with Spatially Stochastic Networks. arXiv:2006.05394 [stat.ML]

Xianxu Hou, Xiaokang Zhang, Linlin Shen, Zhihui Lai, and Jun Wan. 2020. GuidedStyle: Attribute Knowledge Guided Style Manipulation for Semantic Face Editing. arXiv:2012.11856 [cs.CV]

Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. 2018. Disentangling factors of variation by mixing them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3399–3407.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR* (2017).

Ali Jahanian, Lucy Chai, and Phillip Isola. 2020. On the "steerability" of generative adversarial networks. arXiv:1907.07171 [cs.CV]

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. 2018. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 805–820.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. In *Proc. NeurIPS*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.

Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser M. Nasrabadi. 2018. Style and Content Disentanglement in Generative Adversarial Networks. arXiv:1811.05621 [cs.CV]

Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. 2021. StyleMapGAN: Exploiting Spatial Dimensions of Latent in GAN for Real-time Image Editing. arXiv:2104.14754 [cs.CV]

Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. 2019. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4422–4431.

Gihyun Kwon and Jong Chul Ye. 2021. Diagonal Attention and Style-based GAN for Content-Style Disentanglement in Image Generation and Translation. arXiv:2103.16146 [cs.CV]

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Fader Networks: Manipulating Images by Sliding Attributes. arXiv:1706.00409 [cs.CV]

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*. 35–51.

Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. 2020. High-Fidelity Synthesis with Disentangled Representation. arXiv:2001.04296 [cs.CV]

Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. 2021. VOGUE: Try-On by StyleGAN Interpolation Optimization. arXiv:2101.02285 [cs.CV]

Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. 2020. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. arXiv:1912.13457 [cs.CV]

Zachary C. Lipton and Subarna Tripathi. 2017. Precise Recovery of Latent Vectors from Generative Adversarial Networks. arXiv:1702.04782 [cs.LG]

Yunfan Liu, Qi Li, Zhenan Sun, and Tieniu Tan. 2020. Style Intervention: How to Achieve Spatial Disentanglement with Style-based Generators? arXiv:2011.09699 [cs.CV]

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. arXiv:1811.12359 [cs.LG]

Michael Mathieu, Junbo Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. 2016. Disentangling factors of variation in deep representations using adversarial training. arXiv:1611.03383 [cs.LG]

Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. 2020. Semi-supervised StyleGAN for disentanglement learning. In *International Conference on Machine Learning*. PMLR, 7360–7369.

Yotam Nitzan, Amit Bermano, Yangyan Li, and Daniel Cohen-Or. 2020. Face Identity Disentanglement via Latent Space Mapping. arXiv:2005.07728 [cs.CV]

Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E Green, and Nassir Navab. 2021. Segmentation in style: Unsupervised semantic image segmentation with stylegan and CLIP. *arXiv preprint arXiv:2107.12518* (2021).

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. 2020. Swapping Autoencoder for Deep Image Manipulation. In *Advances in Neural Information Processing Systems*.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. arXiv:2103.17249 [cs.CV]

Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. 2016. Invertible Conditional GANs for image editing. arXiv:1611.06355 [cs.CV]

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. 2020. Adversarial Latent Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14104–14113.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2020. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. arXiv:2008.00951 [cs.CV]

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *arXiv preprint arXiv:2106.05744* (2021).

Rohit Saha, Brendan Duke, Florian Shkurti, Graham W. Taylor, and Parham Aarabi. 2021. LOHO: Latent Optimization of Hairstyles via Orthogonalization. arXiv:2103.03891 [cs.CV]

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9243–9252.

Yujun Shen and Bolei Zhou. 2020. Closed-Form Factorization of Latent Semantics in GANs. *arXiv preprint arXiv:2007.06600* (2020).

Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. 2020. MichiGAN: Multi-Input-Conditioned Hair Image Generation for Portrait Editing. arXiv:2010.16417 [cs.CV]

Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020a. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. *arXiv preprint arXiv:2004.00121* (2020).

Ayush Tewari, Mohamed Elgharib, Mallikarjun B R., Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020b. PIE: Portrait Image Embedding for Semantic Control. arXiv:2009.09485 [cs.CV]

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. arXiv:2102.02766 [cs.CV]

Andrey Voynov and Artem Babenko. 2020. Unsupervised Discovery of Interpretable Directions in the GAN Latent Space. *arXiv preprint arXiv:2002.03754* (2020).

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Joe H. Ward. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (1963), 236–244. https://doi.org/10.1080/01621459.1963.10500845 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500845

Zongze Wu, Dani Lischinski, and Eli Shechtman. 2020. StyleSpace Analysis: Disentangled Controls for StyleGAN Image Generation. arXiv:2011.12799 [cs.CV]

Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021a. TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. arXiv:2012.03308 [cs.CV]

Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. 2021b. GAN Inversion: A Survey. arXiv:2101.05278 [cs.CV]

Ceyuan Yang, Yujun Shen, and Bolei Zhou. 2020. Semantic Hierarchy Emerges in Deep Generative Representations for Scene Synthesis. arXiv:1911.09267 [cs.CV]

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. 2021. DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort. In *CVPR*.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020b. In-domain gan inversion for real image editing. *arXiv preprint arXiv:2004.00049* (2020).

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*. Springer, 597–613.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. 2021a. Barbershop: GAN-based Image Compositing using Segmentation Masks. arXiv:2106.01505 [cs.CV]

Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. 2021b. Improved StyleGAN Embedding: Where are the Good Latents? arXiv:2012.09036 [cs.CV]

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020a. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 2020). https://doi.org/10.1109/cvpr42600.2020.00515