MixSyn: Compositional Image Synthesis with Fuzzy Masks and Style Fusion

İlke Demir
Intel Labs
ilke.demir@intel.com

Umur Aybars Çiftçi
Binghamton University
uciftci@binghamton.edu

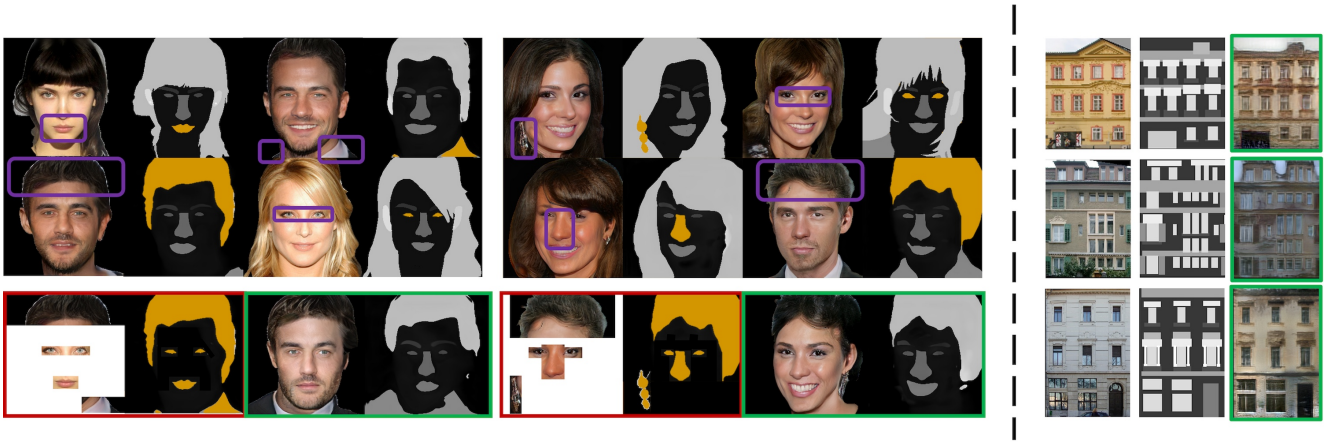


Figure 1. MixSyn learns to generate semantic compositions and styles from multiple sources. Left - From mask (orange) and image (purple) regions, novel compositions and images (green) are generated. Naive copy-paste is shown in red boxes. Right - Each facade (green) is generated from multiple source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

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 benefit from a creative standpoint, expressiveness is still limited by the training data. Most of these approaches are built on the transfer between source and target pairs, or they generate completely new samples based on an ideal distribution, still resembling the closest real sample while missing less frequent or non-existent compositions. We propose MixSyn (read as " mixin' ") to learn novel fuzzy compositions from multiple sources and to create 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 sequential generation and collage generation approaches in terms of quality, diversity, realism, and expressive power; comparing region-wise reconstruction and similarity scores. We also showcase interactive synthesis, mix & match, design space exploration, and edit propagation tasks, with no mask dependency.

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, 34, 46], or style encoding [12, 20, 23, 47]. The foundation 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 [33], 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 (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, and (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), [33] 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 synthesis poses a different challenge. Moreover, the core maleficence of deepfakes arises from impersonation, which comes by default because of this source-target coupling, which raises serious ethical debates. In an attempt to generate *human soups*, MixSyn creates *non-existing synthetic images* by design, as an example of responsible deepfakes.

To overcome these limitations, we jointly learn 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 from a different image. Although humans comfortably edit existing semantic masks, manual assembly of novel masks 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 **novel compositions**, reducing the dependency on exact semantic regions;
- **couples structure and style fusion** for image synthesis, flexing spatial constraints on the style generation by learned fuzzy masks; and
- allows combining **multiple unaligned sources** into a realistic image, enabling style and structure blending, and disabling impersonation for face generation.

We employ two generative architectures for generating the composition (semantic) and the image (visual), encoding structures and styles of images separately per region. Structure generator (Fig. 3) learns feasible compositions from as-is, random, and real samples. Style generator (Fig. 4) learns to generate realistic images using region-adaptive normalization on the novel compositions. 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.

We 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 semantic classes and loss functions. Moreover, we implement applications of MixSyn, such as edit propagation and combinatorial generative space exploration. The multi-source nature of MixSyn also prohibits one-to-one impersonation, which is a positive step towards privacy concerns [43], causing the shift to use 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 [34] 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 style of a source image to a target image by adaptive normalization [23], with explicit domain labels [10], utilizing soft masks [47], 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 images; however spatial regions and 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 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 [33] information to adaptive normalization layers. [50] took it a step further and introduced region-adaptive normalization, which allows introducing per-region styles. Following this line of thought, we introduce MixSyn blocks (MS-block), that contains a novel shape and style aware semantic region adaptive normalization, to broadcast styles per *learned* regions.

Semantic Editing. Semantic image manipulation techniques 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]. 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 image-to-image translation to mask-to-mask [26], sketch-to-sketch [8], or scene graph [15] translation, where the new mask contains structure of the source and style of the target. Our approach is conceptually similar, instead of user-defined masks, the mask is *also learned* from multiple unaligned regions.

3. Multi-Source Composition Learning

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

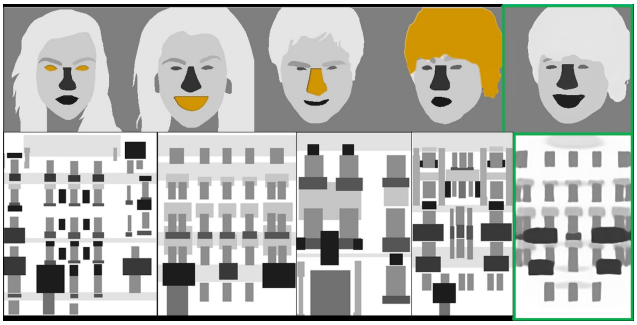


Figure 2. **Compositions.** Faces: Orange regions (r_j^i) generate random composition M'' (green). Buildings: Respectively; molding, window, facade, and balcony regions are combined for the fuzzy mask (green). More samples are demonstrated in Supp. A.

3.1. Compositions

Let r_i^a denote regions making up a source mask $M_a = \{r_i^a\}$, 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 $M'_* = \{r_0^a, r_1^b, \ldots, r_N^c\}$ where each region r_i^* comes from a source mask M_* 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 from one source mask (i.e., $\forall * = a$), we expect $M'_a = \cup_i r_i^a$ 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* composition M'' (Fig. 2 and Supp. A). We select symmetric regions from the same source, e.g., left/right eyes from one mask (Supp. H, symmetry coupling), keeping 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 the 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 $c_* = \bigoplus_i e_i^*$ of size $16 \times 16 \times 128 \times N$ and pass it to the decoder C_* ($\bigoplus$ corresponds to concatenation). C_* learns to decode c_* into novel composition M'_*. Our structure generator forges soft borders (i.e., fuzzy compositions), which creates flexibility for our image generator to produce better results. If a region j does not exist in a composition, we set $e_j^* = [0]$.

The encoder-decoder structure constitutes our structure generator $G_M(M_*) = C_*(\bigoplus_{r_i^* \in M_*} E_i(r_i^*))$, 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 & Fig. 4e) and layer details are listed in Fig. 3 & Supp. B.

3.3. Training Objectives

During training, generator G_M takes a mask $M_x \in M$ and learns to create known compositions M'_x and random compositions M'' with an adversarial loss. The discriminator D_M (Fig. 3c, Supp. B) aims to classify real masks M, generated known compositions $G_M(M'_x)$ and generated random compositions $G_M(M'')$ (Fig. 3b, Supp. B). With batch size ω, the discriminator processes ω reals and 2ω fakes, thus, we balance the contributions with α.

$$\begin{aligned} L_A = & log D_M(M_x) + \\ & \alpha log(1 - D_M(G_M(M'_x))) + \\ & (1-\alpha) log(1 - D_M(G_M(M''))) \end{aligned} \tag{1}$$

For known compositions, we incorporate an L1 reconstruction loss $L_R = ||M_x - G_M(M'_x)||_1$, forming the final objective with hyperparameters λ_A and λ_R as:

$$\min_{G_M} \max_{D_M} \lambda_A L_A + \lambda_R L_R \tag{2}$$

4. Multi-Source Image Synthesis

Having fuzzy compositions, now we learn to create coherent images with them, where styles per segments are pre-

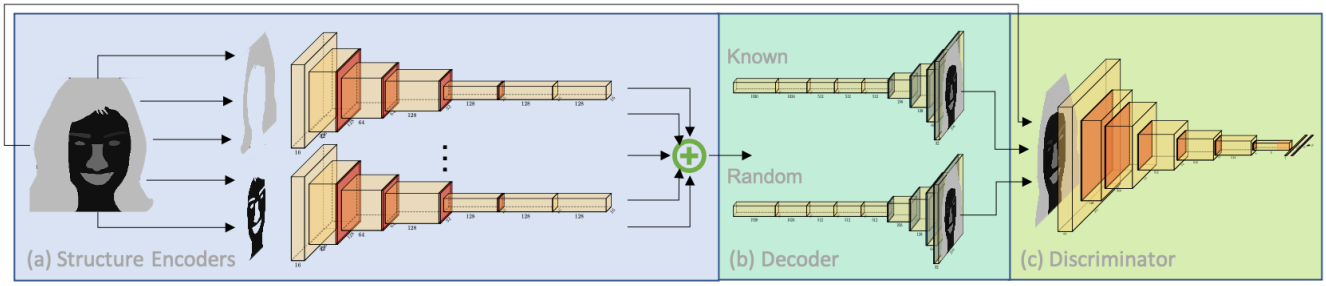


Figure 3. **Structure Generator.** (a) We train separate encoders for each region type, then (b) the structure codes for known and *random* compositions are passed to the decoder. (c) The generator and the discriminator learns to create novel compositions.

served. To aid the reader, *regions* r_i^a in a *mask* M_a are analogous to *segments* k_i^a in an *image* I_a.

4.1. Style Segments

Let k_i^a denote segments making up a source image $I_a = \{k_i^a\}$, with corresponding mask regions $M_a = \{r_i^a\}$. 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 $I'_* = \{k_0^a, k_1^b, \ldots, k_N^c\}$ where each region r_i^* corresponding to the segment k_i^* comes from a mask M_* in M. The concept can be observed in red copy-paste segments in Fig. 1.

Similar to the two types of compositions, now we have three types of images we aim to generate: (1) Known image I_a with initial mask M_a where $\{k_i^a\}$ correspond to $\{r_i^a\}$, (2) approximated image I'_a with the generated known composition $G_M(M'_a)$ where $\{k_i'^a\}$ correspond to $\{r_i'^a\}$,and (3) generated random image I'' with the generated random composition $G_M(M'')$ where $\{k_i''\}$ correspond to $\{r_i''\}$, meaning that all regions and corresponding segments are selected from different source masks and images.

We encode each k_i^* with the specific segment encoder $E_i(k_i^*) = e_i^*$, producing a δ-length style code (Fig. 4b). We expect $\cup_i E_i(k_i^a)$ to represent I_a, and $\cup_i E_i(k_i'^a)$ to approximate I_a. While these two segment generations ensure learning plausible photorealistic images from compositions, the last one ($\cup_i E_i(k_i'')$) is the actual novelty that brings out the style blending from multiple source images, also depicted as the main application in Fig. 5. We combine all style codes to construct a style matrix $\Delta_* = \bigoplus_i e_i^*$ of size $\delta \times N$. For encoders (documented in Fig. 4a and in Supp. B.), not having shared parameters increases memory footprint, however, it enables (1) generating from a varying number of regions, (2) handling non-existing region types, (3) portability across datasets, and (4) better reconstruction that preserves size/shape per region, as discussed by [50].

4.2. Image Generator

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

To selectively include normalization and sampling layers, 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. 3), 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 region, we use region-adaptive normalization with corresponding masks, style matrices, and noise vectors. Layer order in MS block follows pre-activation residual units in [11, 17].

4.3. Training Objectives

Adversarial Loss. Our image generator G_I intakes source images I_x 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 $G_I(I_*, G_M(M''))$. 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 by β and η.

$$\begin{aligned} L_A = & \beta \log D_I(I_x) + \\ & (1-\beta)\,[\eta(\log(1 - D_I(G_I(I_x, M_x))) + \\ & \qquad \log(1 - D_I(G_I(I_x, G_M(M'_x))))) + \\ & \qquad (1-\eta)\log(1 - D_I(G_I(I_*, G_M(M''))))] \end{aligned} \tag{3}$$

Note that, initial r_i^*s from different M_*s that are combined in M'' are stored in order to evaluate the corresponding k_i^*s in I''. Although region-adaptive normalization layers need Δ, 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 $\Delta_* = \bigoplus_i E_i(k_i^*)$ to ensure that the style is preserved for segments of approximated and random images that undergo

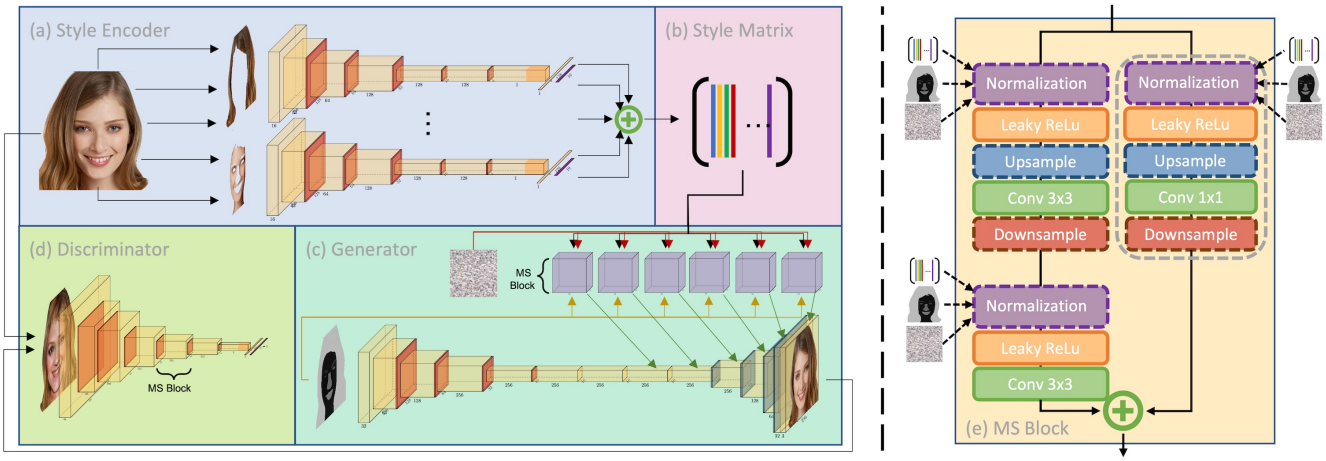


Figure 4. **Style Generator.** (a) We train N encoders for each segment type, and (b) create a style matrix. (c) Style generator translates the masks created by the structure generator into photorealistic images, using a region-adaptive normalization to broadcast the style matrix. (d) Discriminator differentiates between four types of incoming images. (e) The MS Block is the unit of processing for all of our networks, which is a res block with optional resampling and normalization layers.

some transformation.

$$L_S = \frac{1}{2N}(||\bigoplus_i e_i^x - \bigoplus_i E_i(G_I(I_x, G_M(M'_x)))||+ \quad (4)$$
$$||\bigoplus_i e_i^* - \bigoplus_i E_i(G_I(I_*, G_M(M''_x)))||)$$

Reconstruction Loss. Similar to our structure generator, we add a reconstruction loss for known and approximated images, as they originate from the same image. Formulating a piece-wise continuous local reconstruction loss (like [16]) for random images is left for future work.

$$L_R = \frac{1}{2}(||I_x - G_I(I_x, M_x)||_1 + ||I_x - G_I(I_x, G_M(M'_x))||_1) \quad (5)$$

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

$$\min_{G_I,E} \max_{D_I} \lambda_A L_A + \lambda_S L_S + \lambda_R L_R \quad (6)$$

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 [33, 50], we apply Spectral Norm [30] to generators and discriminators. We use instance and region-adaptive normalization for specified layers (see Supp. B). We also add ℓ_1 regularization for training stability [29] in both structure and style generators. We balance the loss contributions with $\alpha = 0.5$ in Eqn. 1, $\lambda_A = 1, \lambda_R = 0.25$ in Eqn. 2, $\beta = 0.25, \eta = 0.6$ in Eqn. 5, and $\lambda_A = 0.25, \lambda_S = 0.4, \lambda_R = 0.3$ in Eqn. 6. Experiments are done on an NVIDIA RTX 2080 with 4 GPUs, with one epoch taking a few hours. We use 30000 images in CelebAMask-HQ [26] for most of the experiments in face domain and Helen [28] for cross-dataset evaluation. We use CMP Facade dataset [40] for the results in architecture domain.

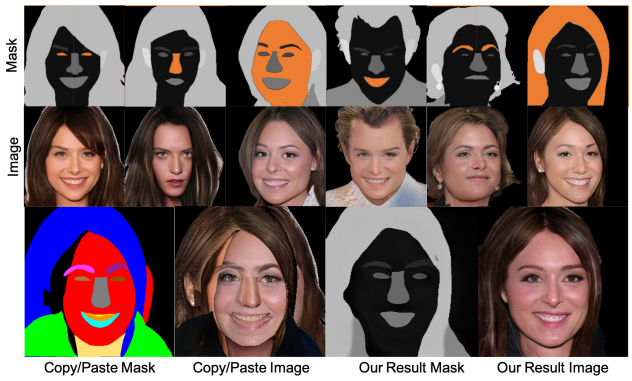


Figure 5. **Multi-Source Synthesis.** MixSyn creates a composition and an image (bottom right) from six regions (orange, top) and six segments (mid). Copy/paste versions are also shown (bottom left).

5.1. Evaluation

Fig. 5 demonstrates our main motivation. 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). More multi-source synthesis results can be observed in Supp. K. Note that purple boxes are not exact, they only outline 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 reference, they are not used in MixSyn. Similarly, Fig. 6 shows random compositions and random images by MixSyn, where each mask region (and image segment) is sourced from a different building.

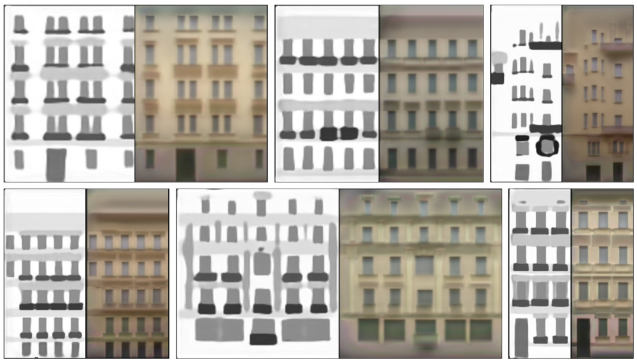


Figure 6. **MixSyn on Facades.** Sample composition and image pairs, where each region is selected from a different building.

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	0.68	0.15	17.14	23.69
SPADE	0.63	0.21	14.30	22.43
SEAN	0.7	**0.12**	18.74	17.66
MixSyn	**0.95**	1.89	**31.32**	**14.41**
MixSyn Structure	0.97	1.15	33.06	NA
MixSyn (H)	0.96	1.46	32.13	NA
MixSyn Structure (H)	0.98	0.92	36.00	NA

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

Finally, we perform a cross-dataset evaluation by testing MixSyn trained on CelebAMask-HQ on Helen (Tab. 1 (H)). High similarity indicates that MixSyn is generalizable, creating multi-source faces from other datasets. Relatively worse RMSE signals that we indeed create novel masks *with inexact reconstructions* where multiple regions adapt/blend. Supp. E declares all cross-dataset scores.

5.2. Comparison

As MixSyn is the first of its kind, we compare it to single-source [11, 26, 33], sequential [16, 50], and collage [5] approaches. We emphasize that they (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/fuzzy/incomplete masks.

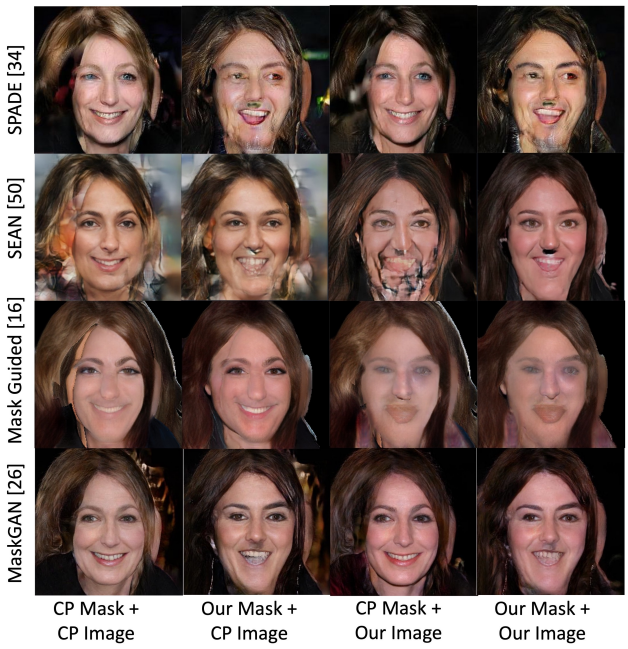
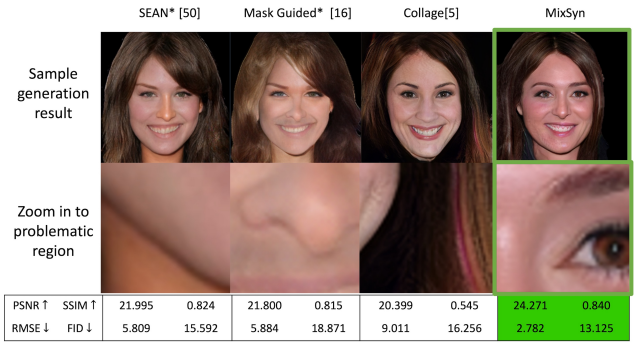


Figure 7. **Comparison**. Output pairs in Fig. 5 bottom, are fed to [16, 26, 33, 50]. None can generate from multi-source, non-existing masks, or realistic results. See Supp. F for detailed scores.

We start with justifying these claims. For (i-ii), we feed four combinations of (copy-paste/our) x (mask/image) pairs in Fig. 5 as alternative inputs to SPADE [33], SEAN [50], Mask Guided CGAN [16], and MaskGAN [26]. Although results improve from copy-paste masks (Fig. 7, col. 1 & 3) to our generated masks (col. 2 & 4), quality of their results are not close to ours (Fig. 5), supporting claims (ii-iii). We also investigate how others reconstruct our result image with our mask (col. 4). Because of claims (iv-v) above, they simply cannot utilize fuzzy masks. We repeat the exercise with blended copy-paste inputs in Supp. G.

In Fig. 8, we select a base mask (top left of Fig. 5) since other approaches cannot blend regions (ii-iii). Then, we swap segments as the sequential component transfer applications of [16, 50] (following claim (i)), shown in the first two columns. Despite looking better than col. 1-4 in Fig. 7, it is akin to a blended copy-paste, creating the zoomed-in artifacts (e.g., different neck and nose colors,

and a shadow mustache), because they are not jointly composing new masks (iv-v). For the third column, we use the copy-paste image as collage input for [5]. Although visual artifacts decrease, there are empty areas and inconsistent hair style. Similarity scores prove that segments are not as well-preserved, questioning [5]'s realism over fidelity.



PSNR ↑	SSIM ↑	21.995	0.824	21.800	0.815	20.399	0.545	24.271	0.840
RMSE ↓	FID ↓	5.809	15.592	5.884	18.871	9.011	16.256	2.782	13.125

Figure 8. **Sequential*/Collage Comparison**. Component transfer distorts nose and neck for [16], and creates a ghost mustache for [50]. Collage synthesis [5] creates color artifacts and empty areas. Quantitative results also support this visual comparison.

Quantitative comparison of MixSyn also supports and generalizes these claims. In Tab. 1, we list SSIM [49], RMSE [19], PSNR [19], and FID [18] scores of Pix2PixHD [41], SPADE [33], SEAN [50], our structure generator, and overall MixSyn architecture on CelebAMask-HQ dataset [26]. Although our reconstruction is not as exact (worse RMSE), MixSyn has more generative capability (better FID). We remark 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. 8, which are also better than the SOTA. Detailed reconstruction scores (Supp. C.) on faces and facades, region similarity scores for *random* images (Supp. D), and scores of Fig. 7 (Supp. F) are documented in the supplemental.

5.3. Experiments

Ablation Study. Fig. 9 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, patterns per region are lost, e.g., curly hair is ironed. Finally, without normalization, style of small regions are dominated (e.g., eyes). The dataset scores are computed similar to Tab 1, but on the results generated with specific loss functions.

Region Types. Starting with 18 base region types of

PSNR↑	SSIM↑	11.68	0.33	13.68	0.38	11.79	0.36	13.49	0.36
RMSE↓	FID↓	8.59	16.64	9.15	17.18	8.59	16.97	9.25	25.38

Figure 9. **Ablation Study.** Samples of source and generated known images with different losses, followed by dataset scores.

faces and their hierarchy [26] (Fig. 10), we couple left-right indices in *random* compositions (blue), to enforce learning correlation of symmetric regions. We experimentally validate that it is better than putting them into the same channel. Supp. H shows results without symmetry coupling on faces and facades, shifting results to the uncanny valley. We also try compact subtypes of face regions (6 yellows), expecting style generator to fill in rare types. Instead, structure network merged them to existing types, creating interesting compositions (Supp. J). To decrease training time, increase accuracy, and fit encoders in the memory, we introduce meta-types by grouping. We intuit that finer granularity regions are needed for better style transfer, but not for synthesis. 15 final meta-types are marked in pink.

Figure 10. **Region Types.** Starting from MaskGAN [26] types, we create meta-types (pink), and couple symmetries for random generation (blue). We also experiment with compact types (yellow).

Mask Dependency. Although MixSyn works best with accurate semantic masks, they can be inexact during inference. We show 2 sets of generated mask/image pairs from 5 sources (not shown) using (a) original masks, (b) mask bounding boxes, and (c) hand-drawn strokes in Fig. 11, except hair. Face features get slightly larger for (b), which is expected but negligible.

Number of Mask Regions. Finally, we experiment with different number of regions, as not all regions exist in all samples such as hat, jewellery, columns, or deco. Not all

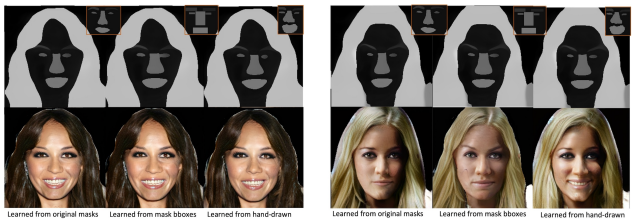


Figure 11. **Mask Variations.** Result mask/image pairs are not significantly effected when using original, bbox, and hand-drawn masks. Insets show sample mask version.

structure codes are needed while generating an image, e.g., there is no r_{cloth} in input regions of Fig. 5, so $e^*_{cloth} = [0]$. The fuzziness between hair/face regions in third and last columns enables both generators to recover in a random composition, placing cloth region in mask and cloth segment in image. Our style and structure GANs behave as unconditional image generators if some input vectors are 0, thus, the results are still photorealistic. Figs. 1, 7, 12, have 4 regions; 5 has 6 regions; Fig. 13 has several regions incrementally; Supp. A and Supp. K has 3, 4, 5 regions; and Supp. J has conflicting regions. Quality is observed to be independent of number of regions.

6. Applications

6.1. Combinatorial Diversity

Each row in Fig. 12 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, simulations with synthetic collections, and data augmentation for DNNs.

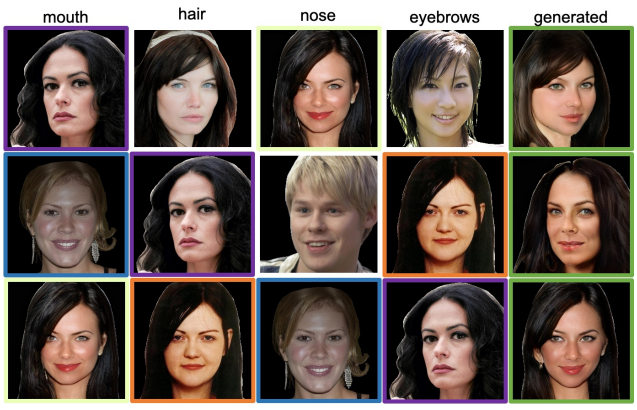


Figure 12. **Design Space**. Using varied regions from same set of images (color-coded), design space grows exponentially (green).

Figure 13. **Perpetual Edits.** After first face (left) is created, each segment is replaced by others from different faces. Bottom left face is very similar to the first, because original face is swapped.

6.2. Edit Propagation

In Fig. 13, we start by generating an image given {mouth, nose, eye_l, eye_r} regions. Then, we change one or more 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 of the initial image is chosen as input, creating a similar face, as region features are preserved during edits.

7. 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.

For improvements, we present hard cases with variations in illumination, semantics, resolution, pose, and cross-dataset transfers in Supp. I. Some combinations cause edge cases naturally, such as a face region from an image with hair and hair region on the sides from a bald person (Supp. J). An interactive editing system can aid in eliminating such.

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.

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), 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), 2019. 2

[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)*, 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, 7

[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)*, 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)*, 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)*, 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, 4, 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)*, 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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 502–508. International Joint Conferences on Artificial Intelligence Organization, 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)*, 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)*, 2019. 1, 2, 3, 5, 6, 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016. 4

[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*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 7

[19] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. 7

[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)*, 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, 2

[22] David K. Han Jeong gi Kwak and Hanseok Ko. Cafe-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)*, 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, 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)*, 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] Jinpeng 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. 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. Helminger, 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. 2

[33] 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)*, 2019. 1, 2, 5, 6, 7

[34] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Computer Vision – ECCV 2020*, pages 319–345, Cham, 2020. Springer International Publishing. 1, 2

[35] Ryohei 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: Multi-input-conditioned hair image generation for portrait editing. *ACM Trans. Graph.*, 39(4), 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)*, 2020. 1, 2

[38] Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. 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, 7

[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)*, 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, 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)*, 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)*, 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 *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV*, 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. 7

[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)*, 2020. 1, 2, 4, 5, 6, 7