Spatially-Adaptive Multilayer Selection for GAN Inversion and Editing

Gaurav Parmar [1,2] Yijun Li [2] Jingwan Lu [2] Richard Zhang[2]
Jun-Yan Zhu[1] Krishna Kumar Singh [2]
[1]Carnegie Mellon University [2]Adobe Research

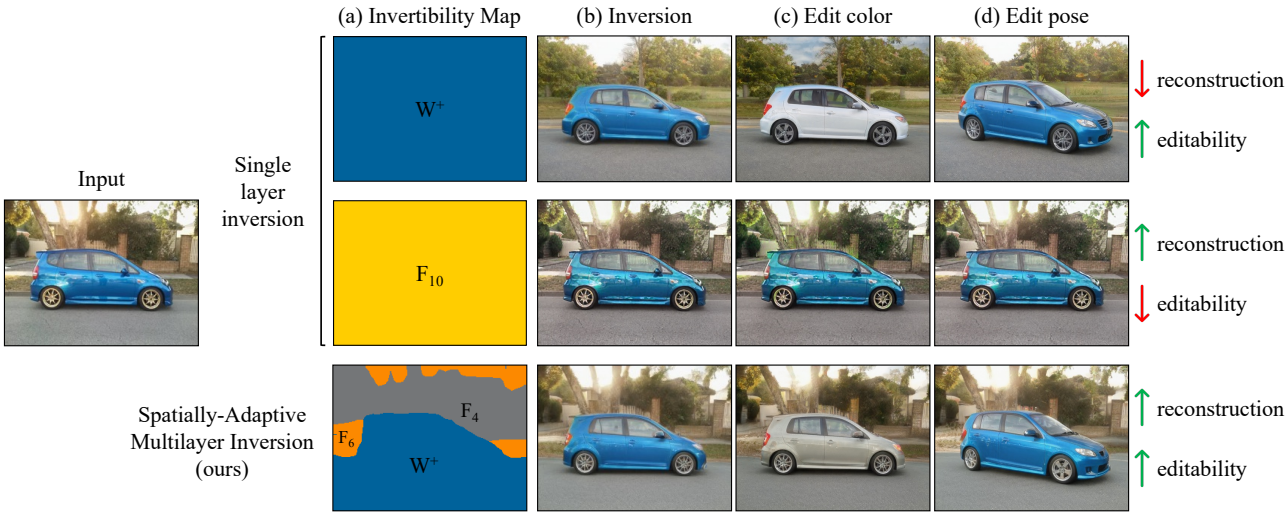

Figure 1. **Inverting and editing an image with spatially adaptive multilayer latent codes.** Choosing a single latent layer for GAN inversion leads to a dilemma between obtaining a faithful reconstruction of the input image and being able to perform downstream edits (1st and 2nd row). In contrast, our proposed method automatically selects the latent space tailored for each region to balance the reconstruction quality and editability (3rd row). Given an input image, our model predicts an invertibility map (a), which contains the layer index used for each region. This allows us to precisely reconstruct the input image (b) while preserving editability (c,d).

Abstract

Existing GAN inversion and editing methods work well for aligned objects with a clean background, such as portraits and animal faces, but often struggle for more difficult categories with complex scene layouts and object occlusions, such as cars, animals, and outdoor images. We propose a new method to invert and edit such complex images in the latent space of GANs, such as StyleGAN2. Our key idea is to explore inversion with a collection of layers, spatially adapting the inversion process to the difficulty of the image. We learn to predict the "invertibility" of different image segments and project each segment into a latent layer. Easier regions can be inverted into an earlier layer in the generator's latent space, while more challenging regions can be inverted into a later feature space. Experiments show that our method obtains better inversion results compared to the recent approaches on complex categories, while maintaining downstream editability. Please refer to our project page at gauravparmar.com/sam_inversion.

1. Introduction

The recent advances of Generative Adversarial Networks [19], such as ProGAN [29], the StyleGAN model family [31–33], and BigGAN [12], have revived the interest in GAN inversion and editing [13, 63]. In GAN editing pipelines, one first projects an image into the latent space of a pre-trained GAN, by minimizing the distance between the generated image and an input image. We can then change the latent code according to a user edit, and synthesize the output accordingly. The latent code can then be changed, in order to satisfy a user edit. The final output image is synthesized with the updated latent code. Several recent methods have achieved impressive editing results for real images [2, 8, 40, 62] using scribbles, text, attribute, and object class conditioning. However, existing methods work well for human portraits and animal faces but are less applicable to more complex classes such as cars, horses, and cats. Compared to faces, these objects have more diverse visual appearance and cluttered backgrounds. In addition, they tend to be less aligned and more often occluded, all of which make inversion more challenging.

In this work, we aim to invert complex images better. We

build our method upon two key observations.

(1) *Spatially-adaptive invertibility*: first, the inversion difficulty varies across different regions within an image. Even if the entire image cannot be inverted in the early latent spaces (e.g., W and W^+ space of StyleGAN2 [33]), if we break the image into multiple segments, the easier regions can still be inverted in these latent spaces with high fidelity. For example, in Figure 1, while the car and sky regions are well-modeled by the LSUN CAR generator, shrubs and fences are not, as they appear less frequently in the dataset. Besides, both regions are occluded by the foreground car.

(2) *The trade-off between invertibility and editability*: as noted by prior work [51, 65], the choice of layer can determine how precisely an image can be reconstructed and the range of downstream edits that can be performed. Early latent layers of a generative model (W, W^+) are often unable to reconstruct challenging images, but allow meaningful global and local editing. In contrast, inversion using later intermediate layers reconstructs the image more precisely at the cost of reduced editing capability. As invertibility increases in later layers, the editability decreases. The first two rows in Figure 1 show these trade-offs concretely for a real car image.

Considering the spatially-varying difficulty and the trade-off between editability and invertibility, we perform spatially-adaptive multilayer (SAM) inversion by choosing different features or latent spaces to use when inverting each image region. We train a prediction network to infer an invertibility map for an input image indicating the latent spaces to be used per segment as shown in the second column of Figure 1. Our approach enables generating images very close to the target input images while maintaining the downstream editing ability.

We conduct experiments on multiple domains such as FACES, CARS, HORSES, and CATS. The results show that our method can maintain editability while reconstructing even challenging images more precisely. We measure reconstruction with standard metrics such as PSNR and LPIPS. Whereas, the image quality and the editability are evaluated using a human preference study. Finally, we demonstrate the generality of our idea on different generator architectures (StyleGAN2 [33], BigGAN-deep [12]), and different paradigms (optimization-based or encoder-based).

2. Related Work

GAN inversion and editing. Since the introduction of GANs [19], several methods have proposed projecting an input image into the latent space of GANs for various editing and synthesis applications [14, 35, 42, 63]. This idea of using GANs as a strong image prior was later used in image inpainting, deblurring, compositing, denoising, colorization, semantic image editing, and data augmentation [7, 8, 15, 16, 20, 54, 59]. See a recent survey [57] for more details. The enormous progress of large-scale GANs [12, 28–33, 61] allows us to adopt GAN inversion for high-resolution images [1, 2]. One popular application is portrait editing [3, 4, 37, 50].

Current methods can be categorized into three groups: optimization-based, encoder-based, and hybrid methods. The optimization-based methods [1, 2, 33, 36, 63] aim to minimize the difference between the optimization output and the input image. Despite achieving fairly accurate results, the slow process requires many iterations and may get stuck in local optimum. To accelerate the process, several works [14, 35, 42, 43, 51, 53, 63] learn an encoder to predict the latent code via a single feed-forward pass. However, the learned encoder is sometimes limited in reconstruction quality compared to the optimization-based scheme. Naturally, hybrid approaches that combine the best of both schemes emerge [5, 8, 10, 24, 53, 63], but the trade-off between quality and speed still persists.

Choosing the latent space. Several previous methods [1, 2] focus on inverting the input image into the latent space of StyleGAN models [32, 33] that use AdaIN layers [23] to control the "style" of an image. In addition to exploring different projection schemes, they demonstrate that the choice of latent space is a key factor due to the unique style-based design of the StyleGAN. Instead of projecting an image into the latent space [14, 63], recent works propose projecting an image into style parameter space [1, 2, 55] and convolutional feature space [64]. As noted by recent work [51, 65], there exists a trade-off between the invertibility and editability, and no layer can maximize both criteria at the same time.

To handle complex images, recent papers propose using multiple codes of the same layer [20, 26, 49], splitting image into segments [18], using consecutive images [58], explicitly handling misaligned objects [6, 24, 27], modifying the generator architecture for better editing ability [34, 39], adopting a class-conditional GAN [24, 38, 49], and fine-tuning the generator to an input image [8, 38, 44].

Different from the above methods that operate on a single layer, we take into account the inversion difficulty across different input image segments and perform the inversion separately for each segment by using multiple latent spaces. We show that our method outperforms a concurrent generator fine-tuning method [44] in our experiments.

Finding editing directions. After inversion, we can edit the inverted code by traversing semantically meaningful directions computed using supervised [9, 25, 47] or unsupervised approaches [17, 21, 41, 48, 52]. Most of these methods compute these directions offline [9, 25, 48] and provide them as pre-canned options for users. Other works calculate the editing directions during inference time to support more flexible editing interfaces with scribbles [63] and text

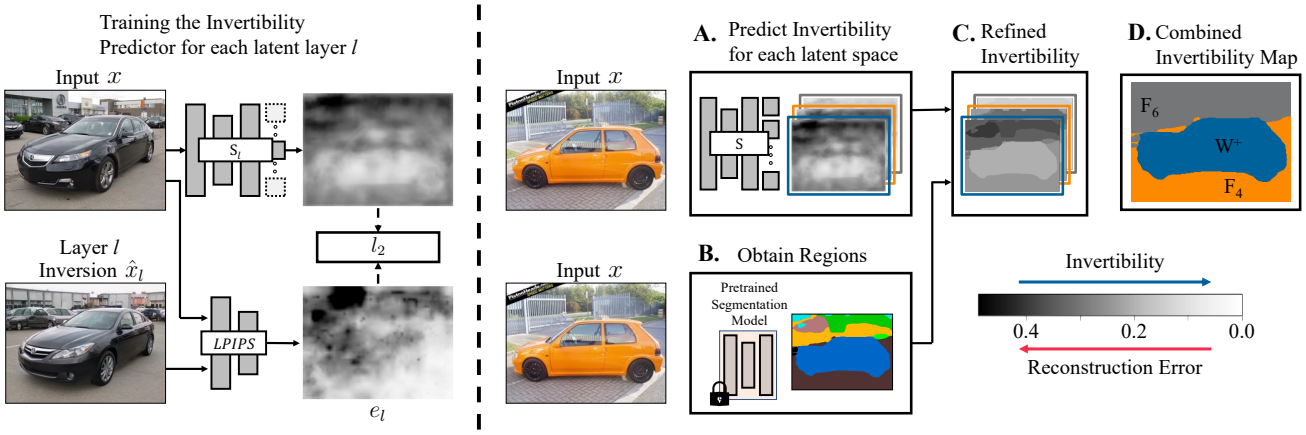

Figure 2. **Training the Invertibility Segmenter.** On the left we show how each of the invertibility predictior S_l are trained. We invert all images in the training set using one of the five candidate latent spaces and use the LPIPS [60] spatial error map e_l as supervision. Next (right) we show how the trained invertibility models are used to generate the final inversion latent map. We first predict how difficult each region of the image is to invert for every latent layer using our aforementioned invertibility network. Subsequently we refine the predicted map using a pre-trained semantic segmentation network and combine them using the user-specified threshold τ. This combined invertibility map shown on the right is used to determine the latent layer to be used for inverting each segment in the image.

inputs [40]. We show that our method can work well with different types of directions.

3. Approach

We aim to invert images using a pretrained GAN while maintaining editability. We begin by learning to predict an invertibility map that indicates which latent spaces should be used for each image region. Next, we fuse features from different latent spaces to generate an image that matches our input and can be edited in the latent space. Additional network training and architecture details are provided in the arXiv version.

3.1. Predicting Invertibility

As discussed previously, different latent spaces have different inversion capabilities. We learn a network to predict what parts of the image are invertible using any given latent space. Here we use "invertibility" to indicate how closely our generated result can match the input image. In Figure 2 (left), we show how we learn invertibility predictor for different latent spaces. We collect a dataset of image pairs that consists of the input image $x \in R^{H\times W\times 3}$ and its reconstruction $\hat{x}_l \in R^{H\times W\times 3}$ into the l^{th} latent space, following the optimization-based inversion suggested by Karras et al. [33]. We consider 5 different latent spaces $\Phi = \{W^+, F_4, F_6, F_8, F_{10}\}$, where the index of F corresponds to the feature layer index of the StyleGAN2 generator and W^+ is the concatenation of different vectors from W space, in which W space is the output space of the MLP network of StyleGAN2. We choose W^+ instead of W, as it provides better inversion results and more fine-grained and disentangled control when performing the downstream edits. Next, we compute the reconstruction loss as follows

$$e_l = \mathcal{L}_{\text{LPIPS}}(x, \hat{x}_l), \tag{1}$$

where $e_l \in R^{H\times W}$ is the LPIPS spatial error map [60] between x and their inversions $\hat{x}_l$ for each latent space.

The parts that are easy to invert have smaller spatial errors, whereas difficult regions induce larger errors. We subsequently train a network to predict the invertibility for each latent space, regressing to the LPIPS spatial error map via an ℓ_2 loss. The training loss can be formulated as follows:

$$S_l = \arg\min_{S_l} \ell_2\big(S_l(x), e_l\big). \tag{2}$$

Once trained, this network predicts the invertibility for any input image, at any layer, in a feed-forward fashion. However, our prediction can be noisy and may not be consistent within the same semantic region. This could potentially result in inconsistent inversions and edits, as different parts of the same region can be assigned to different latent codes. We refine our prediction using a pretrained segmentation model. For every segment, we compute the average predicted invertibility in the region and use the value for the entire segment. As shown in Figure 2 (right), such a refining step helps us align the invertibility map with natural object boundaries in the image.

3.2. Adaptive Latent Space Selection

We observe that latent spaces have an inherent trade-off between reconstructing the input image and utility for downstream image editing tasks, as also noted by recent work [51, 65]. For example, choosing the latent space to be W^+ would result in an inverted latent vector that is amenable for editing, but sub-optimal for obtaining a faithful reconstruction for difficult input images. On the other hand, choosing activation block F_{10} (close to the generated pixel space) would have great reconstruction, but limited editing ability. In Figure 3, we show this trade-off for different choices of latent spaces explicitly. We invert the input

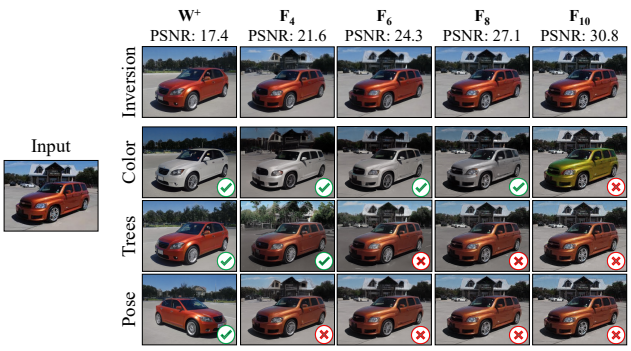

Figure 3. **Trade-offs between invertibility and editibility.** We show inversion and editing when the input is inverted using different *single* latent layers. As we go down in feature space reconstruction improves but editing capabilities decreases. The improvement in reconstruction is shown visually for a single image and quantitatively with PSNR using 1000 images. Whether the edit was applied successfully is indicated by ✓ and ✗.

image using a single latent layer, and observe that the reconstruction quality improves monotonically as we use layers increasingly closer to the output pixels.

Committing to a *single* latent layer for the whole image forces us to a single operating point on the trade-off between editability and reconstruction, across the whole image. Instead, we aim to *adapt* the latent layer selection, depending on the image content in a region. To do this, for each image segment, we choose the earliest latent layer, such that the reconstruction still meets some minimum criteria.

More concretely, for each segment, we choose the most editable latent space from Φ (W^+ being most editable and F_{10} being least), with predicted invertibility above threshold τ for that segment. We choose this threshold value empirically such that the inversion is perceptually close to the input image, without severely sacrificing editability. In Figure 4, we show our final inversion map, with different latent spaces assigned to different segments in the input image. The simple car region gets assigned to the W^+ space, whereas the difficult to generate background regions, which typically cannot be generated by the native latent space, gets assigned to the later F_4 and F_6 latent spaces.

3.3. Training Objective

We implement our multilayer inversion in two settings: 1) optimization-based and 2) encoder-based. In the optimization-based approach, we directly optimize the latent space ϕ for each image. For the encoder-based approach, we train a separate encoder for each latent space.

Image formation model. In Figure 4, we show how the latent codes are combined to generate the final image. Our predicted $w^+ \in W^+$ is directly used to modulate the layers of pre-trained StyleGAN2. For feature spaces $F \in \{F_4, F_6, F_8, F_{10}\}$, we predict the change in values Δf for the regions that are to be inverted in that layer. We predict

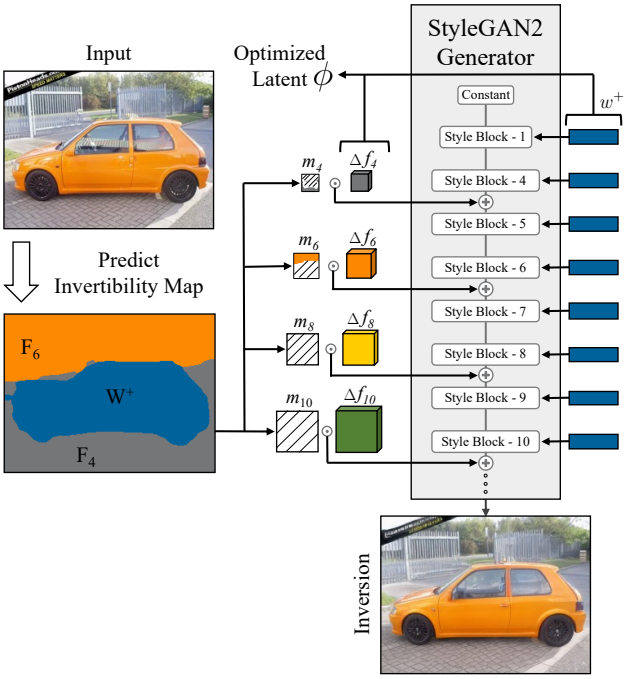

Figure 4. **Image formation using spatially adaptive latent codes.** We show how the predicted invertibility map is used in conjunction with multiple latent codes to generate the final image. $w^+ \in W^+$ directly modulates the StyleBocks of the pretrained StyleGAN2 model. For intermediate feature space F_i, we predict the change in layer's feature value Δf_i and add it to the feature block after masking with the corresponding binary mask m_i.

the change in layer's feature, rather than directly predicting the feature itself, as propagating the features from earlier layers provides a meaningful initialization to adjust from.

The output feature value is a combination of both w^+ and Δf masked by a binary mask indicating which region should be inverted in that layer. For example, to produce the feature $f_4 \in F_4$, we have:

$$f_4 = g_{0\to4}(c, w^+) + m_4 \odot \Delta f_4, \tag{3}$$

where $g_{i\to j}$ denotes the module from the i-th to the j-th layers in the convolutional layers of the StyleGAN2, c is the input constant tensor used in StyleGAN2, m_4 is the refined, predicted invertibility mask bilinearly downsampled to corresponding tensor size, and $\odot$ denotes the Hadamard product. Note that $g_{i\to j}$ is modulated by the corresponding part of the extended latent code w^+. Similarly, we can calculate all the features and the final output image as follows:

$$\begin{aligned} f_6 &= g_{4\to6}(f_4, w^+) + m_6 \odot \Delta f_6 \\ f_8 &= g_{6\to8}(f_6, w^+) + m_8 \odot \Delta f_8 \\ f_{10} &= g_{8\to10}(f_8, w^+) + m_{10} \odot \Delta f_{10} \\ \hat{x} &= g_{10\to16}(f_{10}, w^+). \end{aligned} \tag{4}$$

Next, we present our objective functions to optimize the

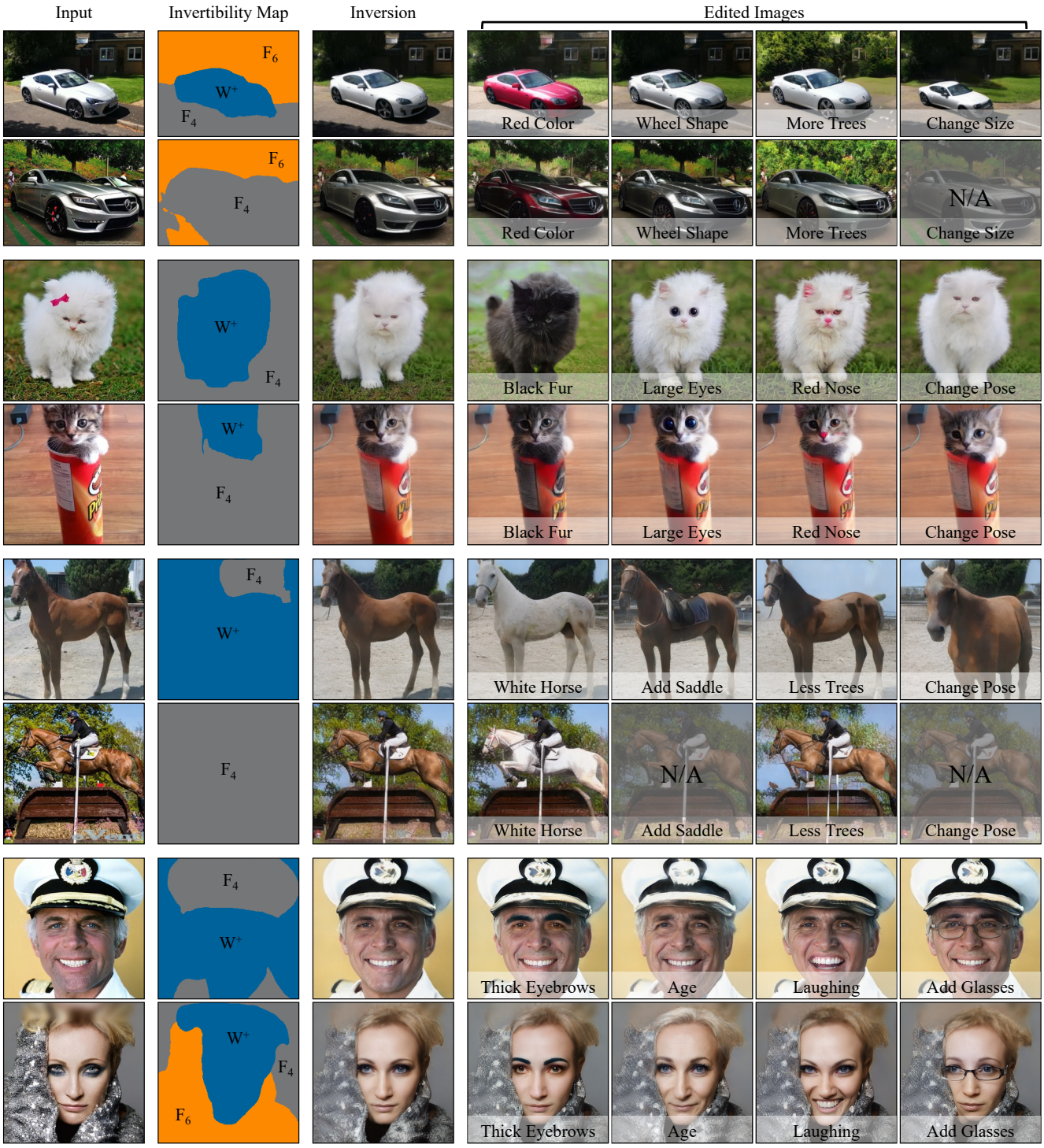

Figure 5. **Qualitative inversion and editing results.** In the first column we show input images for which we predict the invertibility map shown in the second column. We are able to obtain inverted images which closely match the input as shown in third column. In the remaining columns, we show our edit results. We can apply complex spatial edits like pose and size changes in seamless fashion even though different segments were inverted in different latent spaces.

latent code $\phi = \{w^+, \Delta f_4, \Delta f_6, \Delta f_8, \Delta f_{10}\}$. We reconstruct the input image while regularizing the latent codes.

Reconstruction losses. We use the $\mathcal{L}_2$ distance between the inverted image $\hat{x}$ and the input image x along with LPIPS difference as our reconstruction losses.

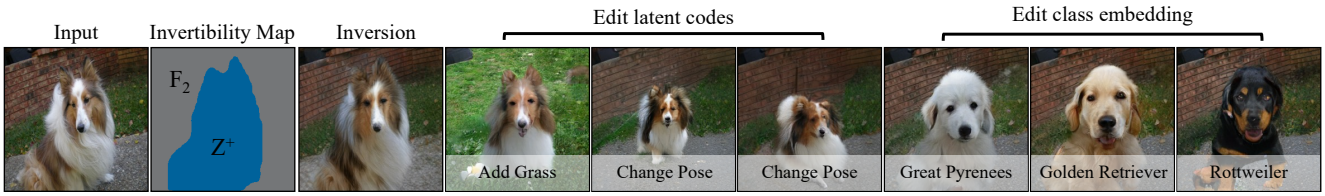

Figure 6. **Inversion and editing using BigGAN-deep.** We show that our spatially-adaptive method of using different latent layers (Z^+, F_2) can be applied to class-conditional models such as BigGAN-deep [12] trained on ImageNet. In the third column we show that the inversion obtained is very close to the input image. Subsequent edits can be performed using either changing the latent code (top row) or modifying class embedding vector (bottom row).

$$\mathcal{L}_{\text{rec}} = \ell_2(x, \hat{x}) + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}(x, \hat{x}), \tag{5}$$

where λ_{LPIPS} is the weight term.

W-space regularization. As noted in [51, 56], inverting an image with just reconstruction losses results in latent codes that are not useful for editing. For our inversion methods, we use different latent regularization losses for different latent spaces. For w^+, we use the following:

$$\mathcal{L}_W = \sum_n^N \left[(\hat{w}_n - \mu)^T \Sigma (\hat{w}_n - \mu) + ||w_n^+ - w_0^+||^2 \right], \tag{6}$$

where w_n^+ is the n^{th} component of the w^+ vector, $\hat{w}_n = \texttt{LeakyReLU}(w_n^+, 5.0)$, μ and Σ are the empirical mean and covariance matrix of randomly sampled W space vectors respectively. The first term applies a Mutlivariate Gaussian prior [56], and the second term minimizes the variation between the individual style codes and the first style code.

F-space regularization. For the feature space, we enforce our predicted change Δf to be small, so that our final feature value does not deviate much from the original value.

$$\mathcal{L}_F = \sum_{\Delta f \in \phi \setminus w^+} ||\Delta f||^2 \tag{7}$$

Final objective. Our full objective is written as follows:

$$\arg\min_{\phi} \mathcal{L}_{\text{rec}} + \lambda_W \mathcal{L}_W + \lambda_F \mathcal{L}_F, \tag{8}$$

where λ_W and λ_F control the weights for each term.

3.4. Image Editing

After obtaining the inverted latent codes, we edit the images by applying the edit direction vector to the inverted w^+ latent vector. We use GANSpace [21] and StyleCLIP [40] for finding an editing direction δw^+ in the W^+ latent space. Segments inverted in W^+ space get modulated by the entire code $w^+ + \delta w^+$, whereas segments inverted in intermediate feature spaces $\{F_4, F_6, F_8, F_{10}\}$ get modulated by $w^+ + \delta w^+$ only for the layers which come after that feature space layer. For example, segments inverted in F_{10} space get modulated by w^+ for the layers until the 10^{th} layer, and $w^+ + \delta w^+$ for the layers afterward. This is necessary, as our inverted feature would not be compatible with $w^+ + \delta w^+$.

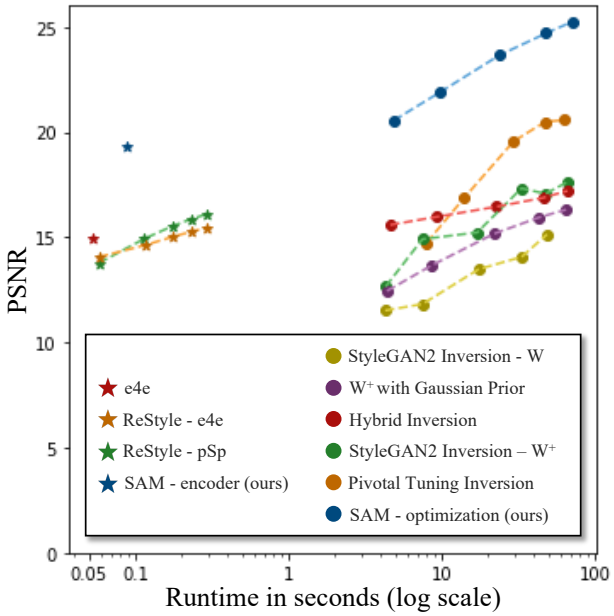

Figure 7. **Reconstruction at different runtimes.** We compare the reconstruction of different GAN inversion methods in the optimization and encoder regimes using 1000 car images. Each of the method uses a single NVIDIA RTX 3090 GPU. Our proposed method achieves a closer reconstruction to the input in a shorter amount of time for both the optimization and encoder paradigms.

4. Experiments

Here we perform detailed quantitative and qualitative analysis to show effectiveness of our inversion method across different datasets. Please refer to the arXiv version for additional details including datasets, BigGAN inversion details, LPIPS architecture variations, more qualitative results, face editing experiments, and ablation studies.

Datasets. We test our method on pretrained StyleGAN2 and BigGAN-deep generators trained on a variety of different challenging domains and follow the commonly used protocol for the different domains [5, 43, 44]. For all experiments we use the official released StyleGAN2 [33] trained on LSUN Cars, LSUN Horses, LSUN Cats, and FFHQ [32] datasets, and the official released BigGAN-deep [12] trained on ImageNet [45]. We use a subset of 10,000 images from the dataset for training our invertibility prediction

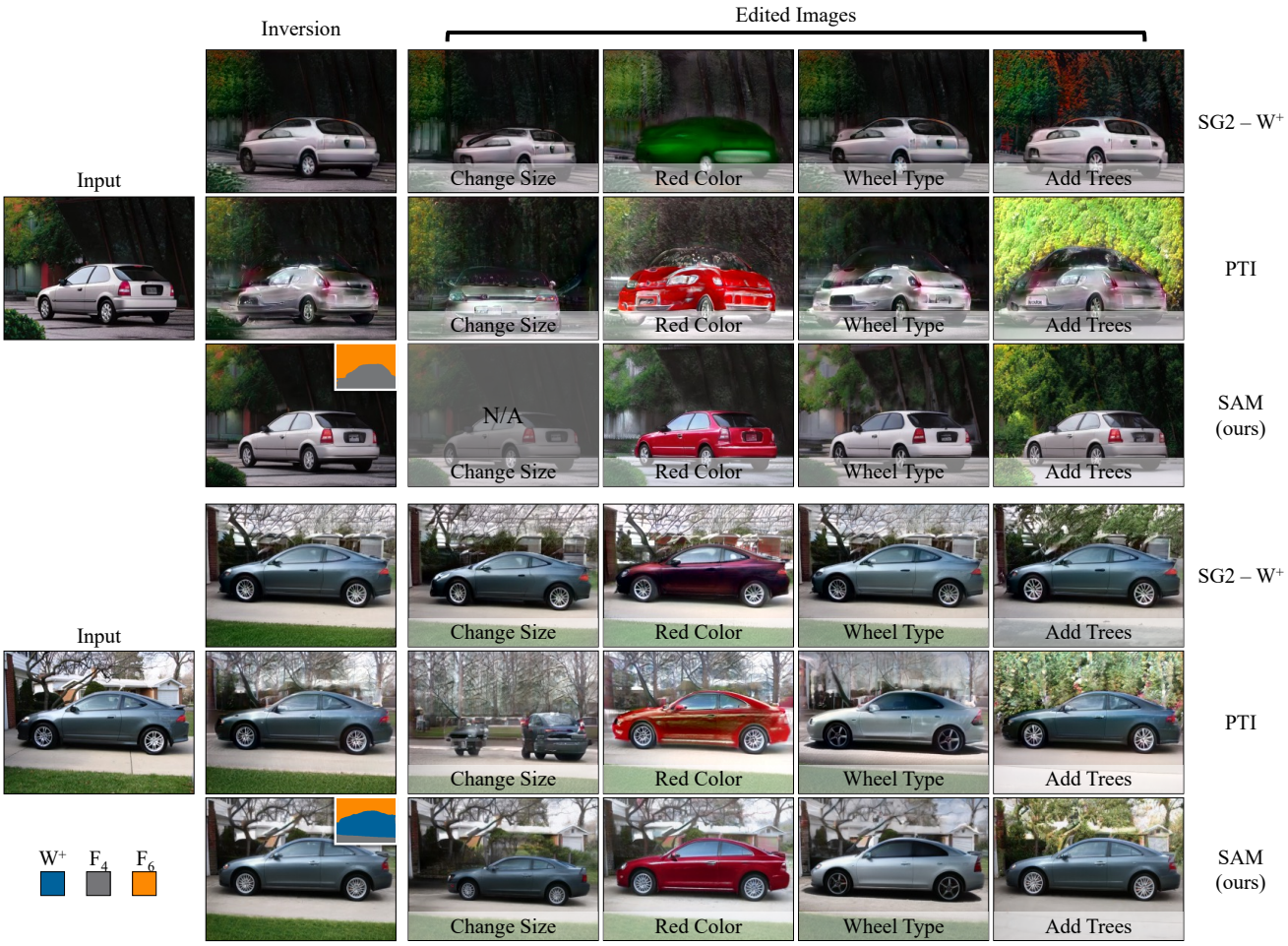

Figure 8. **Comparison with other optimization based inversion methods.** We compare our inversion and editing results with StyleGAN2 W^+ inversion and pivotal tuning. We obtain much closer and detailed inversion to the target image compared to other approaches. Also, we are able to apply semantic edits while maintaining the realism of the image. We are able to perform both low level edits like color change as well as high level edits like size changes. Additional results on other categories are shown on the project website.

network S and 1000 images for the evaluation.

Evaluation. We evaluate the performance of various inversion methods on two tasks - reconstruction and editibility. The reconstruction between the inverted image and the input image is measured using PSNR and LPIPS [60]. Note that different prior inversion methods use different LPIPS backbones. We use LPIPS-VGG for all of our experiments and comparisons. As pointed out by [51], measuring the editing ability of the latent codes is difficult and image quality metrics such as IS [46], FID [22] and KID [11] do not correlate with the user preference. Therefore we show qualitative comparisons and conduct user preference studies to evaluate the quality of inverted and edited images.

Reconstruction comparison. We first compare our inversion method to other state-of-the-art GAN inversions methods in the optimization-based regime. *StyleGAN2 Inversion* and *StyleGAN2 Inversion using* W^+ invert image in W and W^+ latent space respectively. [56] applies multivariant Gaussian prior constraint while doing the inversion. We also compare against *Hybrid* W^+ *Inversion* that uses a pretrained *e4e* encoder [51] for initialization. Recently proposed pivotal tuning inversion (*PTI*) [44] additionally finetunes the weights of pre-trained StyleGAN2 after inverting the image in the W space. Table 1 shows that our method achieves better reconstruction across all the metrics compared to baselines. Our approach is able to invert difficult regions using intermediate layers feature space, whereas baselines struggle to invert by just relying on single W and W^+ space. *PTI* has the ability to change the StyleGAN2 weights to invert the image, but it uses heavy locality regularization to discourage the deviation from original weights, which limits its inversion capability. Also, for simpler image parts, we get better inversion as our W^+ latent code just focuses on parts that it can invert. In contrast, other

Method	Cars		Horses		Cats		Faces	
	LPIPS (↓)	PSNR (↑)	LPIPS (↓)	PSNR (↑)	LPIPS (↓)	PSNR (↑)	LPIPS (↓)	PSNR (↑)
StyleGAN2 Inversion [33]	0.34	14.44	0.45	13.46	0.44	14.47	0.28	18.32
StyleGAN2 Inversion [33] using W^+	0.24	17.29	0.34	15.74	0.35	17.11	0.20	22.10
Inversion with a Gaussian Prior [56]	0.45	15.92	0.42	17.19	0.49	17.01	0.15	25.18
Hybrid W^+ Inversion with e4e [51]	0.36	17.05	0.42	16.68	0.42	17.91	0.15	25.13
PTI [44]	0.38	19.39	0.43	18.73	0.41	20.45	0.26	22.36
SAM - optimization (ours)	**0.16**	**22.81**	**0.23**	**21.07**	**0.22**	**22.91**	**0.13**	**26.89**
e4e [51]	0.47	14.57	0.55	13.98	0.56	14.68	0.34	19.39
ReStyle (pSp) [5]	0.43	16.44	0.45	16.53	0.48	17.58	**0.29**	**21.47**
ReStyle (e4e) [5]	0.45	15.61	0.52	14.50	0.53	15.64	0.34	19.72
SAM - encoder (ours)	**0.28**	**19.21**	**0.34**	**18.61**	**0.37**	**18.59**	**0.29**	21.10

Table 1. **Reconstruction comparison to prior methods.** We use PSNR and the LPIPS-VGG for the evaluating the reconstruction using 1000 images. For the challenging categories, we achieve a better reconstruction than all baseline approaches in both the optimization based and encoder based paradigms. The faces images are simpler and contain fewer challenging regions. Subsequently, our method performs slightly better than prior methods when inverting with optimization and similar to the best performing ReStyle (pSp) with encoders.

approaches try to invert both easy and difficult parts using the same code, resulting in sub-optimal inversion even for the easier part. We perform similar comparisons of encoder based methods and show that an encoder trained using our proposed method outperforms the encoder baselines [5,51] on challenging images. On faces, our encoder obtains a similar reconstruction with just a single forward pass as the best performing baseline *ReStyle (pSp)*, which requires five forward passes. We also compare the runtime of optimization-based and encoder-based inversion methods using 1000 Car images in Figure 7. In both paradigms, our method obtains a better reconstruction in a shorter amount of time.

Qualitative results. Next, we show our ability to edit the reconstructed complex images in Figure 5. In the third column, we show our ability to reconstruct difficult regions using more capable latent layers F_4 and F_6, whereas the easy-to-generate regions use the more editable W^+. This separation allows us to perform challenging edits while faithfully reconstructing the target image. Figure 6 shows inversion and editing results for a class-conditioned BigGAN model. In Figure 8, we observe that we get much closer inversion and realistic edits than baselines approaches. In some cases such as the first image, we can preserve even fine-grained details like the type of light and car wheels during editing stage. *StyleGAN2 inversion using* W^+ generates realistic looking images but does not matches the input images well whereas *PTI* generates images that are closer but lack realism, especially after editing. We hypothesize that this is due to the incompatibility between the finetuned weights and the edit directions learned before finetuning.

User study. We additionally conduct a user preference study to evaluate the realism of inverted and edited images. Table 2 compares our method to three closest baselines methods (*PTI* [44], *StyleGAN2 Inversion using* W^+, and *Hybrid* W^+ *Inversion*) using 500 different target images from each category. Every pair is evaluated by 3 randomized and different users, resulting in 1500 comparisons per baseline per category. The results show that users prefer our results over the baselines for all challenging image categories. Note that face images are noticeably easier and merit the separate treatment shown in the arXiv version.

Method	Inversion			Editing		
	Cars	Horses	Cats	Cars	Horses	Cats
PTI [44]	7.0%	11.6%	11.6%	18.4%	16.4%	38.0%
SAM (ours)	**93.0%**	**88.4%**	**88.4%**	**81.6%**	**83.6%**	**62.0%**
SG2-W+	28.0%	24.7%	20.5%	28.8%	35.7%	35.0%
SAM (ours)	**72.0%**	**75.3%**	**79.5%**	**71.2%**	**64.3%**	**65.0%**
e4e hybrid	23.2%	21.8%	22.4%	36.4%	38.3%	44.6%
SAM (ours)	**76.8%**	**78.2%**	**77.6%**	**62.6%**	**61.7%**	**55.4%**

Table 2. **User preference comparison with prior methods.** We invert and edit 500 from each of the image categories and ask 3 different users (1500 pairs per comparison). Results show that images generated by our method are preferred by the users. The spread in the values computed with bootstrapping is $< 2.5\%$.

5. Conclusion and Limitations

Our key idea is that different regions of an image are best inverted using different latent layers. We use this insight to train networks that predict the "inversion difficulty" of different latent layers for any given input image. Image regions that are easy to reconstruct can be inverted using early latent layers, whereas difficult image regions should use the more capable feature space of the intermediate layers. We show inversion and editing results using our proposed multilayer inversion method on multiple challenging datasets. A limitation of this approach is that if a given input image is extremely difficult, our method will predict the use of the later latent layer that will correspond to being able to edit only limited things.

Acknowledgments. We thank Eli Shechtman, Sheng-Yu Wang, Nupur Kumari, Kangle Deng, George Cazenavette, Ruihan Gao, and Chonghyuk (Andrew) Song for useful discussions. We are grateful for the support from Adobe, Naver Corporation, and Sony Corporation.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)*, 40(3):1–21, 2021. 2

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Only a matter of style: Age transformation using a style-based regression model. *ACM Trans. Graph.*, 40(4), 2021. 2

[5] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2, 6, 8

[6] Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, and Peer-Timo Bremer. Mimicgan: Robust projection onto image manifolds with corruption mimicking. *International Journal of Computer Vision*, pages 1–19, 2020. 2

[7] Muhammad Asim, Fahad Shamshad, and Ali Ahmed. Blind image deconvolution using deep generative priors. *arXiv preprint arXiv:1802.04073*, 2018. 2

[8] 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 SIGGRAPH*, 38(4):1–11, 2019. 1, 2

[9] 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. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[10] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[11] Mikołaj Bińkowski, Danika J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 7

[12] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019. 1, 2, 6

[13] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017. 1

[14] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2

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

[16] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14997–15007, 2021. 2

[17] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[18] David Futschik, Michal Lukáč, Eli Shechtman, and Daniel Sỳkora. Real image inversion via segments. *arXiv preprint arXiv:2110.06269*, 2021. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 1, 2

[20] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[21] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Advances in Neural Information Processing Systems*, 2020. 2, 6

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 7

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

[24] 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 (ECCV)*, 2020. 2

[25] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2020. 2

[26] Omer Kafri, Or Patashnik, Yuval Alaluf, and Daniel Cohen-Or. Stylefusion: A generative model for disentangling spatial segments. *arXiv preprint arXiv:2107.07437*, 2021. 2

[27] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[28] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[29] 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 (ICLR)*, 2018. 1, 2

[30] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *NIPS*, 33, 2020. 2

[31] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021. 1, 2

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 6

[33] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 6, 8

[34] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[35] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, 2016. 2

[36] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017. 2

[37] Xuan Luo, Xuaner Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, and Steven M Seitz. Time-travel rephotography. *arXiv preprint arXiv:2012.12261*, 2020. 2

[38] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 2

[39] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems*, 2020. 2

[40] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 6

[41] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 2

[42] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NIPS Workshop on Adversarial Training*, 2016. 2

[43] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020. 2, 6

[44] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 6, 7, 8

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[46] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved

techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016. 7

[47] 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 (TPAMI)*, 2020. 2

[48] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[49] 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. 2

[50] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[51] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. In *ACM SIGGRAPH*, 2021. 2, 3, 6, 7, 8

[52] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning (ICML)*, 2020. 2

[53] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. A simple baseline for stylegan inversion. *arXiv preprint arXiv:2104.07661*, 2021. 2

[54] Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. Towards vivid and diverse image colorization with generative color prior. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[55] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021. 2

[56] Jonas Wulff and Antonio Torralba. Improving inversion and generation diversity in stylegan using a gaussianized latent space. *arXiv preprint arXiv:2009.06529*, 2020. 6, 7, 8

[57] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *arXiv preprint arXiv:2101.05278*, 2021. 2

[58] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13910–13918, 2021. 2

[59] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 7

[61] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Advances in Neural Information Processing Systems*, 2020. 2

[62] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020. 1

[63] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[64] Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: Gan-based image compositing using segmentation masks. *arXiv preprint arXiv:2106.01505*, 2021. 2

[65] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents? *arXiv preprint arXiv:2012.09036*, 2020. 2, 3