MixSyn: Learning Composition and Style for Multi-Source Image Synthesis

İlke Demir
Intel
idemir@purdue.edu

Umur A. Çiftci
Binghamton University
uciftci@binghamton.edu

Abstract

Synthetic images created by generative models increase in quality and expressiveness as newer models utilize larger datasets and novel architectures. Although this photorealism is a positive side-effect from a creative standpoint, it becomes problematic when such generative models are used for impersonation without consent. Most of these approaches are built on the partial transfer between source and target pairs, or they generate completely new samples based on an ideal distribution, still resembling the closest real sample in the dataset. We propose MixSyn (read as “mixin’”) for learning novel fuzzy compositions from multiple sources and creating novel images as a mix of image regions corresponding to the compositions. MixSyn not only combines uncorrelated regions from multiple source masks into a coherent semantic composition, but also generates mask-aware high quality reconstructions of non-existing images. We compare MixSyn to state-of-the-art single-source and collage generation approaches in terms of quality, diversity, realism, and expressive power; while also showcasing interactive synthesis, mix & match, and edit propagation tasks, with no mask dependency.

Figure 1. MixSyn learns to generate semantic compositions and styles from multiple sources. Left - From mask (orange) and image (purple) regions, novel compositions and images (green) are generated. Naive copy-paste is shown in red boxes. Right - Each facade (green) is generated from multiple source images for each region in the given mask.

1. Introduction

Image-based synthesis has been an interesting topic for decades in both computer vision and graphics. Recent generative approaches set this task forth as conditional generation [4, 16, 21, 35, 37, 41, 42], image-to-image translation [7, 10, 11, 21, 22, 31, 33, 46], or style encoding [12, 20, 23, 47]. The corner stone of these approaches has been learning the mapping between a source and a target image, for modeling specific styles, segments, or domains. Most of those approaches utilize semantic masks to conditionally generate realistic images [34], to represent diverse inter-domain images [11], and to replace the content or style of specific parts seamlessly [50]. However, all of them operate on given masks of source and target pairs. Some enable sequentially modifying regions with multiple targets, but they need aligned segments with a constant mask.

Being able to incorporate style information through normalization parameters accelerated conditional image generation research, which yields higher quality results [20] as the details are retained deeper in the network. However
this constraint also increased the dependency on the input semantic masks (sometimes also called maps or compositions). Observing the state of the art in semantic image synthesis, three main limitations restrain the expressive power: (1) generation is restricted to source-target (pairwise) transfer of styles and regions, (2) semantic masks are mostly manually modified and they are neither novel, nor flexible, (3) uncorrelated and unaligned regions are compositionally not coherent. For (1), [50] intakes per region style images, however they pursue pairwise processing per region. This is a serious limitation amongst most of the semantic image synthesis approaches as the interactions and contributions from multiple sources are dismissed. For (2), [34] provides a UI for drawing semantic masks. However, the generation is based on the label encoding, and it is not possible to guide the generation with a specific image, sweeping the mask dependency under the hood. For (3), [16] learns mapping and interpolation between masks, however our motivation to transcend pairwise manipulation to multi-source images synthesis poses a different challenge. Moreover, this source-target coupling enables impersonation by deepfakes, which raises serious ethical debates.

To overcome these limitations, we jointly learn semantic compositions and styles from multiple images. We tackle this problem by learning to generate fuzzy semantic compositions from input masks and by learning to synthesize novel photorealistic images from these compositions, preserving the style of each input region. Although humans are comfortable editing existing semantic masks, manual assembly of novel masks from scratch is challenging due to (i) non-exact region boundaries, (ii) unassigned pixels, (iii) overlapping regions, and (iv) misalignment. MixSyn takes as input multiple unaligned segments from several source images (i.e., eyes of A, mouth of B, and nose of C), and creates a new coherent image (i.e., a new face) based on the learned semantic maps (Fig. 1). Our approach

- learns to generate coherent novel compositions, reducing the dependency on semantic regions and increasing the quality;
- couples structure and style generation for image synthesis, flexing spatial constraints on the style generation by learned fuzzy masks; and
- allows combining multiple sources into a photorealistic image, enabling style and structure blending, and disabling impersonation for face generation.

We employ two architectures for generating the composition (semantic) and the image (visual), encoding structures and styles of images separately per region. The structure generator (Fig. 2) learns feasible compositions from as-is, random, and real samples. The style generator (Fig. 4) learns to generate realistic images using region-adaptive normalization layers with generated masks. The two generators are trained jointly in order to couple structure and style creation. We also introduce MS block (Fig. 4e) with optional normalization and resampling layers per module.

We demonstrate and compare our results to single-source sequential editing and collage-based synthesis approaches in terms of similarity, reconstruction, visual, and generative quality. We train and test MixSyn on several datasets in two domains: faces and buildings, with promising results for extension to others. We conduct ablation studies on our region classes and loss functions. Moreover, we implement several applications of MixSyn, such as edit propagation and combinatorial generative space exploration. The multi-source nature of MixSyn also prevents one-to-one impersonations in face domain, which is a positive step towards privacy concerns [43], causing the shift to synthetic datasets [44].

2. Related Work

Patch-based Synthesis. Traditional approaches provide semantically guided synthesis using patch similarity [2], graph cuts exploiting repetitions [25], and guided inverse modeling exploiting instances [13]. Their deep generative counterparts flex similarity and repetition coercion, so the synthesis can be much efficient [27], adaptive [45], complex [39], yielding detailed results [38], due to simplistic part-based similarity [48] and contrastive [33] losses. Inspired by patch-based approaches, we propose a novel semantic image synthesis method where patches are replaced with fuzzy semantic regions, shifting our focus from patch selection to patch composition.

Style Transfer. Recently, popular image manipulation tasks emerge from applying the style of a source image to a target image by adaptive normalization [23], with explicit domain labels [10], utilizing soft masks [37], transferring segment by segment [37], for attribute editing [22], and in multiple domains [11]. In particular for combining multiple sources, [35, 46] blend features in GAN layers of multiple reference images; however the spatial regions and blended features are provided manually. [36] conditions hair generation on multi-input; however masks are kept constant. [5] can translate a collage image to a photorealistic image, but there is no semantic structure and the collage creation is a manual pre-processing step.

Conditional Normalization. As semantic synthesis approaches and conditional GANs start to demand more accuracy and realism, supplying masks only as an input to the first layers did not suffice to preserve the contribution of regions as the network grows deeper. Later, the quality of results has been significantly enhanced by injecting style [20] and structure [34] information in the adaptive normalization layers. [50] took it a step further and introduced region-adaptive normalization, which allows introducing per region styles. Building upon, we introduce MixSyn blocks
(MS-block), a slight modification over [34] with [50] normalization, to broadcast styles per learned regions.

**Semantic Editing.** Another semantic image manipulation direction is to modify or create some binary mask for inclusion/exclusion [21, 32], manipulate the underlying mask [3, 16], generate the mask only with label collections [9], replace foreground objects [6], learn binary compositions [1], use encoder-decoder networks to learn the blending [31], or infill with another image [14] to inpaint the manipulated parts. Although such approaches provide control over semantic labels, (1) generation is not controllable or guided by a certain image, (2) they mostly do inpainting instead of synthesis, and (3) there is no multi-source capability, i.e., all of them utilize source-target pairs. Meanwhile, other approaches push the image-to-image translation to mask-to-mask translation [26], sketch-to-sketch translation [8], or scene graph editing [15], where the new mask contains structure of the source and style of the target. Our approach is conceptually similar, but instead of user-defined masks, the mask is a learned composition of regions from multiple masks.

3. Multi-Source Composition Learning

In order to learn coherent fuzzy compositions from multiple regions as in Fig. 3, first we define our compositions, then we describe our architecture with a multi-encoder, single decoder generator with a simple discriminator (Fig. 2).

3.1. Compositions

Let \( r^*_i \) denote regions making up a source mask \( M_a = \{ r^*_i \} \), in a predetermined order for \( i \leq N \), where \( N \) is the number of all possible regions. \( M \) corresponds to the list of all \( S \) source masks \( M = \{ M_a, M_b, \ldots, M_S \} \). We would like to assemble a composition \( M'_* = \{ r^*_0, r^*_1, \ldots, r^*_N \} \) where each region \( r^*_i \) comes from a source mask \( M_a \) in \( M \). It is important to note that a source mask can be selected multiple times for different regions (\( a, b, \ldots, S \) can repeat), however a region can be selected only once (\( 0, 1, \ldots, N \) is unique). Masks have sharp boundaries between regions, whereas compositions combine fuzzy regions.

Needless to say, if all regions are selected from the same source mask (i.e., \( \forall* = a \)), we expect \( M'_* = \cup_i e^*_i \) to represent \( M_a \). We call this known composition \( M'_* \) for each \( M_* \). In contrast, if each \( r^*_i \) is selected from different \( M_*'s \) in the batch, we call it random compositions \( M''_* \) (Fig. 3 and Supp. A). If a region does not exist in a composition, we set \( e^*_x = [0] \). For the face domain, we select symmetric regions from the same source, e.g., left/ right eyes from one mask (see Supp. G for symmetry coupling), to keep random compositions consistent.

3.2. Structure Generator

There is no initial alignment between regions of random compositions, so it does not make sense to put many random regions into same composition in image space. However, we want to learn how they would transform and blend to create realistic compositions, thus we encode each \( r^*_i \) with the specific region encoder \( E_i(r^*_i) = e^*_i \), producing a \( 16 \times 16 \times 128 \) structure code. We use separate encoders, so that the codes are disentangled and each region can be used interchangeably. Then, we combine structure code \( e^*_i \)’s into a composition code \( e_* = \bigoplus_i e^*_i \) of size \( 16 \times 16 \times 128 \times N \) and pass it to the decoder \( C_* \). \( C_* \) learns to decode \( e_* \) into...
novel compositions $M_s$. Our structure generator forges soft borders (i.e., fuzzy compositions), which creates flexibility for our image generator to produce better results.

The encoder-decoder structure constitutes our structure generator $G_M(M_s) = C_s(\bigoplus_{i \in M_s} E_i(r_i^s))$, which is trained with our discriminator $D_M$ to create coherent and realistic masks. The encoder, decoder, and discriminator models use MS blocks (Sec. 4.2.1 and Fig. 4e) and the layer details are documented in Fig. 2 and Supp. B.

3.3. Training Objectives

During training, generator $G_M$ takes a mask $M_s$ and learns to create known compositions $M'_s$ and random compositions $M''_s$ with an adversarial loss. We also add $R_1$ regularization for training stability [29]. The discriminator $D_M$ (Fig. 2b, Supp. B) aims to classify real masks $M$, generated known compositions $G_M(M'_s)$ and generated random compositions $G_M(M''_s)$ (Fig. 2c, Supp. B). With a batch size of $\omega$, the discriminator processes $\omega$ reals and $2\omega$ fakes, thus we balance the contributions in the loss function.

$$
L_A = \log D_M(M_s) + \alpha \log(1 - D_M(G_M(M'_s))) + (1 - \alpha) \log(1 - D_M(G_M(M''_s)))
$$

For known compositions, we incorporate an L1 reconstruction loss $L_R = \| M_s - G_M(M'_s) \|_1$, forming final objective:

$$
\min_{G_M} \max_{D_M} \lambda_A L_A + \lambda_R L_R
$$

4. Multi-Source Image Synthesis

We continue with image generation from compositions, where each segment preserves its style. To help the reader, regions in masks are analogous to segments in images.

4.1. Style Segments

Let $k^a_i$ denote segments making up a source image $I_a = \{k^a_i\}$, with corresponding mask regions $M_a = \{r^a_i\}$. $I$ corresponds to the list of all $S$ source images $I = \{I_a, I_b, \ldots, I_S\}$. We would like to assemble an image $I'_s = \{k^a_1, k^b_1, \ldots, k^c_1\}$ where each region $r^s_i$ corresponding to the segment $k^s_i$ comes from a mask $M_s$ in $M$. The concept can be observed in red copy-paste segments in Fig. 1.

We utilize three types of segments: (1) $\{k^a_i\}$ from the initial mask $M$, (2) $\{k^b_i\}$ from $\{r^b_i\}$ in the generated known composition $G_M(M'_s)$, and (3) $\{k^c_i\}$ from $\{r^c_i\}$ in the generated random composition $G_M(M''_s)$. We expect $\bigcup_i E_i(k^a_i)$ to represent $I_a$, and $\bigcup_i E_i(k^b_i)$ to approximate $I'_s$. While these two segment generations ensure learning plausible photorealistic images from compositions, the last one $\bigcup_i E_i(k^c_i)$ is the actual novelty that brings out the style blending from multiple source images. This is also depicted as the main application in Fig. 5.

Similar to the structure generator, we encode each $k^a_i$ with the specific segment encoder $E_i(k^a_i) = e^a_i$, producing a $\delta$-length style code (Fig. 4b). Then we combine the style codes to construct a style matrix $\Delta_s = \bigoplus e^a_i$ of size $\delta \times N$. Encoder layers are listed in Fig. 4a and in Supp. B.

4.2. Image Generator

We use a full generator with adaptive normalization layers for image synthesis $G_I(M_s, \Delta_s) = I_s$ (Fig. 4c), which is trained with our image discriminator $D_I$ (Fig. 4d) to create realistic images. Supp. B delineates all architectures.

4.2.1 MS Block

To selectively include normalization and sampling layers throughout our architecture, we introduce our minimum
computation unit: MS block (Fig. 4e). MS is a configurable res block with a shortcut, with optional downsampling (red layers in Fig. 2), upsampling (blue layers in Fig. 4c), and normalization (purple boxes in Fig. 4c) layers. Samplings are done with bilinear interpolation and average pooling. For encoders and structure decoder, instance normalization is enabled in MS block. For broadcasting styles per learned regions, we use region-adaptive normalization [50] with corresponding masks and style matrices. Layer order in MS block follows pre-activation residual units in [11, 17].

4.2.2 Training Objectives

Adversarial Loss. Our image generator $G_I$ intakes source images $I$ and compositions $M_x$, $G_M(M'_x)$ and $G_M(M''')$, outputting known $G_I(I_x, M_x)$, approximated $G_I(I_x, G_M(M'_x))$, and random images $G_I(I_x, G_M(M''''))$. The discriminator $D_I$ (Fig. 4d and Supp. B) classifies these images as real or fake using loss 3, balancing contributions of real and three subsets of fake images. We add $R_1$ regularization for training stability [29].

$$L_A = \beta \log D_I(I_x) + (1 - \beta) [\eta \log(1 - D_I(G_I(I_x, M_x))) + \log(1 - D_I(G_I(I_x, G_M(M'_x)))) + (1 - \eta) \log(1 - D_I(G_I(I_x, G_M(M''''))))]$$

Note that, initial $r_i^*$s from different $M_x$s that are combined in $M'''$ are stored in order to evaluate the corresponding $k_i^*$s in $I_x$. Although the region-adaptive normalization layers need $\Delta$, we push the extraction of the style matrix per composition, to better fill approximate masks.

Style Loss. We add loss 4 based on style matrix $\Delta_s = \bigoplus_i E_i(k_i^*)$ to ensure that the style is preserved for segments of approximated and random images that undergo some transformation.

$$L_S = \frac{1}{2N} \left( \| \bigoplus_i e_i^x - \bigoplus_i E_i(G_I(I_x, G_M(M'_x))) \| + \right)$$

$$\| \bigoplus_i e_i^s - \bigoplus_i E_i(G_I(I_x, G_M(M'''')) \|$$

Reconstruction Loss. Similar to the structure generator, we incorporate a reconstruction loss for the known and approximated images, as they originate from the same image.

$$L_R = \frac{1}{2} (\| I_x - G_I(I_x, M_x) \|_1 + \| I_x - G_I(I_x, G_M(M'_x)) \|_1)$$

Formulating a piecewise continuous local reconstruction loss (like [16]) for random images is left for future work.

Overall, our training can be formulated as below, with the corresponding hyperparameters for each loss term.

$$\min_{G_I, E} \max_{D_I} \lambda_A L_A + \lambda_S L_S + \lambda_R L_R$$

5. Results

We set 0.0001 and 0.0003 for the learning rates of $G_M$, $G_I$ and $D_M$, $D_I$, using ADAM [24] with $\beta_1 = 0$ and $\beta_2 = 0.999$ with a decay of 0.0001. Similar to other normalization approaches [34, 50], we apply Spectral Norm [30] to generators and discriminators. We use instance and region-adaptive normalization for specified layers (see Supp. B). Experiments are done on an NVIDIA RTX 2080 with 4 GPUs/We use 30000 images in CelebAMask-HQ [26] for most of the experiments in face domain, Helen [28] for cross-dataset evaluation, and CMP Facade dataset [40] for results in architecture domain. We use SSIM [49], RMSE [19], PSNR [19], and FID [18] scores for quantitative evaluations and comparisons.

5.1. Evaluation

Fig. 5 demonstrates our main purpose. If selected regions (orange) are to be naively copy-pasted, bottom left mask-image pair is obtained, which is not desirable. In contrast, our approach is able to combine six segments from six images into a coherent composition and image (bottom right). Fig. 13 shows examples of generated composition and image pairs for buildings with 4+ different sources, whereas Fig. 14 and 1 demonstrate results (green) for faces, as a seamless combination of purple segments from 3, 4, and 5 source images. Note that purple boxes are not exact, they only mark the region which is actually represented with the orange masks (e.g., box on a head represents all hair in orange hair region). The copy-paste versions are only demonstrated as examples, they are not used in MixSyn.

Figure 5. Multi-Source Synthesis. MixSyn creates a composition and image (right) from six regions (orange) of six segments (mid).

We evaluate our method quantitatively in Tab. 1 with different image similarity metrics applied per region. We also document region-based scores in Supp. C-E, revealing that our bottleneck is to learn hair styles (highest FID). MixSyn realistically generates frequent segments (eyes,
nose, mouth) with high SSIM and PSNR, however similarity scores of rare ones (hat, glasses) are much lower.

| Method            | SSIM  | RMSE  | PSNR  | FID    |
|-------------------|-------|-------|-------|--------|
| Pix2PixHD [41]    | 0.68  | 0.15  | 17.14 | 23.69  |
| SPADE [34]        | 0.63  | 0.21  | 14.30 | 22.43  |
| SEAN [50]         | 0.7   | **0.12** | 18.74 | 17.66  |
| MixSyn Str        | 0.97  | 1.15  | 33.06 | 18.13  |
| MixSyn            | **0.95** | 1.89  | **31.32** | 14.41  |
| MixSyn Str (H)    | 0.98  | 0.92  | 36.00 | NA     |
| MixSyn (H)        | 0.96  | 1.46  | 32.13 | NA     |

Table 1. **Reconstruction Scores** on CelebAMask-HQ and Helen (H) datasets. Non-MixSyn scores are taken from [50].

Finally, we perform a cross-dataset evaluation and test MixSyn trained on CelebAMask-HQ on Helen (Fig. 5 (H)). High similarity indicates that MixSyn is generalizable to create multi-source faces from other datasets. Relatively lower RMSE signals that we indeed create novel (fuzzy) masks with inexact reconstructions where multiple regions adapt and blend. Supp. E declares all cross-dataset scores.

### 5.2. Comparison

As MixSyn is the first of its kind, we compare our results to single-source [11, 34], sequential multi-source [16, 50], and collage-based [5] approaches. These approaches (i) cannot generate from multiple sources simultaneously, (ii) depend on given/modified mask, (iii) cannot compose novel masks, (iv) do not learn BOTH structure and style end-to-end, and (v) cannot generate from partial or fuzzy masks.

We start with justifying these claims. As per (i-ii), we feed four combinations of (copy-paste/our) x (mask/image) pairs in Fig. 5 as alternative inputs to SPADE [34], SEAN [50] and Mask Guided CGAN [16]. Although results improve from copy-paste masks (Fig 6, col. 1 & 3) to our generated masks (2 & 4), quality of their results are not close to ours (Fig 5), supporting (ii-iii). We also investigate how others reconstruct our result image with our mask (col. 4). Because (iv-v) above, they simply cannot.

Figure 6. **Comparison.** Output pairs in Fig. 5 bottom, are fed to [16, 34, 50]. None can generate from multi-source, create non-existing masks, or output realistic results similar to originals.

In Fig. 7, we select a base mask (top left in Fig. 5) because of (ii-iii), and swap segments following the sequential (hence (i)) component transfer applications of [16, 50], shown in the first two columns. Despite looking better than col. 1-4 in Fig. 6, it is akin to a blended copy-paste, which creates the zoomed-in artifacts (e.g., different neck and nose colors, and shadow mustache), because they are not jointly composing new masks and generating new images as MixSyn does (iv-v). For the third column, we use the copy-paste image as the collage for [5] input. Although there are less visual artifacts compared to the other two, there are empty areas and inconsistent hair style within the segment. The difference in similarity scores also proves that styles per regions are not as-well preserved as ours, questioning the realism over fidelity of [5].

Quantitative comparison of MixSyn also supports and generalizes these claims. In Table 1, we list SSIM, RMSE, PSNR, and FID scores of Pix2PixHD [41], SPADE [34], SEAN [50], our structure generator, and overall MixSyn architecture on CelebAMask-HQ dataset [26]. Although our reconstruction is not as exact (worse RMSE), our generator network is better (better FID). We note that from our compositions to our images, similarity decreases (better SSIM and PSNR for MixSyn Str) as expected, but our style generator exploits novel compositions and achieves a better FID. We list same metrics for the example in Fig. 7, which are

Figure 7. **Sequential/Collage Comparison.** Component transfer causes artifacts on nose and neck for [16], and a ghost mustache for [50]. Collage synthesis [5] creates color artifacts and empty areas. See Supp. F for detailed image scores.
5.3. Experiments

Fig. 8 demonstrates and documents the contribution of each loss function. With only adversarial loss, we generate some humans fitting to compositions, but neither color, nor style, and not even the domain is preserved. Without reconstruction loss, we are able to mimic the style, but the colors are off. Without style loss, we lose patterns of each region, e.g., curly hair is ironed, even though they are from the same region. Finally, without region adaptive normalization, style of small regions are dominated (e.g., eyes). The dataset scores below are computed similar to Tab 1, but on the results generated with the specific loss functions.

Figure 8. Ablation Study. Samples of source and known image with different losses, followed by dataset scores for each setting.

Another key construct is the selection of region types. 18 base types and their hierarchy are mostly known [26] (Fig. 9). However, for our problem, we experiment (i) without symmetry coupling for random compositions, and (ii) with 5 types only, before we converge on (iii) meta types. Early signs for detecting fakery were broken symmetries, such as mismatched eyes and brows. To enforce learning correlation of symmetric regions, we couple left-right indices in random compositions (Fig. 9, blue) for faces. Similarly for buildings, we coupled windows, cornices, and sills together for preserving patterns in random compositions. We experimentally validate that it is better than putting them into same channel. Fig. 10 shows results without symmetry coupling. Although they look realistic at a first glance, different eye colors, gaze directions, and eyebrow styles give away their synthetic nature, shifting faces to the uncanny valley. When those regions are selected randomly without following the same pattern in buildings, less dominant classes such as cornicles and sills start to appear as phantoms on the buildings, as shown in the zoom ins. We also tried compact subtypes of face regions (6 yellow in Fig. 9). We expected style generator to fill in rare types such as necklaces, hat, etc. Instead, structure network merged them to existing types, creating interesting compositions (Supp. H).

6. Applications

6.1. Combinatorial Diversity

Each row in Fig. 11 demonstrates combinations of different regions (mouth, hair, etc.) from similar sets of reference images (color-coded pairs), to create visually varying faces (green). As we can create an exponentially diverse set of combinations, we claim that such a combinatorial design space enables interactive editing systems, simulations with synthetic collections, and data augmentation for DNNs.
6.2. Edit Propagation

In Fig. 12, we start by generating an image given a set of segments, such as \{mouth, nose, eye_l, eye_r\}. Then, we change one or multiple segments with other known or suggested ones. Observe that other segments are structurally and stylistically preserved at each step, while the specified segments are changed according to an unseen reference. In the last step, the face in the first image is given as reference to change the face, creating a very similar face as the region features are preserved throughout the edits.

7. Discussion & Limitations

While generating an image, not all structure codes are needed, e.g., there is no cloth in input regions of Fig. 5, so \(e^*_{\text{cloth}} = [0]\). As there is some fuzziness between hair/face regions in third and last columns, both structure and style generators can recover it in a random composition, and place cloth region in mask, and cloth segment in image. On the other hand, some random combinations cause edge cases naturally, such as a random combination of face region from an image with hair and hair region on the sides from a bald person (Supp. H). An interactive editing system can aid in eliminating such random combinations. As a synthesis approach, MixSyn cannot be used for analysis of data for harming populations. It can only create novel samples based on the training datasets or editing operations. Furthermore, it is almost guaranteed that the synthesized image is either a combination of parts from multiple images, or the same image; thus, it cannot be used for retargeting/reanimation/impersonation of existing people, causing misinformation. As a positive impact, we hope that our approach can spearhead anonymization efforts for sensitive data, when only a part of an image is needed.

8. Conclusion and Future Work

We introduce mixed synthesis (MixSyn) for generating photorealistic images from multiple sources by learning semantic compositions and styles simultaneously. We train structure and style generators end-to-end, while preserving details by adaptive normalization on learned regions. We introduce a flexible MS block as the unit of processing for semantic synthesis. We demonstrate our results on three datasets and two domains, report our FID, SSIM, RMSE, and PSNR scores, qualitatively and quantitatively compare to prior work, and propose novel applications.

We observe that controlled synthesis with multiple images brings a new dimension to expressive creation. Our approach helps create non-existing avatars or architectures. It enables partial manipulation, region transfer, and combinatorial design without mask editing. Anonymization and de-identification are also facilitated by MixSyn. Finally, with the proliferation of adaptive normalization, multi-source synthesis will bloom, foreseeing MixSyn as a pioneer.
Figure 14. **Additional Results.** Purple-highlighted segments in the first 3, 4, or 5 columns are used to synthesize new images (green).

**References**

[1] Samaneh Azadi, Deepak Pathak, S. Ebrahimi, and Trevor Darrell. Compositional gan: Learning image-conditional binary composition. *International Journal of Computer Vision*, pages 1–16, 2020. 3

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3), July 2009. 2

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 38(4), July 2019. 3

[4] Navaneeth Bodla, Gang Hua, and Rama Chellappa. Semi-supervised fusedgan for conditional image generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1

[5] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. In *International Conference on Learning Representations*, 2021. 2, 6

[6] Bor-Chun Chen and Andrew Kae. Toward realistic image compositing with adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[7] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. DeepFaceDrawing: Deep generation of face images from sketches. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2020)*, 39(4):72:1–72:16, 2020. 1

[8] Zhuo Chen, Chaoyue Wang, Bo Yuan, and Dacheng Tao. Puppeteergan: Arbitrary portrait animation with semantic-aware appearance transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[9] Yen-Chi Cheng, Hsin-Ying Lee, Min Sun, and Ming-Hsuan Yang. Controllable image synthesis via segvae. In *European Conference on Computer Vision*, 2020. 3

[10] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 5, 6

[12] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[13] Ilke Demir and Daniel G. Aliaga. Guided proceduralization: Optimizing geometry processing and grammar extraction for
architectural models. *Computers & Graphics*, 74:257 – 267, 2018. 2

[14] Qiyao Deng, Jie Cao, Yunfan Liu, Zhenhua Chai, Qi Li, and Zhenan Sun. Reference guided face component editing. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 502–508. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. 3

[15] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[16] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 5, 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016. 5

[18] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 5

[19] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. 5

[20] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2

[21] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *European Conference on Computer Vision*, pages 17–34, Springer, 2020. 1, 3

[22] David K. Han Jeong gi Kwak and Hanseok Ko. Caffe-gan: Arbitrary face attribute editing with complementary attention feature. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[23] 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 (CVPR)*, June 2019. 1, 2

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 5

[25] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003. 2

[26] 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 (CVPR)*, June 2020. 3, 5, 6, 7

[27] Joo Ho Lee, Inchang Choi, and Min H Kim. Laplacian patch-based image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2727–2735, 2016. 2

[28] Jipeng Lin, Hao Yang, Dong Chen, Ming Zeng, Fang Wen, and Lu Yuan. Face parsing with roi tanh-warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5654–5663, 2019. 5

[29] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018. 4, 5

[30] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 5

[31] J. Naruniec, L. Helmingher, C. Schroers, and R.M. Weber. High-resolution neural face swapping for visual effects. *Computer Graphics Forum*, 39(4):173–184, 2020. 1, 3

[32] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. *arXiv preprint arXiv:2004.04977*, 2020. 3

[33] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Ian-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 319–345, Cham, 2020. Springer International Publishing. 1, 2

[34] 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 (CVPR)*, June 2019. 1, 2, 3, 5, 6

[35] Ryohi Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable image synthesis with internal representation collaging. *arXiv preprint arXiv:1811.10153*, 2018. 1, 2

[36] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. MichiGAN: Towards diverse and interactive facial image manipulation. In *European Conference on Computer Vision*, pages 242–257, Springer, 2020. 2

[37] Hao Tang, Dan Xu, Yan Yan, Philip H.S. Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[38] Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Arbitrary style transfer using neurally-guided patch-based synthesis. *Computers & Graphics*, 87:62–71, 2020. 2

[39] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. Retrievegan: Image synthesis via differentiable patch retrieval. In *European Conference on Computer Vision*, pages 242–257, Springer, 2020. 2
[40] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In Proc. GCPR, Saarbrucken, Germany, 2013. 5

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1, 6

[42] Zongwei Wang, Xu Tang, Weixin Luo, and Shenghua Gao. Face aging with identity-preserved conditional generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 1

[43] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. This person (probably) exists. identity membership attacks against gan generated faces. arXiv preprint arXiv:2107.06018, 2021. 2

[44] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 3681–3691, October 2021. 2

[45] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 2

[46] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In Proceedings of the European Conference on Computer Vision (ECCV), September 2018. 1, 2

[47] Huihuang Zhao, Paul L. Rosin, and Yu-Kun Lai. Automatic semantic style transfer using deep convolutional neural networks and soft masks. CoRR, abs/1708.09641, 2017. 1, 2

[48] Haitian Zheng, Haofu Liao, Lele Chen, Wei Xiong, Tianlang Chen, and Jiebo Luo. Example-guided image synthesis using masked spatial-channel attention and self-supervision. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV, volume 12359 of Lecture Notes in Computer Science, pages 422–439. Springer, 2020. 2

[49] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004. 5

[50] 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 (CVPR), June 2020. 1, 2, 3, 5, 6