High-fidelity GAN Inversion with Padding Space

Qingyan Bai[1⋆], Yinghao Xu[2⋆], Jiapeng Zhu[3], Weihao Xia[4], Yujiu Yang[1†], and Yujun Shen[5]

[1]Shenzhen International Graduate School, Tsinghua University [2]CUHK
[3]HKUST [4]University College London [5]ByteDance Inc.
bqy20@mails.tsinghua.edu.cn xy119@ie.cuhk.edu.hk
{jengzhu0, xiawh3}@gmail.com
yang.yujiu@sz.tsinghua.edu.cn shenyujun0302@gmail.com

Abstract. Inverting a Generative Adversarial Network (GAN) facilitates a wide range of image editing tasks using pre-trained generators. Existing methods typically employ the latent space of GANs as the inversion space yet observe the insufficient recovery of spatial details. In this work, we propose to involve the *padding space* of the generator to complement the latent space with spatial information. Concretely, we replace the constant padding (*e.g.*, usually zeros) used in convolution layers with some instance-aware coefficients. In this way, the inductive bias assumed in the pre-trained model can be appropriately adapted to fit each individual image. Through learning a carefully designed encoder, we manage to improve the inversion quality both qualitatively and quantitatively, outperforming existing alternatives. We then demonstrate that such a space extension barely affects the native GAN manifold, hence we can still reuse the prior knowledge learned by GANs for various downstream applications. Beyond the editing tasks explored in prior arts, our approach allows a more flexible image manipulation, such as the separate control of face contour and facial details, and enables a novel editing manner where users can *customize* their own manipulations highly efficiently.[1]

1 Introduction

Generative Adversarial Network (GAN) [12] has received wide attention due to its capability of synthesizing photo-realistic images [7, 24, 26]. Recent studies have shown that GANs spontaneously learn rich knowledge in the training process, which can be faithfully used to control the generation [6, 21, 40]. However, the emerging controllability is hard to apply to real-world scenarios. That is because the generator in a GAN typically learns to render an image from a randomly sampled latent code, and hence lacks the ability to make inferences on a target sample, limiting its practical usage.

The advent of GAN inversion techniques (*i.e.*, inverting the generation process of GANs) [36, 54, 56] appears to fill in this gap. The core thought is to convert

⋆ Equal contribution. † Corresponding author.
[1] Project page can be found here.

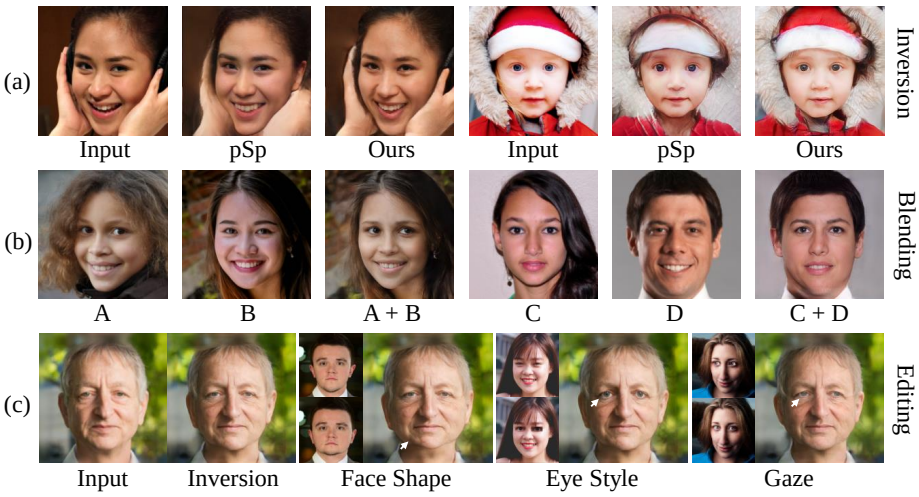

Fig. 1. Inversion and editing results obtained by PadInv. (a) Our method better reconstructs the out-of-distribution objects (*e.g.*, hands and hat) than pSp [37]. (b) Introducing the padding space to complement the latent space allows separate control of face contour and facial details, facilitating face blending. (c) Versatile manipulations can be *customized* with *only one* image pair (*i.e.*, on the left of each editing result)

a given image to some GAN-interpretable representations [47], which can be decoded by the generator to reconstruct the source. That way, real image editing can be simply achieved by manipulating the inverted representations, reusing the pre-trained generator as a learned renderer.

The crux of GAN inversion is to find the appropriate representations that can recover the input as much as possible. A common practice is to regularize the representations within the latent space of GANs [36, 37, 42, 54], which best matches the generation mechanism (*i.e.*, the latent code uniquely determines the synthesis with the generator fixed). However, merely using the latent space suggests unsatisfactory reconstruction performance. The major reason causing such an issue is the inadequate recovery of some out-of-distribution objects, like the hands and hat in face images shown in Fig. 1a.

In this work, we re-examine the procedure of how an image is produced, oriented to GANs with convolution-based generators, and get a deeper understanding of why some spatial details cannot be well recovered. Recall that, to maintain the spatial dimensions of the input feature map, a convolution layer is asked to pad the feature map (*e.g.*, usually with zeros) before convolution. The padding is *not learned* in the training phase and hence introduces some inductive bias [46], which may cause the "texture-sticking" problem [24]. For instance, some padded constants may encode hair information such that hair gets stuck to some synthesized pixels however the latent code varies [24]. Hereafter, when those

pixels are filled with other objects (*e.g.*, hat) in the target image, it becomes hard to invert them through parameter searching within the latent space.

To alleviate such a problem, we propose a high-fidelity GAN inversion approach, termed as **PadInv**, by *incorporating the padding space of the generator as an extended inversion space* in addition to the latent space. Concretely, we adapt the padding coefficients used in the generator for every individual image instead of inheriting the inductive bias (*e.g.*, zero padding) assumed in GAN training. Such an instance-aware reprogramming is able to complement the latent space with adequate spatial information, especially for those out-of-distribution cases. It is noteworthy that the convolutional weights are still shared across samples, leaving the GAN manifold preserved. Consequently, the prior knowledge learned in the pre-trained model is still applicable to the inversion results. We also carefully tailor an encoder that is compatible with the newly introduced padding space, such that an image can be inverted accurately and efficiently.

We evaluate our algorithm from the perspectives of both inversion quality and image editing. On the one hand, PadInv is capable of recovering the target image with far better spatial details than state-of-the-art methods, as exhibited in Fig. 1a. On the other hand, our approach enables two novel editing applications that have not been explored by previous GAN inversion methods. In particular, we manage to control the image generation more precisely, including the *separate manipulation of spatial contents and image style*. As shown in Fig. 1b, we achieve face blending by borrowing the contour from one image and facial details from another. Furthermore, we come up with an innovative editing manner, which *allows users to define their own editing attributes*. As shown in Fig. 1c, versatile manipulations are customized with *only one* image pair, which can be created super efficiently either by graphics editors like Photoshop (*e.g.*, face shape) or by the convenient copy-and-paste (*e.g.*, eye style).

2 Related Work

Generative Adversarial Networks (GANs). Recent years have witnessed the tremendous success of GANs in producing high-resolution, high-quality, and highly-diverse images [7, 24–26]. Existing studies on GAN interpretation have affirmed that, pre-trained GAN models own great potential in a range of downstream applications, such as object classification [11, 47], semantic segmentation [51], video generation [24], and image editing [5, 13, 15, 21, 29, 32, 39, 40, 44, 48, 52, 53].
GAN Inversion. GAN inversion [45, 56] aims at finding the reverse mapping of the generator in GANs. Active attempts broadly fall into two categories, which are optimization-based [8, 13, 18, 30, 33, 36] and learning-based [2, 34, 35, 37, 42, 47, 54–56]. However, they typically employ the native latent space of the generator as the inversion space yet observe inadequately recovered spatial details. Some also [3, 10, 38] try to fine-tune the pre-trained generator, leading to better inversion but much slower speed. To make GAN inversion spatially aware, some attempts [1, 26] incorporate the noise space of StyleGAN [25], while some concurrent studies [22, 43, 57] propose to invert an image to an intermediate

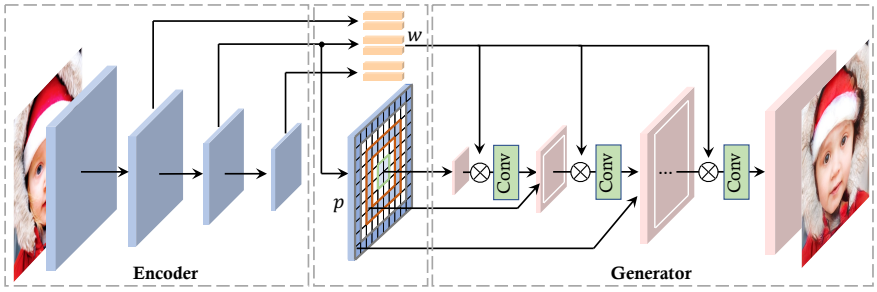

Fig. 2. Framework of the proposed PadInv, which learns an encoder to invert the generation of a pre-trained convolution-based generator, $G(\cdot)$. Given a target image, our encoder not only maps it to layer-wise latent codes (*i.e.*, $\mathbf{w}$), but also produces a coefficient tensor (*i.e.*, $\mathbf{p}$) to *replace the padding* used in the convolution layers of $G(\cdot)$. $\otimes$ denotes the AdaIN [17] operation. The convolutional kernels of $G(\cdot)$ are *fixed* in the training phase and *shared* across all samples

feature map such that only half of the generator will be used as the decoder. By contrast, we involve the *padding space* of the generator to complement the latent space with spatial information. In doing so, the prior knowledge contained in the pre-trained model is barely affected, and hence our inversion results support various editing tasks well.

Padding in Convolutional Neural Networks (CNNs). The side effect of constant padding used in CNNs has recently been studied [4, 19, 20, 24, 27, 47]. It is revealed that padding would implicitly offer the absolute spatial location as the inductive bias [19, 20, 27], which may bring blind spots for object detectors [4]. A similar phenomenon is also observed in generative models [46]. The unwanted spatial information would be utilized by the generator to learn fixed texture in some coordinates, also known as the "texture-sticking" problem [24]. In this work, we manage to *convert such a side effect to an advantage for GAN inversion* through replacing the constant padding assumed in the pre-trained generator with instance-aware coefficients.

3 Method

3.1 Preliminaries

The StyleGAN family [24–26] has significantly advanced image generation of convolution-based GANs. With the start-of-the-art synthesis performance, they are widely used for GAN inversion studies [1, 2, 22, 36, 37, 42, 43, 47, 54, 57]. Like most GAN variants [12], the generator of StyleGAN takes a latent code, $\mathbf{z}$, as the input and outputs an image, $\mathbf{x} = G(\mathbf{z})$. Differently, StyleGAN [25] learns a more disentangled latent space, $\mathcal{W}$, on top of the original latent space, $\mathcal{Z}$, and feeds the disentangled latent code, $\mathbf{w}$, to every single convolution layer through AdaIN [17]. Prior arts have verified that $\mathcal{W}^+$ space [36], which is an extension of $\mathcal{W}$ via

repeating $\mathbf{w}$ across layers, yields the most promising results for GAN inversion. However, existing approaches still suffer from the insufficient recovery of some spatial details, like hats and earrings within face images. We attribute such an issue to the fact that AdaIN acts on the feature map globally [17,25], therefore, the latent code fails to provide adequate spatial information for image reconstruction. In this work, we propose an innovative solution, which complements the latent space with the padding space of the generator.

3.2 Padding Space for GAN Inversion

Recall that the StyleGAN generator [25,26] stacks a series of convolution layers (*i.e.*, with 3×3 kernel size) for image generation. For each layer, it pads the input feature map before convolution to keep its spatial dimensions unchanged. Zero padding is commonly adopted and frozen in GAN training. Recent studies [24,47] have pointed out that, the constant padding acts as an inductive bias through exposing the spatial location to the generator [47], and hence causes the "texture-sticking" problem [24]. In other words, some coordinates may *get stuck with fixed texture* no matter what latent code is given [24]. Such a side effect of padding complicates the inversion task, especially when the target image contains out-of-distribution objects (*e.g.*, a hat appears at the pixels that are supposed to be hair as encoded by padding). From this perspective, merely employing the latent space as the inversion space is highly insufficient for a fair reconstruction.

To tackle the obstacle, we propose to *adapt the padding coefficients* used in the generator for high-fidelity GAN inversion. That way, the inductive bias offered in the generator can be appropriately adjusted to fit every individual target. We call the parameter space, which is constructed by those instance-aware convolutional paddings, as the padding space, $\mathcal{P}$. Intuitively, $\mathcal{P}$ is incorporated as an extended inversion space to complement the latent space (*i.e.*, $\mathcal{W}^+$) with spatial information. Besides, the StyleGAN generator learns image synthesis starting from a constant tensor [25]. The constant input can be viewed as a special padding (*i.e.*, from a tensor with shape 0×0) without performing convolution. In our approach, the initial constant is also adapted for each target sample.

3.3 Encoder Architecture

To accelerate the inversion speed, we learn an encoder, $E(\cdot)$, following prior works [2,37,42,54]. The architecture of the encoder is carefully tailored to get compatible with the newly introduced padding space, $\mathcal{P}$. In particular, in addition to the layer-wise latent codes, $\{\mathbf{w}_\ell\}_{\ell=1}^{L}$, our encoder also predicts a coefficient map from the input, as shown in Fig. 2. Here, L stands for the total number of layers in the generator. Values from the coefficient map will be used to replace the paddings used in $G(\cdot)$, denoted as $\{\mathbf{p}_\ell\}_{\ell=0}^{L}$, at the proper resolution. $\mathbf{p}_0$ represents the initial constant input. For example, the coefficients at the outer of the central 18×18 patch are borrowed as the padding for the convolution at 16×16 resolution. Note that, we only apply one set of coefficients for each resolution to avoid the two convolution layers of the same resolution from sharing

the same padding. In practice, we replace $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5, \ldots$, keeping the padding for other layers untouched.

Our encoder backbone is equipped with several residual blocks [16], and a Feature Pyramid Network (FPN) [28] is adopted to inspect the input at multiple levels. The outputs of the last three blocks (*i.e.*, with the smallest resolutions) are employed to learn the latent codes, $\{\mathbf{w}_\ell\}_{\ell=1}^{L}$, in a hierarchical manner. They correspond to the generator layers in reverse order. Taking an 18-layer StyleGAN generator as an example, the last block produces latent codes for layers 1-3, the second last for layers 4-7, while the third last for layers 8-18, respectively. Meanwhile, we also choose a max resolution[2] where padding replacement will be performed. The coefficient map is projected (*i.e.*, implemented with a convolution layer with 1×1 kernel size) from the FPN feature with target resolution, and then properly resized to allow padding in the generator. For instance, given the max resolution as 32, we convolve the 32×32 FPN feature and upsamples the convolution result to 34×34, which is the minimum size to pad 32×32 features. More details can be found in *Supplementary Material*.

3.4 Training Objectives

Encoder. We develop the following loss functions for encoder training.

1. *Reconstruction loss.* Image reconstruction is the primary goal of GAN inversion. Hence, we optimize the encoder with both pixel guidance and perceptual guidance, as

$$\mathcal{L}_{pix} = ||\mathbf{x} - G(E(\mathbf{x}))||_2, \tag{1}$$
$$\mathcal{L}_{per} = ||\phi(\mathbf{x}) - \phi(G(E(\mathbf{x})))||_2, \tag{2}$$

   where $||\cdot||_2$ denotes the l_2 norm. $\phi(\cdot)$ is a pre-trained perceptual feature extractor [50], which is popularly used for GAN inversion [37,54].
2. *Identity loss.* This loss function is particularly designed for inverting face generation models, following [37]. It can help the encoder to focus more on the face identity, which is formulated as

$$\mathcal{L}_{id} = 1 - \cos(\psi(\mathbf{x}), \psi(G(E(\mathbf{x})))), \tag{3}$$

   where $\cos(\cdot, \cdot)$ computes the cosine similarity. $\phi(\cdot)$ is a pre-trained identity feature extractor [9]. This loss will be *disabled* for non-face models.
3. *Adversarial loss.* To avoid blurry reconstruction [37], the encoder is also asked to compete with the discriminator, resulting in an adversarial training manner. The adversarial loss is defined as

$$\mathcal{L}_{adv} = -\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}}[D(G(E(\mathbf{x})))], \tag{4}$$

   where $\mathbb{E}[\cdot]$ and $\mathcal{X}$ are the expectation operation and the real data distribution, respectively. $D(\cdot)$ stands for the discriminator.

[2] We empirically verify that 32 is the best choice in Sec. 4.2.

4. *Regularization loss.* Recall that StyleGAN generator maintains an averaged latent code, $\overline{\mathbf{w}}$, which can be viewed as the statistics of the latent space $\mathcal{W}$. We expect the inverted code to be subject to the native latent distribution as much as possible [42]. Hence, we regularize it with

$$\mathcal{L}_{reg} = ||E_{latent}(\mathbf{x}) - \overline{\mathbf{w}}||_2, \tag{5}$$

where $E_{latent}(\cdot)$ indicates the latent part of the encoder, excluding padding.

To summarize, the full objective for our encoder is:

$$\mathcal{L}_E = \lambda_{pix}\mathcal{L}_{pix} + \lambda_{per}\mathcal{L}_{per} + \lambda_{id}\mathcal{L}_{id} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{reg}\mathcal{L}_{reg}, \tag{6}$$

where λ_{pix}, λ_{per}, λ_{id}, λ_{adv}, and λ_{reg} are loss weights to balance different terms.

Discriminator. The discriminator is asked to compete with the encoder, as

$$\mathcal{L}_D = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}}[D(G(E(\mathbf{x})))] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}}[D(\mathbf{x})] - \frac{\gamma}{2}\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{X}}[||\nabla_{\mathbf{x}} D(\mathbf{x})||_2], \tag{7}$$

where γ is the hyper-parameter for gradient penalty [14].

4 Experiments

In this part, we conduct extensive experiments to evaluate the effectiveness of the proposed PadInv. Sec. 4.1 introduces the experimental settings, such as datasets and implementation details. In Sec. 4.2, we experimentally show the superiority of the proposed method PadInv in terms of inversion quality and real image editing with off-the-shelf directions. We also conduct ablation studies. In Sec. 4.3, we explore the property of the proposed padding space, which enables separate control of spatial contents and image style. A face blending application is introduced to further present the superiority of the padding space. Sec. 4.4 shows a novel editing method that could be achieved by providing customized image pairs. We also quantitatively evaluate the editing effects of semantic directions in $\mathcal{W}+$ space and $\mathcal{P}$ space.

4.1 Experimental Settings

We conduct experiments on the high-quality face datasets FFHQ [25] and CelebA-HQ [23,31] for face inversion. We use the former 65k FFHQ faces as the training set (the rest 5k for further evaluation) and test set of CelebA-HQ for quantitative evaluation. For scene synthesis, we adopt LSUN Church and Bedroom [49] and follow the official data splitting strategy. The GANs to invert are pre-trained following StyleGAN [25]. As for the loss weights, we set $\lambda_{id} = 0.1$ and $\lambda_{adv} = 0$ for encoders trained on FFHQ, $\lambda_{adv} = 0.03$ and $\lambda_{id} = 0$ for encoders trained on LSUN. For all the experiments, we set $\lambda_{pix} = 1$, $\lambda_{per} = 0.8$, $\lambda_{reg} = 0.003$, and $\gamma = 10$. Our codebase is built on Hammer [41].

Table 1. Quantitative comparisons between different inversion methods on three datasets, including FFHQ (human face) [25], LSUN Church (outdoor scene) [49], and LSUN Bedroom (indoor scene) [49]

	Face			Church		Bedroom	
	MSE↓	**LPIPS↓**	**Time↓**	**MSE↓**	**LPIPS↓**	**MSE↓**	**LPIPS↓**
StyleGAN2 [26]	0.020	0.09	122.39	0.220	0.39	0.170	0.42
ALAE [35]	0.15	0.32	**0.0208**	-	-	0.33	0.65
IDInvert [54]	0.061	0.22	0.0397	0.140	0.36	0.113	0.41
pSp [37]	0.034	0.16	0.0610	0.127	0.31	0.099	0.34
e4e [42]	0.052	0.20	0.0610	0.142	0.42	-	-
Restyle$_{pSp}$ [2]	0.030	0.13	0.2898	0.090	0.25	-	-
Restyle$_{e4e}$ [2]	0.041	0.19	0.2898	0.129	0.38	-	-
Ours	**0.021**	**0.10**	0.0629	**0.086**	**0.22**	**0.054**	**0.21**

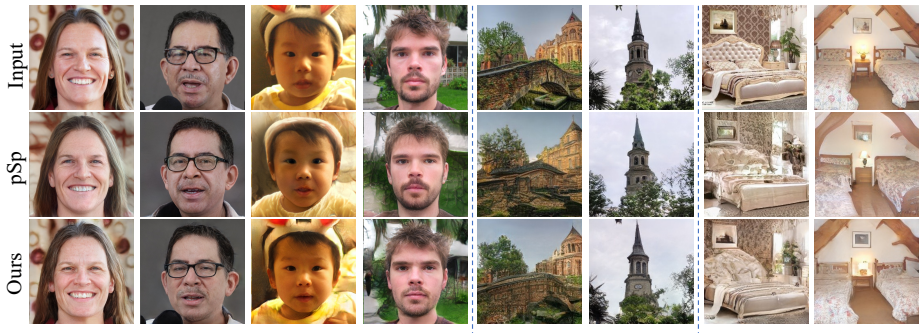

Fig. 3. Qualitative comparisons between our approach and the baseline, pSp [37], where we better recover the spatial details of the input images

4.2 GAN Inversion Performance

Quantitative Results. We quantitatively evaluate the quality of the reconstructed images in Tab. 1. Pixel-wise MSE and perceptual LPIPS [50] are adopted as metrics to evaluate the reconstruction performance. We also include the inference time of face inversion to measure model efficiency. As shown in Tab. 1, our approach leads to a significant improvement on face images compared with encoder-based baselines. It even achieves a comparable performance to the optimization-based StyleGAN2 with better efficiency. When applied to more challenging datasets *e.g.*, church and bedroom, StyleGAN2 fails to reconstruct images well because $\overline{\mathbf{w}}$ as the starting point for optimization cannot handle the unaligned data with large spatial variations. However, our method is robust and still outperforms all baselines. It demonstrates that the proposed padding space can provide fine-grained spatial details of images, resulting in better reconstruction quality.

Qualitative Results. In Fig. 3, we compare the reconstructed images of our method with the baseline method pSp [37] on the face, church, and bedroom.

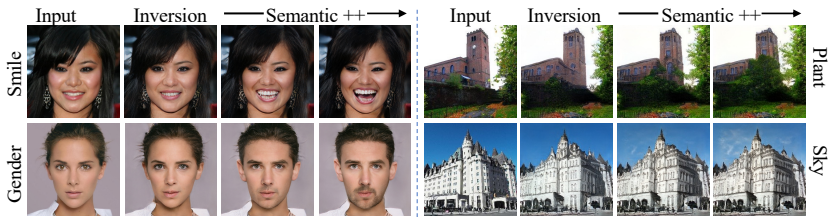

Fig. 4. Real image editing with *off-the-shelf* semantics identified by InterFaceGAN [40] (smile and gender), HiGAN [48] (plant), and LowRankGAN [52] (sky)

Even though pSp can restore the foreground of the given face images, it struggles to reconstruct spatial details, as in Fig. 3 (a), (b), (c), (d). Thanks to the proposed padding space, our approach enhances the spatial modeling capacity of the inversion space so that the fine-grained background texture, microphone, and headwear can be reconstructed well, and the identity can be preserved. When it comes to challenging church and bedroom datasets with large spatial variations, pSp can achieve satisfying performance on restoring the texture but still struggles to reconstruct the spatial structure of the given images, as shown in Fig. 3 (e). Our method can preserve structural details better because the proposed padding space can complement the original $\mathcal{W}+$ space with the instance-aware padding coefficients. For instance, the structural information of buildings, such as the number of spires, is preserved more. The bridge, sheet and painting in Fig. 3(e), (g), (h) are also well reconstructed.

Ablation Studies. The proposed padding space is adopted to complement the $\mathcal{W}+$ space, and thus its scale is critical to the quality of the reconstructed image. We adjust its scale by varying the padding layers. Tab. 2 shows the reconstruction performance on the test sets of CelebA-HQ and LSUN Church. We use pSp [37] as the baseline, which only utilizes $\mathcal{W}+$ space for inversion. We firstly extend the inversion space with $\mathbf{p}_0$, which represents the constant input of the generator. The performance is slightly improved due to the introduction of the case-specific information. Then we progressively enlarge the padding space by predicting padding coefficients for more layers as shown in Fig. 2. As the padding space increases, the reconstruction performance becomes better. This demonstrates that more padding coefficients can provide more precise information for spatial modeling, resulting in the improvement of image reconstruction. When padding layers come to 9 (resolution of 64×64), the inversion performance gets a drop. We assume that the parameter of the padding space is too large, leading to overfitting on the training set. Thus the largest index of padding layers is chosen as 7 (resolution of 32×32) to maintain the reconstruction performance. We also study the effects of the regularization loss $\mathcal{L}_{reg}$. This regularization is designed to encourage the inverted code in $\mathcal{W}+$ space to be subject to the native latent distribution. Equipping $\mathcal{L}_{reg}$ to the model leads to a slight drop in reconstruction performance, but the editing ability becomes better as in *Supplementary Material*.

Table 2. Ablation studies on the number of layers to replace padding, as well as the regularization loss, $\mathcal{L}_{reg}$, introduced in Eq. (5)

	Face		Church	
	MSE↓	**LPIPS↓**	**MSE↓**	**LPIPS↓**
Baseline	0.0344	0.161	0.1272	0.311
$+\mathbf{p}_0$	0.0343	0.156	0.1257	0.307
$+\mathbf{p}_{0,1}$	0.0341	0.154	0.1241	0.309
$+\mathbf{p}_{0,1,3}$	0.0252	0.115	0.1005	0.253
$+\mathbf{p}_{0,1,3,5}$	0.0190	0.091	0.0891	0.218
$+\mathbf{p}_{0,1,3,5,7}$	0.0186	0.090	0.0838	0.209
$+\mathbf{p}_{0,1,3,5,7,9}$	0.0216	0.108	0.0979	0.271
$+\mathbf{p}_{0,1,3,5,7}$ & $\mathcal{L}_{reg}$	0.0214	0.103	0.0866	0.222

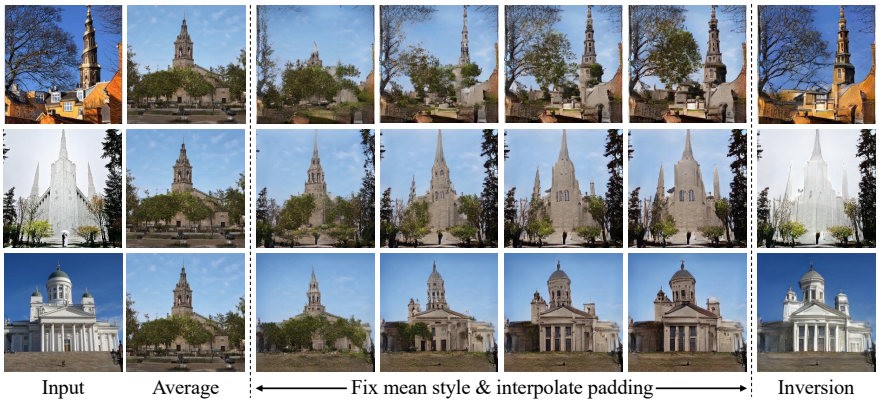

Fig. 5. Analysis of the padding effect on LSUN Church [49]. We fix the latent code as the statistical average and interpolate the padding from the fixed constants in the generator to the coefficients specifically learned for inversion. It verifies that padding encodes the spatial information

Editability. Having verified our method in better reconstruction quality, here we explore its editability. We utilize off-the-shelf semantic directions from [40, 48, 52] to edit the inversion results. Fig. 4 presents the results of manipulating faces and churches. Our method can preserve most other details when manipulating a particular facial attribute. For churches, the semantics changes smoothly on the high-fidelity reconstructed images. These editing results support that the extra padding space can not only invert the given images in high quality but also facilitate them with good properties on image manipulation.

4.3 Separate Control of Spatial Contents and Image Style

Property of Padding Space. In this part, we investigate the proposed padding space's properties using interpolation experiments on the FFHQ and LSUN test sets. We first conduct interpolation experiments between a real image and the average image synthesized with the average latent code $\overline{\mathbf{w}}$ and original paddings.

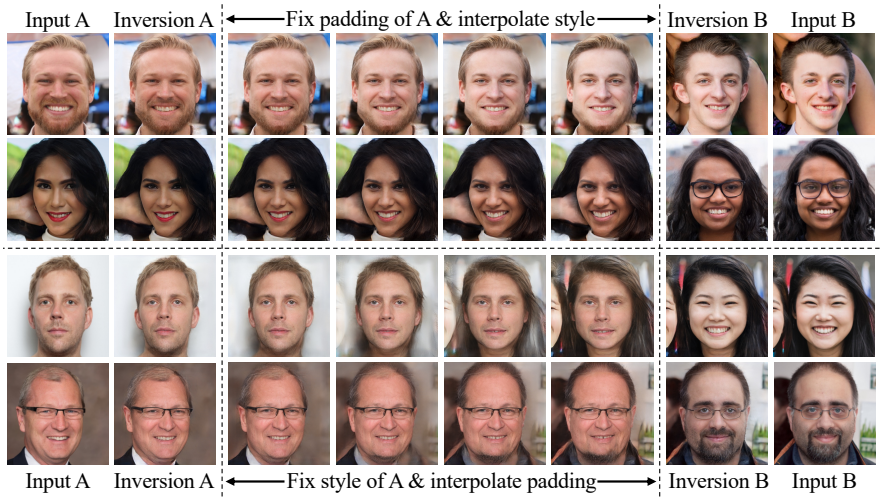

Fig. 6. Analysis of the extended inversion space on FFHQ [25]. We perform interpolation both in the latent space and in the padding space. It turns out that the latent and padding tends to encode style and spatial information, respectively

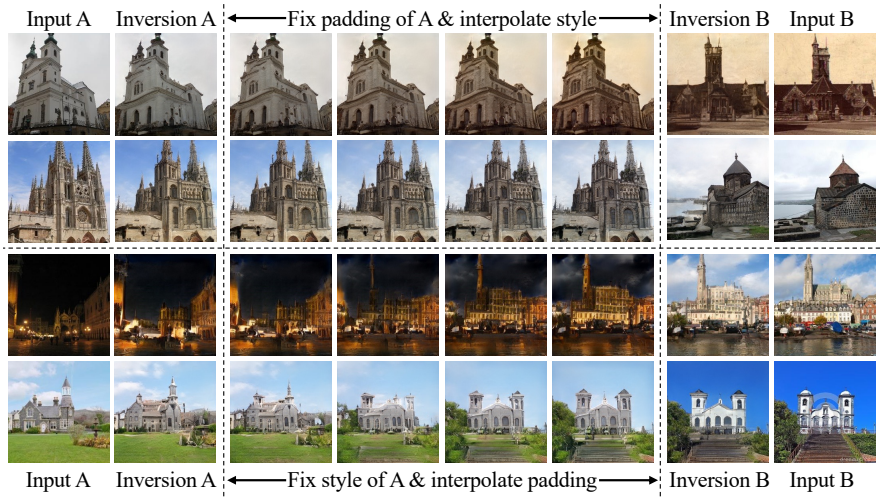

Fig. 7. Analysis of the extended inversion space on LSUN Church [49]. We perform interpolation both in the latent space and in the padding space

The real image is inverted with our encoder into the style latent codes and padding coefficients. The paddings between the average image and the inversion result are then interpolated, while the latent codes from $\mathcal{W}+$ space are kept as $\overline{\mathbf{w}}$. Fig. 5 demonstrates the interpolation results on the test sets of LSUN Church. The results of interpolation on FFHQ and LSUN Bedroom are shown in

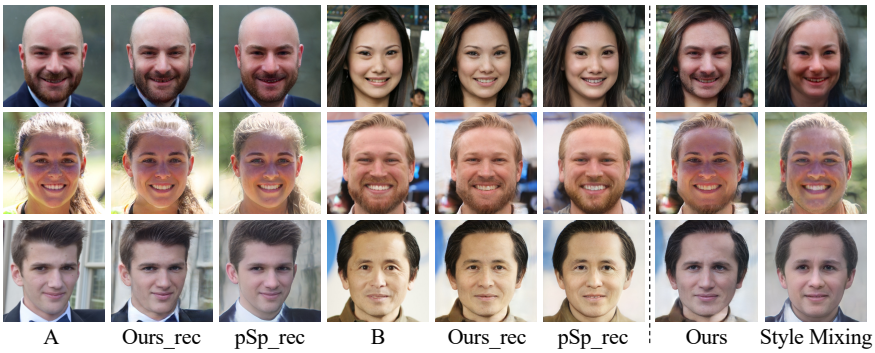

Fig. 8. Face blending results by borrowing the inverted latent of A and the inverted padding of B. Different from style mixing, our approach enables the separate control of the facial details (*e.g.*, the features of A), and the face contour (B)

Supplementary Material. The style of the interpolated images is nearly identical, while the spatial structure is gradually transformed from the average image to the given image. It suggests that the learned padding coefficients encode the structural information of the given images.

Interpolation experiments between two real images are further conducted to validate the separate controllability of the spatial contents and image style. We first invert two real images A and B to $\mathcal{P}$ and $\mathcal{W}+$ space. Then we fix one of the paddings or latent codes and interpolate the other. As illustrated in Fig. 6 and Fig. 7, when padding coefficients are kept to be fixed, the style varies smoothly when the latent code changes. Interpolating paddings with fixed style latent codes leads to the results with the same style and different spatial structures. The interpolation results on LSUN Bedroom can be found in *Supplementary Material*. Above interpolation results indicate that the paddings of $\mathcal{P}$ space encode structure information such as the pose and shape, which is complementary to style information encoded in latent codes of $\mathcal{W}+$ space. We can leverage the properties of two spaces to achieve the independent control of spatial contents and image style.

Application on Face Blending. As discussed in Sec. 4.3, paddings of $\mathcal{P}$ space control the spatial structure of the image. For face images, it controls the face shape and background. Based on this property, we can naturally achieve face blending which integrates the shape and background of one face and facial features of another. Specifically, we invert a real source face A and target portrait B by the encoder. Then the blended face can be synthesized by sending the latent codes of A and padding coefficients of B to the pre-trained generator. In Fig. 8, we compare our face blending results with naively mixing the latter 11 style latent codes as pSp [37]. Obviously, the naive style mixing in pSp introduces redundant information of the source image A and cannot keep the background and hair of the target image B. However, our method can transfer the identity information of face A well and inherit the structure of target face B perfectly. It is worth

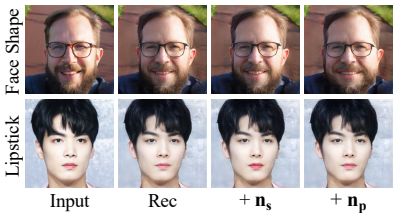

Fig. 9. Visual results on attribute editing within the latent space and the padding space

Attribute	$\mathbf{MSE_p}$	$\mathbf{MSE_s}$	Factor
face shape	116×1e-4	9×1e-4	12.89:1
bangs	471×1e-4	65×1e-4	7.25:1
glasses	245×1e-4	42×1e-4	5.83:1
lipstick	0.437×1e-4	6×1e-4	1:13.73
eyebrow	0.880×1e-4	14×1e-4	1:15.90

Table 3. Qualitative comparisons on latent editing and padding editing. "Factor" indicates the ratio of MSE_p to MSE_s

noting that, the whole face blending process does not require any pre-trained segmentation or landmark models. It leverages the structure information encoded in $\mathcal{P}$ space, enabling the independent control of the face structure and details.

4.4 Manipulation with Customized Image Pair

Implementation. Based on the padding space, we propose a novel editing method. We find the semantic direction defined by one pair of customized image labels can be successfully applied to arbitrary samples. The paired images can be obtained by human retouching or naive copy-and-paste. Specifically, we firstly invert the paired label images A and A' into $\mathcal{P}$ and $\mathcal{W}+$ space. The difference of the padding coefficients $\mathbf{n_p}$ in $\mathcal{P}$ space and difference of style codes $\mathbf{n_s}$ in $\mathcal{W}+$ space can be viewed as the semantic directions defined by the given paired images. Then other real images can be edited using $\mathbf{n_p}$ and $\mathbf{n_s}$. Considering that the paddings and latent codes encode various information, we propose to apply the semantic direction of one (either $\mathcal{P}$ or $\mathcal{W}+$) space, to maintain the editing performance.

Evaluation and Analysis. We firstly explore the effects of $\mathbf{n_p}$ and $\mathbf{n_s}$ with respect to various attributes. As shown in Fig. 9, for a spatial attribute such as face shape, editing with $\mathbf{n_p}$ works fine while $\mathbf{n_s}$ rarely affects the image. However, $\mathbf{n_s}$ works well for the style-related attributes like lipstick while the editing effect brought by $\mathbf{n_p}$ can be ignored. To quantitatively evaluate the editing effects of $\mathbf{n_p}$ and $\mathbf{n_s}$, MSE_p and MSE_s between the inversion and editing results in the non-edited region like [52] are calculated and reported. Specifically, MSE_p is calculated over the 50 randomly-sampled faces and their editing results driven by $\mathbf{n_p}$. MSE_s is obtained in a similar manner with $\mathbf{n_s}$. Tab. 3 presents the quantitative results of semantic directions in different space. The factor between MSE_p and MSE_s is also reported for better comparison. We discover that $\mathbf{n_p}$ influences target faces far more than $\mathbf{n_s}$ for spatial attributes such as face shape and bangs. When it comes to style-related attributes like lipstick and brows, $\mathbf{n_s}$ appears to change the inversion images much more than $\mathbf{n_p}$. Given this property, we can only use the $\mathbf{n_p}$ or the $\mathbf{n_s}$ for spatial or style-related editing, respectively. The use of semantic directions separately is extremely beneficial in avoiding unwanted changes to the inversion result. Fig. 10 presents the editing results

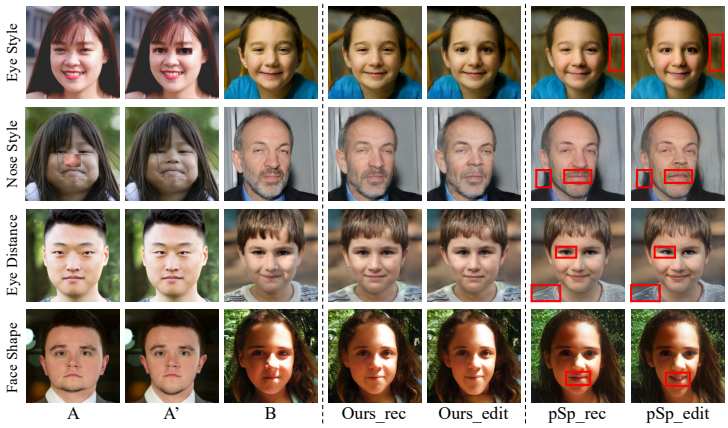

Fig. 10. Customized manipulation defined by *only one* image pair (A and A'). Compared to pSp [37], which introduces unwanted changes to the target image (B), our approach presents a more precise editing performance

of $\mathbf{n_p}$ and $\mathbf{n_s}$. We also include the one-shot editing results of pSp, which can be achieved in $\mathcal{W}+$ space. Naively computing latent difference and applying these directions in $\mathcal{W}+$ space with pSp often interferes with other unconcerned attributes. For example, a change to eye or nose style can be accompanied by a change in the background. Besides, the face shape direction in $\mathcal{W}+$ space produces a larger smile in the portrait, whereas ours does not affect any attributes other than face shape. Additional customized manipulation results of other attributes are provided in *Supplementary Material*.

5 Conclusion

In this work, we attribute the unsatisfactory GAN inversion performance to inductive bias brought by zero padding. Thus we propose padding space ($\mathcal{P}$ space) to complement $\mathcal{W}+$ space for the reconstruction of spatial details, particularly for the out-of-distribution objects. A carefully designed encoder is further proposed to learn the latent code and the padding coefficients jointly. The experiments demonstrates that the padding coefficients can encode rich spatial information such as pose and contour and thus our method enables two novel applications, *i.e.*, face blending and one-shot customized editing. However, the distribution gap between the learned paddings and original features occasionally causes artifacts at the edge. This phenomenon can be largely relieved by adversarial training. We strongly oppose the abuse of our method in violating privacy and security. We hope it can be used to improve the fake detection systems.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (Grant No. 61991450) and the Shenzhen Key Laboratory of Marine IntelliSense and Computation (ZDSYS20200811142605016). We thank Zhiyi Zhang for the technical support.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2StyleGAN++: How to edit the embedded images? In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
2. Alaluf, Y., Patashnik, O., Cohen-Or, D.: ReStyle: A residual-based StyleGAN encoder via iterative refinement. In: Int. Conf. Comput. Vis. (2021)
3. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 18511–18521 (2022)
4. Alsallakh, B., Kokhlikyan, N., Miglani, V., Yuan, J., Reblitz-Richardson, O.: Mind the pad–cnns can develop blind spots. In: Int. Conf. Learn. Represent. (2021)
5. Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A.: Paint by word. arXiv preprint arXiv:2103.10951 (2021)
6. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: GAN dissection: Visualizing and understanding generative adversarial networks. In: Int. Conf. Learn. Represent. (2019)
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: Int. Conf. Learn. Represent. (2019)
8. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. TNNLS (2018)
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019)
10. Dinh, T.M., Tran, A.T., Nguyen, R., Hua, B.S.: Hyperinverter: Improving stylegan inversion via hypernetwork. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11389–11398 (2022)
11. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: Adv. Neural Inform. Process. Syst. (2019)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural Inform. Process. Syst. (2014)
13. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code GAN prior. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein GANs. In: Adv. Neural Inform. Process. Syst. (2017)
15. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: GANSpace: Discovering interpretable GAN controls. In: Adv. Neural Inform. Process. Syst. (2020)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. (2016)
17. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Int. Conf. Comput. Vis. pp. 1501–1510 (2017)
18. Huh, M., Zhang, R., Zhu, J.Y., Paris, S., Hertzmann, A.: Transforming and projecting images to class-conditional generative networks. In: Eur. Conf. Comput. Vis. (2020)
19. Islam, M.A., Jia, S., Bruce, N.D.: How much position information do convolutional neural networks encode? Int. Conf. Learn. Represent. (2019)
20. Islam, M.A., Kowal, M., Jia, S., Derpanis, K.G., Bruce, N.D.: Position, padding and predictions: A deeper look at position information in CNNs. arXiv preprint arXiv:2101.12322 (2021)
21. Jahanian, A., Chai, L., Isola, P.: On the "steerability" of generative adversarial networks. In: Int. Conf. Learn. Represent. (2020)

22. Kang, K., Kim, S., Cho, S.: GAN inversion for out-of-range images with geometric transformations. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13941–13949 (2021)
23. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: Int. Conf. Learn. Represent. (2018)
24. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Adv. Neural Inform. Process. Syst. (2021)
25. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019)
26. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
27. Kayhan, O.S., Gemert, J.C.v.: On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
28. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2117–2125 (2017)
29. Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S.: EditGAN: High-precision semantic image editing. In: Adv. Neural Inform. Process. Syst. (2021)
30. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. In: Int. Conf. Learn. Represent. Worksh. (2017)
31. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Int. Conf. Comput. Vis. (2015)
32. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. In: Eur. Conf. Comput. Vis. (2020)
33. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. In: Eur. Conf. Comput. Vis. (2020)
34. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional GANs for image editing. In: Adv. Neural Inform. Process. Syst. Worksh. (2016)
35. Pidhorskyi, S., Adjeroh, D.A., Doretto, G.: Adversarial latent autoencoders. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
36. Rameen, A., Yipeng, Q., Peter, W.: Image2StyleGAN: How to embed images into the stylegan latent space? In: Int. Conf. Comput. Vis. (2019)
37. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a StyleGAN encoder for image-to-image translation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
38. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744 (2021)
39. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of GANs for semantic face editing. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
40. Shen, Y., Yang, C., Tang, X., Zhou, B.: InterFaceGAN: Interpreting the disentangled face representation learned by GANs. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
41. Shen, Y., Zhang, Z., Yang, D., Xu, Y., Yang, C., Zhu, J.: Hammer: An efficient toolkit for training deep models. https://github.com/bytedance/Hammer (2022)
42. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for StyleGAN image manipulation. ACM Trans. Graph. (2021)

43. Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: High-fidelity GAN inversion for image attribute editing. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
44. Wu, Z., Lischinski, D., Shechtman, E.: StyleSpace analysis: Disentangled controls for StyleGAN image generation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
45. Xia, W., Zhang, Y., Yang, Y., Xue, J.H., Zhou, B., Yang, M.H.: GAN Inversion: A survey. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
46. Xu, R., Wang, X., Chen, K., Zhou, B., Loy, C.C.: Positional encoding as spatial inductive bias in GANs. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
47. Xu, Y., Shen, Y., Zhu, J., Yang, C., Zhou, B.: Generative hierarchical features from synthesizing images. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
48. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. Int. J. Comput. Vis. (2020)
49. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
50. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. (2018)
51. Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.F., Barriuso, A., Torralba, A., Fidler, S.: DatasetGAN: Efficient labeled data factory with minimal human effort. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
52. Zhu, J., Feng, R., Shen, Y., Zhao, D., Zha, Z., Zhou, J., Chen, Q.: Low-rank subspaces in GANs. In: Adv. Neural Inform. Process. Syst. (2021)
53. Zhu, J., Shen, Y., Xu, Y., Zhao, D., Chen, Q.: Region-based semantic factorization in GANs. In: Int. Conf. Mach. Learn. (2022)
54. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain GAN inversion for real image editing. In: Eur. Conf. Comput. Vis. (2020)
55. Zhu, J., Zhao, D., Zhang, B., Zhou, B.: Disentangled inference for GANs with latently invertible autoencoder. Int. J. Comput. Vis. (2022)
56. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Eur. Conf. Comput. Vis. (2016)
57. Zhu, P., Abdal, R., Femiani, J., Wonka, P.: Barbershop: GAN-based image compositing using segmentation masks. arXiv preprint arXiv:2106.01505 (2021)