Labels4Free: Unsupervised Segmentation using StyleGAN

Rameen Abdal[1]   Peihao Zhu[1]   Niloy J. Mitra[2]   Peter Wonka[1]

[1]KAUST   [2]UCL and Adobe Research

Abstract

We propose an unsupervised segmentation framework for StyleGAN generated objects. We build on two main observations. First, the features generated by StyleGAN hold valuable information that can be utilized towards training segmentation networks. Second, the foreground and background can often be treated to be largely independent and be swapped across images to produce plausible composited images. For our solution, we propose to augment the Style-GAN2 generator architecture with a segmentation branch and to split the generator into a foreground and background network. This enables us to generate soft segmentation masks for the foreground object in an unsupervised fashion. On multiple object classes, we report comparable results against state-of-the-art supervised segmentation networks, while against the best unsupervised segmentation approach we demonstrate a clear improvement, both in qualitative and quantitative metrics.

Project Page : https://rameenabdal.github.io/Labels4Free

1. Introduction

Given the high quality and photo-realistic results of current generative adversarial networks (GANs), we are witnessing their widespread adaptation for many applications. Examples include various image and video editing tasks, image inpainting [13, 47, 46, 41], local image editing [42, 2], low bit-rate video conferencing [40], image super resolution [28, 12], image colorization [4, 27], and extracting 3D models [30].

While originally conjectured that GANs are merely great at memorizing the training data, recent work in GAN-based image editing [18, 3, 36, 42] demonstrates that GANs learn non-trivial semantic information about a class of objects, e.g., faces or cars. For example, GANs are able to learn the concept of pose and they can show the same, or at least a very similar looking, object with different orientations. Even though the background changes in subtle ways during this editing operations, in this paper we explore to what extent the underlying generator network actually learns the distinction between foreground and background, and then

Figure 1: We propose an unsupervised segmentation framework that enables foreground/background separation for raw input images. At the core of our framework is an unsupervised network, which segments class-specific Style-GAN images, and is used to generate segmentation masks for training supervised segmentation networks.

to encourage it to disentangle foreground and background, without explicit mask-level supervision. As an important byproduct, we can extract information from an unsupervised GAN that is useful for general object segmentation. For example, we are able to create a large synthetic dataset for training a state-of-the-art segmentation network and then segment arbitrary face (or horse, car, cat) images into foreground and background *without* requiring any manually assigned labels (see Fig. 1).

Our implementation is based on StyleGAN [24, 23], generally considered the state-of-the-art for GANs trained on individual object classes. Our solution is built on two ideas. First, based on our analysis of GAN-based image editing, the features generated by StyleGAN hold a lot of information useful for segmentation, and can be used towards corresponding mask synthesis. Second, the foreground and background should be largely independent and be composited in different ways. The exact coupling between foreground and background is highly non-trivial however and there are multiple ways of decoupling foreground and background that we analyze in our work.

For our solution, we propose to augment the StyleGAN generator architecture with a segmentation branch and to

split the generator into a foreground and background network. This enables us to generate soft segmentation masks for the foreground object. In order to facilitate easier training, we propose a training strategy that starts from a fully trained network that only has a single generator and utilizes it towards unsupervised segmentation mask generation.

To summarize, our main contributions are:

- A novel architecture modification, a loss function, and a training strategy to split StyleGAN into a foreground and background network.

- Generating synthetic datasets for segmentation. Our framework can be used to create a complete dataset of labeled GAN generated images of high quality in an unsupervised manner. This dataset can then be used to train other state-of-the-art segmentation networks to yield compelling results.

2. Related Work

High Quality GANs. From the seminal works of Goodfellow et al. [17] and Radford et al. [31], subsequent GAN research contributed to big improvements in the visual quality of the generated results. The state-of-the-art networks like ProGAN [20], BigGAN [10], StyleGAN [23], StyleGAN2 [24] and StyleGAN2-ada [21] demonstrate superior performance in terms of diversity and quality of the generated samples. While StyleGAN series by Karras et al. [21] has demonstrated high quality and photo-realistic results on human faces using the high quality FFHQ [23] dataset, BigGAN can produce high quality samples using complex datasets like ImageNet. In our work we build on StyleGAN2 which is the current state of the art for many smaller data-sets, including faces.

GAN Interpretability and Image Editing. GAN interpretability has been an important aspect of the GAN research since the beginning. Some recent works in this domain [5, 6, 16, 18, 49, 3, 36, 2, 37] study the structure of the activation and latent space. For instance GANspace simplifies the latent space of StyleGAN (W Space) to be linear and extracts meaningful directions using PCA. StyleRig [37] mapped StyleGAN latent space to a riggable face model. StyleFlow [3] studied the non-linear nature of the StyleGAN latent space using normalizing flows and is able produce high-quality sequential edits on the generated and real images. On the other hand, the layer activation based methods [16, 2, 5, 42] try to understand the properties of GANs using an analysis of the activation space. Investigation into StyleSpace [42, 43] studies the style parameters of the channels to determine various properties and editing abilities of StyleGAN.

Another interesting approach for interpretability of GANs is image embedding. Abdal et. al. [1] demonstrated high quality embeddings into the extended W space called W^+ space for real image editing. Subsequent works [48, 32, 38] try to improve upon the embedding quality, e.g. by suggesting new regularizers for the optimization. The embedding methods combined with image editing capabilities of StyleGAN has lead to many applications and even commercial software such as Adobe Photoshop's Neural Filters [15]. A known problem in the domain of image editing is the disentanglement of the features during an editing operation. While state-of-the-art editing frameworks [3, 37, 18] achieve high quality fine grained edits using supervised and unsupervised approaches, background/foreground aware embeddings and edits still pose a challenge. Often, the background of the scene changes a lot during the embedding itself or during sequential edits performed on the images. Later in Sec. 4, we show how our unsupervised segmentation framework leads to improved image editing.

Unsupervised Object Segmentation. Among the few attempts in this domain are works by Xu et al. [19] and Yassine et al. [29] which learn a clustering function in an unsupervised setting. A more recent work, Van et al. [39] adopts a predetermined prior in a contrastive optimization objective to learn pixel embeddings. In the GAN domain, some works [7, 11] formulate multiple tasks in unsupervised settings. On higher resolution images, PSeg [9] is currently the only unsupervised approach to segmenting foreground and background and therefore our main competitor.

3. Method

Our goal is to segment StyleGAN generated images (i.e., a generative model that allows sampling from a normal distribution $z \sim \mathcal{N}(0, \mathbb{I})$ to produce class-specific images $\mathcal{G}(z)$) into foreground and background layers. We achieve this *without* access to ground truth masks. Our self-supervised algorithm relies on two observations: first, for many object classes, foreground and background layers can be swapped and recomposited to still produce good quality images; and second, the interior layers of a trained StyleGAN already contain valuable information that can be used to extract such foreground/background segmentation.

Our framework consists of two generators that share a single discriminator. The first generator $\mathcal{G}$ is a pretrained StyleGAN generator whose channels are used by the Alpha-Generator Network $\mathcal{A}$ to eventually extract the foreground image as $\mathcal{A}(z) \odot \mathcal{G}(z)$ (see Sec. 3.1). The second generator $\mathcal{G}_{\text{bg}}$, the background generator is responsible to generate background image samples $\mathcal{G}_{\text{bg}}(z')$ from $z' \sim \mathcal{N}(0, \mathbb{I})$ (see Sec. 3.2). Note that both are derived from pretrained generators. The final image, obtained by mixing background/foreground from different branches, is composited using standard α-blending as $\mathcal{A}(z) \odot \mathcal{G}(z) + (1 - \mathcal{A}(z)) \odot \mathcal{G}_{\text{bg}}(z')$, which a discriminator $\mathcal{D}$ cannot tell apart from the
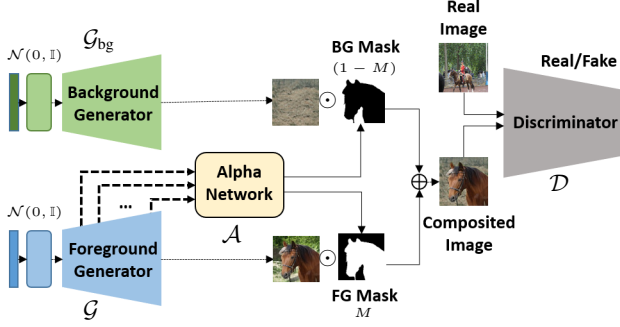
Figure 2: Our unsupervised segmentation network that makes use of pretrained generators $\mathcal{G}$ and $\mathcal{G}_{bg}$ to simultaneously train a segmentation network $\mathcal{A}$ (see Fig. 3) and a 'weak' discriminator $\mathcal{D}$, without requiring supervision for ground truth masks (see Sec 3).

real images. Note that only the network $\mathcal{A}$ is trained in this setup using the regular GAN loss. Fig. 2 shows an overview of this architecture. We expand on the training dynamics of this network in Sec. 3.3.

3.1. Foreground Generator with Alpha Network

Features of the pretrained generator $\mathcal{G}$ already contain sufficient information to segment image pixels into foreground and background. We extend the foreground generator with an Alpha Network ($\mathcal{A}$) to learn the alpha mask for the foreground (see Fig. 3). In this context, we overcome multiple challenges.

First, the feature maps from lower layers of the generator need to be upsampled. To this effect, we introduce upsampling blocks in the alpha network.

Second, the number of features per pixel is quite high (e.g., 9000+ for StyleGAN2). We therefore use 1×1 convolutions for compression and feature selection. This encourages dropping the several features that do not contain segmentation information.

Third, we optionally discard channels and complete layers using a semi-automatic analysis using the tRGB layers of the StyleGAN2. These layers are RGB channels conditioned on the output tensors with the W codes at each resolution contributing to the final image. We initialize the tRGB layers in different resolutions with noise $\mathcal{N}(0, \mathbb{I})$ at each pixel position and analyze the SSIM loss with respect to the original sample. We average the results at each resolution and discard the ones with high SSIM scores. For example, for faces, the tRGB corresponding to the 4th layer of StyleGAN (32×32 resolution) is retained as the average SSIM for the 3rd layer is about 25% higher than those at the 4th layer. Hence we select all the latter tensors including the 4th layer tensors for the construction of $\mathcal{A}$. Similarly, we identify the dominant layers to be 2nd for LSUN-Horse and LSUN-Cat and 3rd for the LSUN-Car (see Appendix B).
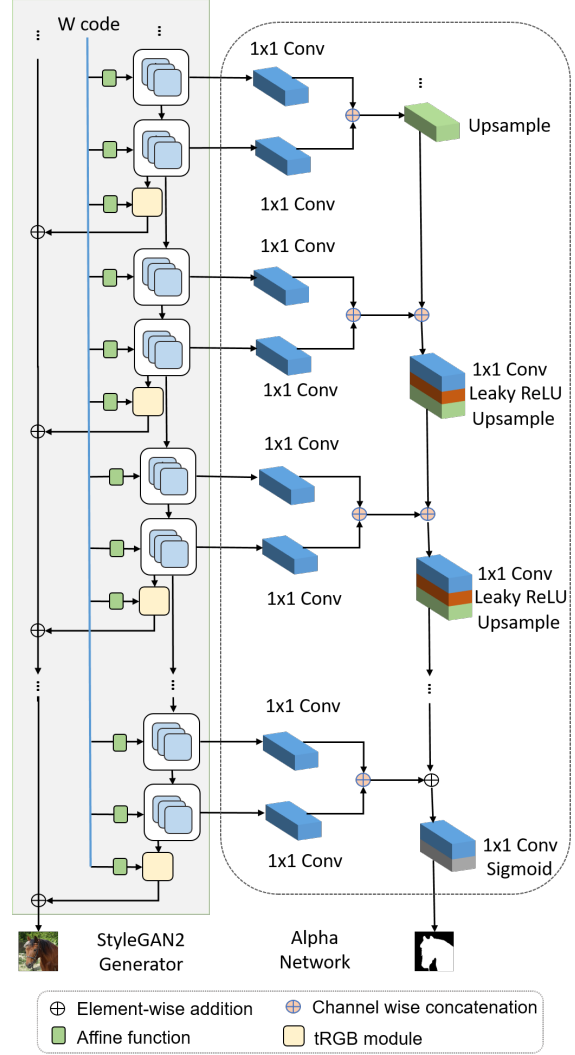

Figure 3: Architecture of the Alpha Network $\mathcal{A}$ that directly operates on pretrained features of StyleGAN to directly produce a binary segmentation mask M.

Fourth, the features per pixel have to be processed to output an alpha value. We therefore use several non-linear activation functions in the upsampling blocks and add a sigmoid function at the end to make the output in the range [0,1]. This yields a lightweight but simplified network to extract segmentation information. In particular, neighboring pixels do not interact with each other.

3.2. Background Generator

The challenge for the background generator is the training dynamics. Specifically, when initializing with a pretrained generator, the method fails because the background already includes the foreground object. When pre-training the background generator on a different dataset, our at-

tempts failed because the discriminator can easily detect that the backgrounds are out of distribution (see Appendix A). We therefore adopted an approach based on the conjecture that the foreground and background image are already composited by StyleGAN.

We start out by seeking channels which are responsible for generating the background pixels in a StyleGAN image. Let $\mathcal{G}$ be the StyleGAN generator and, w and n be the latent and noise variables, respectively. In order to bootstrap the network to identify the channels, we first identify and collect generic StyleGAN backgrounds from multiple sampled images by cropping. We notice that background having, for example, a white, black and blue tinge are common in-domain backgrounds. Then we find the gradient of the objective function: $\|(\mathcal{G}(w, n) - x)\|_2^2$ with respect to the tensors in the StyleGAN2 layers at all resolutions, where x is the upsampled background (crops). The above process allows identifying the layers which are most responsible for deleting an object (e.g., faces, cats, horses) from a composited StyleGAN2 image.

In order to quantify the calculated gradient maps, we the calculate the sum of gradients norm over the channels of the respective layers. We found that the first layer (the constant layer and the first layer in which the W latent is injected excluding the tRGB layer [24]) has the maximum value of the above measure. Hence, we hypothesise that switching off (i.e., zeroing out) the identified channels would exclude the object information from the tensor representation and produce an approximate background. Fig. 4 and 5 show the sampled backgrounds following the steps described above. Note that we could produce higher quality curated backgrounds by empirically setting some channels to be active based on a threshold of the above measure. However, we noticed that such backgrounds occasionally contained traces of foreground objects. Hence, in the training phase, we adopt a safe strategy and zero out all the channels of the selected layer. We refer to this trimmed background generator as $\mathcal{G}_{bg}$.

3.3. Training Dynamics

Our unsupervised setup consists of the Alpha Network $\mathcal{A}$, a pre-trained StyleGAN generator $\mathcal{G}$ for the foreground, a modified pre-trained StyleGAN generator $\mathcal{G}_{bg}$ for the background, and a weak discriminator $\mathcal{D}$. We freeze the generators $\mathcal{G}_{bg}$ and $\mathcal{G}$ during training and only train the Alpha Network $\mathcal{A}$ and the discriminator $\mathcal{D}$ using adversarial training. Note that we do not apply the path regularization to the discriminator $\mathcal{D}$ and style mixing during the training. Unlike others [9], our framework does not alter the generation quality of samples. The final composited image given to the Discriminator $\mathcal{D}$ is:

$$I(w, w', n, n') := M \odot \mathcal{G}(w, n) + (1 - M) \odot \mathcal{G}_{bg}(w', n'),$$
(1)

where M is the mask predicted by the Alpha Network (i.e., $M = \mathcal{A}(z)$). We use W space for the training, and show in Sec. 4 that the method can generalize to W^+ space to handle real images projected into StyleGAN2. In order to robustly train the network and avoid degenerate solutions (e.g., $\mathcal{A}$ to produce all 1s), we make several changes as described next.

(a) Discriminator $\mathcal{D}$: Unlike using a frozen pre-trained generator $(\mathcal{G}, \mathcal{G}_{bg})$, we train the discriminator from scratch. Our method fails to converge by starting the training from a pre-trained discriminator. We hypothesize that that the pre-trained discriminator is already very strong in detecting the correlations between the foreground and the background [35]. For example, environmental illumination, shadows and reflections in case of cars, cats, and horses. Hence to enable our network to train on independently sampled backgrounds, we use a 'Weak' discriminator, in the sense that it does not have a strong prior about the correlation between foregrounds and backgrounds.

(b) Truncation trick: We observed that unlike FFHQ trained StyleGAN2, the samples from the LSUN-Object trained StyleGAN2 have variable quality produced without truncation. To avoid the rejection sampling and ensure high quality samples during the training, we use truncation $\psi \in [0,1]$ for the generator $\mathcal{G}$ in case of LSUN-Object training [22].

(c) Regularizer: As the alpha segmentation network may choose to converge into a sparse map, we use a binary enforcing regularizer, $B(M) := \min(M, 1 - M)$ to guide the training. Note that we control this regularizer to still get soft segmentation maps. The truncation trick above may result in backgrounds from $\mathcal{G}_{bg}$ more aligned to the original distribution than the composite image from $\mathcal{G}$. Hence, for the LSUN-Object, we use another regularizer $C(M) := \text{ReLU}(\phi_1 - \frac{1}{m \times m} \sum_i M_i)$ which ensures that the optimization does not converge to a trivial solution of only using the background network. Similarly, the regularizer as $E(M) := \text{ReLU}(\phi_2 - \frac{1}{m \times m} \sum_i (1 - M_i))$ checks that the solution does not degenerate to only using the $\mathcal{G}$ network.

4. Results

4.1. Evaluation Measures

For the task of segmentation, IOU (Intersection Over Union) and mIOU (mean Intersection Over Union) give the measure of overlap between the ground truth and predicted segments. Since there are two classes we calculate IOU for both the classes in the experiments. We also report the final mIOU. Additional metrics that we report to evaluate the segmentation are Precision (Prec), Recall (Rec), F1 Score, and accuracy (Acc). As surrogate for visual quality, we use FID in our experiments. This can only be seen as a rough approximation to visual quality and is mainly useful in conjunction with visual inspection of the results.

Table 1: Evaluation of unsupervised $PSeg$ [8, 9] against our unsupervised approach using results from supervised network BiSeNet [44, 50], trained on CelebA-Mask dataset [25], as ground truth. Note that for $PSeg$, since the samples can be of low-quality, we use the Detectron2 model for person detection before evaluating the masks. See Fig. 4 for assessing the visual quality of the BiSeNet generated masks.

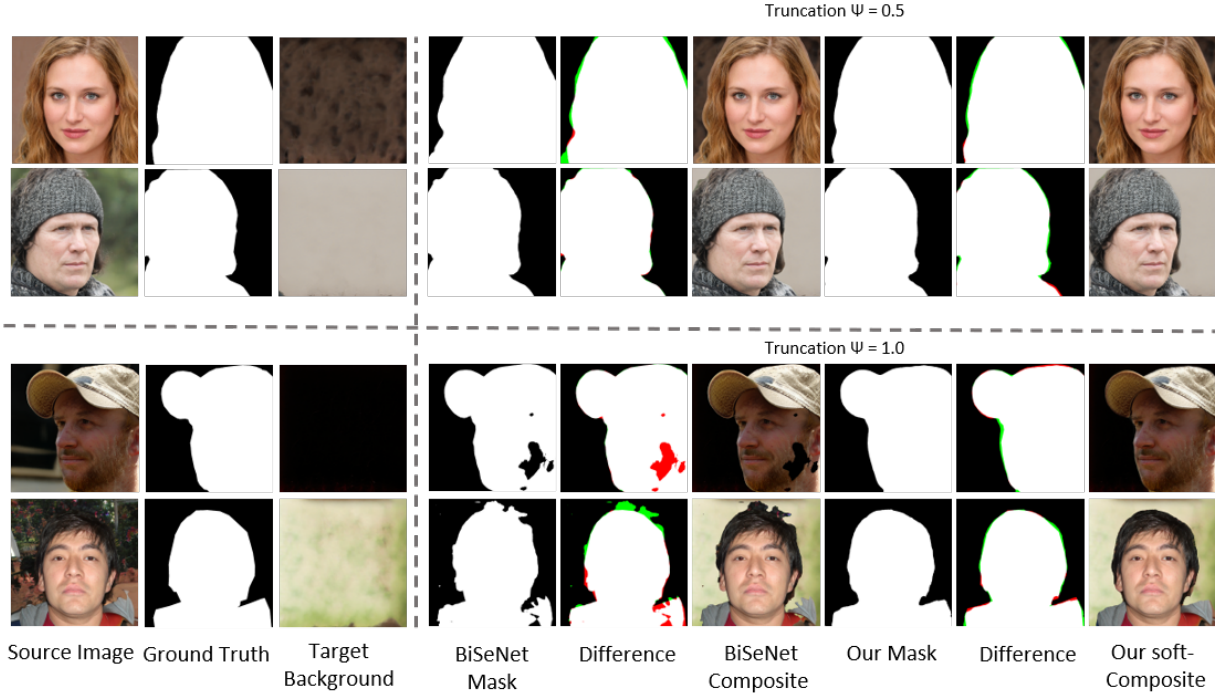| Method	Truncation $\Psi = 0.7$						Truncation $\Psi = 1.0$					
	IOU fg/bg	mIOU	F1	Prec	Rec	Acc	IOU fg/bg	mIOU	F1	Prec	Rec	Acc
$PSeg$	0.52/0.82	0.67	0.80	0.78	0.81	0.85	0.50/0.81	0.66	0.78	0.77	0.80	0.84
$Ours$	**0.87/0.94**	**0.90**	**0.95**	**0.95**	**0.94**	**0.95**	**0.75/0.89**	**0.82**	**0.90**	**0.92**	**0.89**	**0.92**

Figure 4: Qualitative results of our unsupervised framework on StyleGAN2 trained on FFHQ compared with BiSeNet trained on CelebA-HQ Masks. Note that the green and red areas are the 'False Positives' and 'False Negatives' with respect to the foreground in ground truth. We report our results on two truncation levels.

4.2. Datasets

We use StyleGAN2 pretrained on FFHQ [22], LSUN-Cars, LSUN-Cats, and LSUN-Horse [45] datasets. FFHQ is a face image dataset with resolution 1024×1024. The facial features are aligned canonically. The LSUN-Object dataset contains various objects. It is relatively diverse in terms of poses, position, and number of objects in a scene. We use the subcategories for cars, cats, and horses.

4.3. Competing Methods

We compare our results with two approaches, a supervised approach and an unsupervised approach.

In the *supervised setting*, we use BiSeNet [44, 50]

trained on the CelebA-Mask dataset [25] for the evaluation of the faces and Facebook's Detectron 2 Mask R-CNN Model (R101-FPN [14]) with the ResNet101 architecture pre-trained on the MS-COCO dataset [26] for the evaluation on LSUN-Object datasets. As our method is unsupervised, these methods are mainly suitable to judge how close we can come to supervised training. They are not direct competitors to our method. We also create a custom evaluation dataset of 10 images per class to directly compare the two approaches (see Appendix C).

In the *unsupervised setting*, we compare our method with PSeg [8, 9] using the parameters in the open source GitHub repo. This method is the only published competitor to ours.
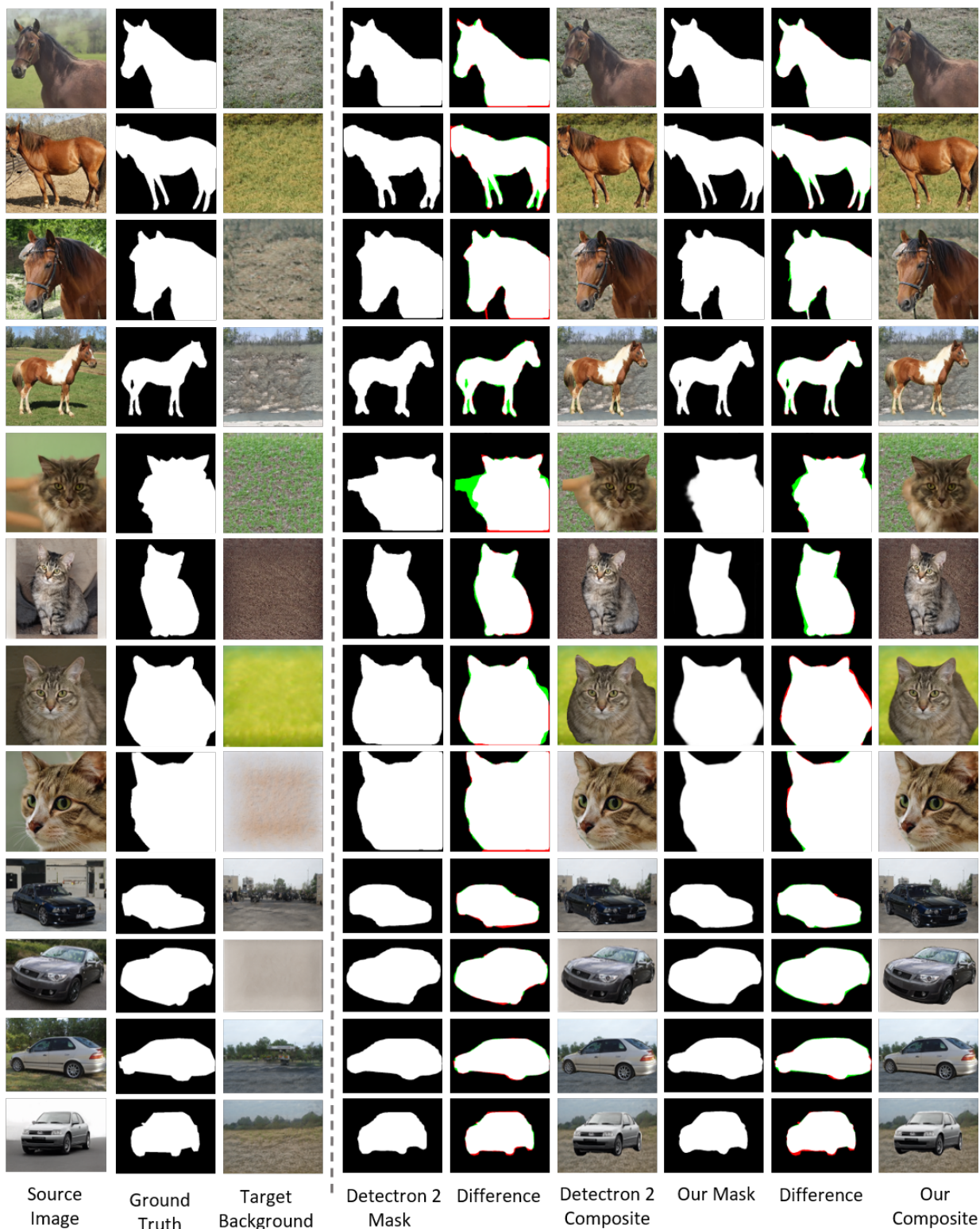
Figure 5: Qualitative results of our unsupervised framework on StyleGAN2 trained on LSUN-Horse, LSUN-Cat and LSUN-Car (LSUN-Object) compared with Detectron 2 trained on MS-COCO. Note that the green and red areas are the 'False Positives' and 'False Negatives' with respect to the foreground in ground truth.

Figure 7: Samples from $PSeg$ method trained on FFHQ dataset. For samples on LSUN-Object refer to Fig. 3 and Fig. 4 in $PSeg$ [9] paper. Note that the method struggles with the quality of samples.
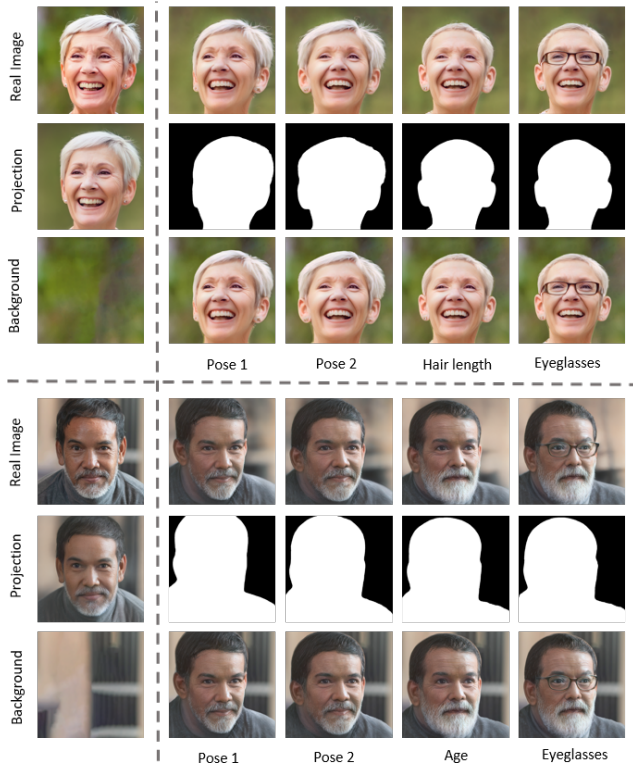
Figure 6: Our method achieves better background preservation compared to original semantic edits in StyleFlow [3]. For the real image, we first obtain a background layer, segmented using Label4Free and then completed using ContentAwareFill, and then for each edit using StyleFlow, we again segment them using our method and then composite back with the completed background layer (obtained above). Please compare the first row versus third rows.

4.4. Comparison on Faces

We compare and evaluate segmentation results on both sampled and real images quantitatively and qualitatively. First we show qualitative results of the learned segmentation masks using the images and backgrounds sampled from the StyleGAN2 in Fig. 4. To put these results into context, we compare the segmentation masks of the learned alpha network $\mathcal{A}$ with the *supervised* BiSeNet. The figure shows results at different truncation levels. The truncation setting $\psi = 0.5$ leads to higher quality images. The setting $\psi = 1.0$ leads to lower quality images that are more difficult to segment from the composited representation. In such cases our method is able to outperform the supervised segmentation network. In Table 1, we compare the results of the unsupervised segmentation of StyleGAN2 generated images using our method with unsupervised Pseg. We train PSeg [8, 9] using the parameters in the Github repo on the FFHQ dataset. In the absence of a large corpus of testing

data, we estimate the ground truth using a supervised segmentation network (BiSeNet). We sample $1k$ images and backgrounds at different truncation levels and compute the evaluation metrics. The quantitative results clearly show that our method is able to produce results of much higher quality than our only direct (unsupervised) competitor. The PSeg method tends to extract some random attributes from the learned images and is very sensitive to hyperparameters (see Fig. 7). In contrast, our method ensures that the original generative quality of StyleGAN is maintained. We also compare the FID of the sampled images in Table 5. The results show that PSeg drastically affects the sample quality.

Background preserving face editing In order to show the application for our method in real image editing, we use state-of-the-art image editing framework StyleFlow [3]. One of the limitations of sequential editing using StyleGAN2 is that it has a tendency to change the background. Fig. 6 shows some edits on the real and sample images using StyleFlow and corresponding results where our method is able to preserve the background using the same edit. Notice that our method is robust to pose changes.

Performance on real face images. One method to obtain segmentation results on real images is to project them into the StyleGAN2 latent space. We do this for $1k$ images from the CelebA-Mask dataset using the PSP encoder [32]. We show quantitative results in Table 4 comparing the output segmentation masks with the ground truth in the dataset. Note that the Alpha Network $\mathcal{A}$ is not trained on the W^+ space but is able to generalize well on the real images. A downside of this approach is that the projection method introduces its own projection errors affecting segmentation.

Labels4Free to obtain synthetic training data. A better method to extend to arbitrary images is to generate synthetic training data (see Table 4). We train a UNet [33] and BiSeNet on $10k$ sampled images and backgrounds produced by our unsupervised framework and report the scores. Note that the scores are comparable to the supervised setup. This demonstrates that our method can be applied to synthetically generate high quality training data for class specific foreground/background segmentation.

Table 2: Evaluation of unsupervised $PSeg$ [8, 9] against our unsupervised approach on LSUN-Object categories (LSUN-Horse, LSUN-Cat and LSUN-Car) using results from supervised network Detectron 2 R101-FPN [14] trained on MS-COCO dataset. We report the results without truncation and the threshold for the masks is 0.9. A: LSUN-Cat ; B : LSUN-Horse ; C: LSUN-Car.

Method	IOU fg/bg	mIOU	F1	Prec	Rec	Acc
$PSeg$(A)	0.65/0.68	0.66	0.80	0.81	0.80	0.79
Ours(A)	**0.84/0.77**	**0.81**	**0.89**	**0.89**	**0.90**	**0.90**
$PSeg$(B)	0.50/0.40	0.45	0.71	0.69	0.73	0.63
Ours(B)	**0.83/0.67**	**0.75**	**0.85**	**0.84**	**0.91**	**0.87**
$PSeg$(C)	0.81/0.73	0.77	0.83	0.83	0.84	0.85
Ours(C)	**0.93/0.84**	**0.89**	**0.94**	**0.93**	**0.95**	**0.95**

Table 3: Evaluation on custom dataset. Dt2 : Detectron 2 A: FFHQ ; B: LSUN-Cat ; C: LSUN-Horse ; D: LSUN-Car.

Method	IOU fg/bg	mIOU	F1	Prec	Rec	Acc
BiSeNet(A)	0.94/0.97	0.95	**0.98**	0.98	0.97	**0.98**
Ours(A)	**0.95/0.98**	**0.96**	**0.98**	**0.99**	**0.98**	**0.98**
Dt2(B)	0.93/0.95	0.94	0.97	0.97	0.97	0.97
Ours(B)	**0.95/0.96**	**0.96**	**0.98**	**0.98**	**0.98**	**0.98**
Dt2(C)	0.96/0.94	0.95	0.97	0.97	0.97	0.98
Ours(C)	**0.97/0.96**	**0.97**	**0.98**	**0.98**	**0.98**	**0.98**
Dt2(D)	**0.99/0.96**	**0.97**	**0.99**	**0.99**	0.98	**0.99**
Ours(D)	0.98/0.95	**0.97**	0.98	0.98	**0.99**	**0.99**

Table 4: Evaluation of the unsupervised object segmentation of real images using a projection method (PSP) and segmentation networks (Unet and BiSeNet) trained on our generated synthetic data.

Method	IOU fg/bg	mIOU	F1	Prec	Rec	Acc
BiSeNet	**0.84/0.92**	**0.88**	**0.93**	0.93	**0.94**	**0.94**
Ours(PSP)	0.83/**0.92**	**0.88**	**0.93**	**0.94**	0.93	**0.94**
Ours(UNet)	0.75/0.88	0.82	0.90	0.90	0.90	0.91
Ours(BiSeNet)	0.81/0.91	0.86	0.92	0.92	0.92	0.93

Table 5: FID comparison of the generation quality of samples from $PSeg$ method vs StyleGAN2. A : FFHQ ; B : LSUN-Cat ; C : LSUN-Horse ; D : LSUN-Car.

Method	A	B	C	D
$PSeg$	24.33	57.32	53.60	30.87
StyleGAN2	2.82	6.93	3.43	2.32

4.5. Comparison on other datasets

We also train our framework on LSUN-Cat, LSUN-Horse and LSUN-Car [45]. These are more challenging datasets than FFHQ. We have identified two related problems with StyleGAN2 trained on these datasets. Firstly, the quality of the samples at lower truncation or no truncation

levels is not as high as the FFHQ trained StyleGAN2. Second, these datasets can have multiple instances of the objects in a scene. StyleGAN2 does not handle such samples well. Both these factors affect the training as well as the evaluation of the unsupervised framework. In Fig. 5 we show the qualitative results of our unsupervised segmentation approach on LSUN-Object datasets. Notice that the quality of segmentation masks produced by our framework is comparable to the supervised network. In Table 2, we calculated the quantitative results of $1k$ sampled images from StyleGAN2. Note that here we resort to rejection sampling based on the technique to reject the sample not identified by the detectron 2 model. For simplicity we also reject the multiple instance object samples. The results show that the metrics scores are comparable to a state-of-the-art supervised method. In order to directly compare the supervised and unsupervised approaches we compare the metrics on a custom dataset mentioned in Section 4.3. Table 3 shows that our supervised approach is either better or has similar segmentation capabilities as the supervised approach.

4.6. Training Details

Our method is faster to train than the competing methods. We train our framework on 4 RTX 2080 (24 GB) cards with a batch size of 8 using the StyleGAN pytorch implementation [34]. Let λ_1 and λ_2 be the weights of the regualarizers $B(M)$ and $C(M)$, respectively. For FFHQ dataset, we run $1k$ iterations and set $\lambda_1 = 1.2$. For LSUN-Object, we set $\phi = 0.25$. For LSUN-Cat, we run 900 iterations and set $\lambda_1 = \lambda_2 = 3$ and $\psi = 0.5$. For LSUN-Horse, we run 500 iterations and set $\lambda_1 = \lambda_2 = 3$ and $\psi = 1.0$. For LSUN-Car, we run 250 iterations and set $\lambda_1 = \lambda_2 = 20$ and $\psi = 0.3$. Also, we set the weight of the non-saturating discriminator loss to 0.1. For FFHQ, LSUN-Horse and LSUN-Cat, we set the learning rate to 0.0002. For LSUN-Car, we set the learning rate to 0.002. Each model takes less than 30 minutes to converge.

5. Conclusion

We proposed a framework for unsupervised segmentation of StyleGAN generated images into a foreground and a background layer. The most important property of this segmentation is that it works entirely without supervision. To that effect, we leverage information that is already present in the layers to extract segmentation information. In the future, we would like to explore the unsupervised extraction of other information, e.g., illumination information, the segmentation into additional classes, and depth information.

6. Acknowledgements

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 1, 2

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv e-prints*, pages arXiv–2008, 2020. 1, 2, 7

[4] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. *arXiv preprint arXiv:2011.11731*, 2020. 1

[5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks, 2018. 2

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019. 2

[7] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *European Conference on Computer Vision*, pages 514–530. Springer, 2020. 2

[8] Adam Bielski. perturbed-seg. https://github.com/adambielski/perturbed-seg. 5, 7, 8

[9] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *arXiv preprint arXiv:1905.12663*, 2019. 2, 4, 5, 7, 8

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 2

[11] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[12] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G Dimakis. Intermediate layer optimization for inverse problems using deep generative models. *arXiv preprint arXiv:2102.07364*, 2021. 1

[13] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018. 1

[14] FaceBook. detectron2. https://github.com/facebookresearch/detectron2. 5, 8

[15] Neural Filters. Adobe photoshop. https://helpx.adobe.com/photoshop/using/neural-filters.html#NeuralFilters. 2

[16] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. Tilegan: synthesis of large-scale non-homogeneous textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019. 2

[17] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 2

[18] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 1, 2

[19] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019. 2

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 2

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018. 4, 5

[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*, pages 4401–4410, 2019. 1, 2

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 1, 2, 4

[25] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 5

[27] Xuan Luo, Xuaner Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, and Steven M Seitz. Time-travel rephotography. *arXiv preprint arXiv:2012.12261*, 2020. 1

[28] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2445, 2020. 1

[29] Yassine Ouali, Céline Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020. 2

[30] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d

shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020. 1

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[32] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 2, 7

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 7

[34] rosinality. stylegan2-pythorch. https://github.com/rosinality/stylegan2-pytorch. 8

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016. 4

[36] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[37] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. 2

[38] Ayush Tewari, Mohamed Elgharib, Mallikarjun BR, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. volume 39, December 2020. 2

[39] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. *arXiv preprint arXiv:2102.06191*, 2021. 2

[40] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. *arXiv preprint arXiv:2011.15126*, 2020. 1

[41] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. Detecting overfitting of deep generative networks via latent recovery. 2019. 1

[42] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. *arXiv preprint arXiv:2011.12799*, 2020. 1, 2

[43] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images, 2021. 2

[44] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 5

[45] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5, 8

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution. 2018. 1

[47] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 1

[48] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020. 2

[49] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2020. 2

[50] zllrunning. face-parsing.pytorch. https://github.com/zllrunning/face-parsing.PyTorch. 5