High-Fidelity Image Inpainting with GAN Inversion

Yongsheng Yu[1,2], Libo Zhang[1,2,3](✉), Heng Fan[4], and Tiejian Luo[2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
yuyongsheng19@mails.ucas.ac.cn, libo@iscas.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
tjluo@ucas.ac.cn
[3] Nanjing Institute of Software Technology, Nanjing, China
[4] Department of Computer Science and Engineering, University of North Texas, Denton, USA
heng.fan@unt.edu

Abstract. Image inpainting seeks a semantically consistent way to recover the corrupted image in the light of its unmasked content. Previous approaches usually reuse the well-trained GAN as effective prior to generate realistic patches for missing holes with GAN inversion. Nevertheless, the ignorance of hard constraint in these algorithms may yield the gap between GAN inversion and image inpainting. Addressing this problem, in this paper we devise a novel GAN inversion model for image inpainting, dubbed *InvertFill*, mainly consisting of an encoder with a pre-modulation module and a GAN generator with $\mathcal{F}\&\mathcal{W}^+$ latent space. Within the encoder, the pre-modulation network leverages multi-scale structures to encode more discriminative semantic into style vectors. In order to bridge the gap between GAN inversion and image inpainting, $\mathcal{F}\&\mathcal{W}^+$ latent space is proposed to eliminate glaring color discrepancy and semantic inconsistency. To reconstruct faithful and photorealistic images, a simple yet effective Soft-update Mean Latent module is designed to capture more diverse in-domain patterns that synthesize high-fidelity textures for large corruptions. Comprehensive experiments on four challenging dataset, including Places2, CelebA-HQ, MetFaces, and Scenery, demonstrate that our InvertFill outperforms the advanced approaches qualitatively and quantitatively and supports the completion of out-of-domain images well. The code is available at https://github.com/yeates/InvertFill.

Keywords: Image inpainting · GAN inversion · $\mathcal{F}\&\mathcal{W}^+$ latent space

1 Introduction

Image inpainting is an ill-posed problem that requires to recover the missing or corrupted content based on incomplete images with masks. It has been

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19787-1_14.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
S. Avidan et al. (Eds.): ECCV 2022, LNCS 13676, pp. 242–258, 2022.
https://doi.org/10.1007/978-3-031-19787-1_14

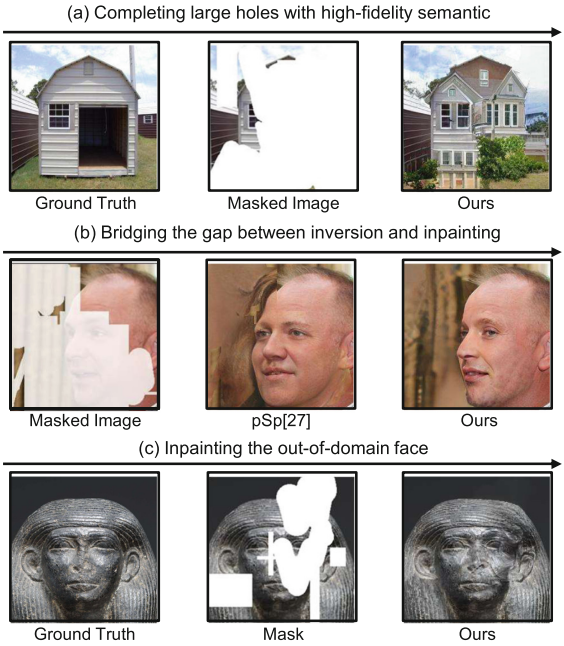

Fig. 1. Visual results of our contributions. Image (a) shows the high-fidelity inpainting results for large corruptions, image (b) exhibits the improvement of our method for the "gapping" issue over previous inversion-based inpainting method pSp [28], and image (c) demonstrates the semantically consistent results by our model for the out-of-domain masked image. *Best viewed in color for all figures throughout the paper.* (Color figure online)

widely adopted for manipulating photographs, such as corrupted image repairing, unwanted object removal, or object position modification [3,30,31].

The mainstream approaches [20,27] often employ an encoder-decoder architecture in UNet style [29] for image inpainting, and have demonstrated promising results in dealing with narrow holes or removing small objects. To apply to more complicated cases, later works have been focused on improving the performance with various discriminators [16,43,45], contextual attention mechanisms [16,45,47], and auxiliary information [8,21,26,46]. Nevertheless, limited by their model capacity, it remains challenging for these UNet-like methods to fill large corruptions with visually realistic patches.

Recently, generative adversarial network (GAN) models [11,13,14] have been verified to successfully produce high-resolution photorealistic images. In these models, GAN inversion [38,51] plays an important role. Specifically, when simply fed with stochastic vectors of latent space, GAN is not applicable to any image-to-image translation. To handle this problem, GAN inversion method uses a pre-trained GAN as prior, and encodes the given images into stochastic vectors that represent the target images, resulting to high-fidelity translation results. Inspired

by this, several approaches [2,6,28] have made great efforts to introduced GAN inversion for image inpainting. Despite excellent performance, existing methods may suffer from following issues:

- ***Distortion for extreme image inpainting.*** Due to large corruptions, current methods (*e.g.*, [8,16,47]) may become degenerated because these models are *not* able to effectively extract correlation from inadequate knowledge in extremely degraded images. Such correlation information is crucial in eliminating the ambiguity of large continuous holes, especially where far from the boundary.
- ***Inconsistency caused by hard constraint.*** Unlike in regular conditional translation (*e.g.*, super-resolution [6], face editing [50] and label-to-image [28]), image inpainting has a hard constraint that *the unmasked regions in the input and the output should be the same.* Current inversion-based algorithms [2,6,28], however, ignore this constraint, which results in color discrepancy and semantic inconsistency as displayed in Fig. 1(b) and may require additional post-processing such as image blending [2]. We call this problem "gapping" in the following sections.
- ***Robustness for out-of-domain inputs.*** In order to reconstruct faithful images, the key is to find an in-domain latent code that can align with the domain of a well-trained GAN model [50]. Unfortunately, the encoder fails to invert out-of-domain inputs to produce accurate results. For example, the pSp [28] is hard to tackle the corrupted images with contents or masks from unseen domains, which is harmful to the applicability of GAN inversion.

To solve the above issues, we introduce a novel InvertFill network for image inpainting. It follows the encoder-based inversion fashion architecture [28] that consists of an encoder and a GAN generator. We first develop a new latent space $\mathcal{F}\&\mathcal{W}^+$ (as explained later) that encodes the original images into style vector to enable the accessibility of the generator backbone to inputs, decreasing color discrepancy and semantic inconsistency. Besides, to make full use of the encoder, we present pre-modulation networks to amplify the reconstruction signals of the style vector based on the predicted multi-scale structures, further enhancing the discriminative semantic. Then, we propose a simple yet effective soft-update mean latent technique to sample a dynamic in-domain code for the generator. Compared to using a fixed code, our method is able to facilitate diverse downstream goals while reconstructing faithfully and photo-realistically, even in the task of unseen domain. To verify the superiority of our method, we conduct extensive experiments on four datasets, including CelebA-HQ [11], Places2 [49], MetFaces [12], and Scenery [41]. The results demonstrate that our method achieves favorable performance, especially for images with large corruptions. Furthermore, our approach can handle images and masks from unseen domains by optimizing a lightweight encoder without retraining the GAN generator on a large-scale dataset. Figure 1 shows several visual results of our approach.

The contributions of our work are summarized as three-fold: **(1)** We introduce a novel $\mathcal{F}\&\mathcal{W}^+$ latent space to resolve the problems of color discrepancy and

semantic inconsistency and thus bridge the gap between image inpainting and GAN inversion. **(2)** We propose (a) pre-modulation networks to encode more discriminative semantic from compact multi-scale structures and (b) soft-update mean latent to synthesize more semantically reasonable and visually realistic patches by leveraging diverse patterns. **(3)** Extensive experiments on CelebA-HQ [11], Places2 [49], MetFaces [12], and Scenery [41] show that the proposed approach outperforms current state-of-the-arts, evidencing its effectiveness.

2 Related Work

Image Inpainting. Image inpainting could be treated as a conditional translation task with hard constraint. The seminal learning-based work by Pathak *et al.* [27] integrates UNet [29] and GAN discriminator [5] for image inpainting, and subsequently derives many variants that effectively deal with narrow holes or remove small objects. More recently, several works have attempted to extending the idea in [27] to more complicated cases. Roughly speaking, these methods can be categorized into three types. The first one is to explicitly dispose of invalid signals at masked regions [20,23,43,44]. Among them, Liu *et al.* [20] attach heuristic mask update step to standard convolution and Yu *et al.* [43] formally replace the mask update process with a learnable convolution layer. The second type is called valuable signals shifting that is inspired by the traditional exemplar-based approach [3], which presently tends to model contextual attention to achieve [16,22,40,42,46]. In particular, RFR [16] applies multiple iterations at the bottleneck while sharing the attention scores to guide a patch-swap process. ProFill [47] iteratively performs inpainting based on the confidence map calculated by spatial attention. CRFill [46] yields a contextual reconstruction objective function that learns query-reference feature similarity. The third branch is to adopt auxiliary labels, which generate intermediate structures to assist with more accurate semantic [8,17,21,26]. In specific, EC [26] introduces canny edge to deliver finer inpainting structures. MEDFE [21] jointly learns to represent structures and textures and utilizes spatial and channel equalization to ensure consistency. CTSDG [8] couples texture and structure through parallel pathways and then fuses them by bidirectional gated layers. In addition to the above methods, there also exist other approaches. One notable example is Score-SDE [32] which proposes a scoring model that saves the gradient computation of energy-based models for efficient sampling.

Inpainting with GAN Inversion. StyleGAN [13] implicitly learns hierarchical latent styles $w \in \mathbb{R}^{1\times 512}$ instead of the initial stochastic vector z, which provides control over the style of outputs at coarse-to-fine levels of detail by style-modulation modules [10]. StyleGAN2 [14] further proposes weight demodulation, path length regularization, and generator redesign for improved image quality. They are adept in the generation without any given images, but requires specialized networks [24] or regularization [7,25] and paired training data. GAN inversion [51] is a common practice that takes advantage of the intrinsic statistics of well-trained large-scale GAN as prior for generic applications [1,50].

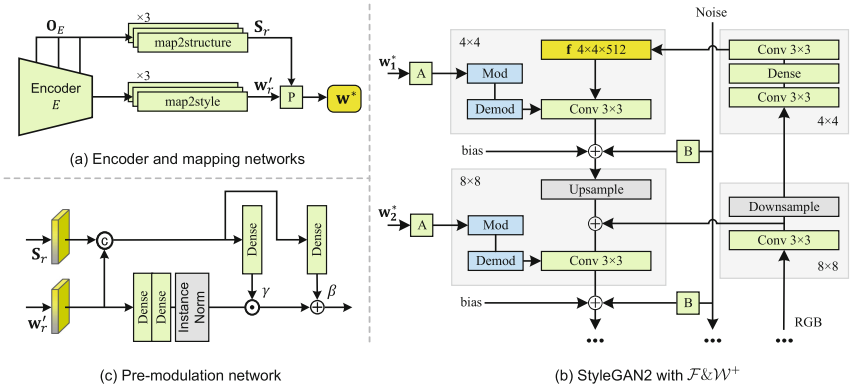

Fig. 2. The main components of InvertFill, including a feature pyramid-based encoder (image (a)), mapping network with pre-modulation network (image (c)) and a StyleGAN2 generator with the proposed $\mathcal{F}\&\mathcal{W}^+$ latent space (image (b)).

Existing GAN inversion approaches could be roughly divided as optimized-based [2,6,34,37] and encoder-based [28,39,50]. Among these methods, mGAN-prior [6] utilizes multiple latent codes and adaptive channel importance for faithful reconstruction and shows applications in different tasks including inpainting. pSp [28] synthesizes images with the mapping network to extract style vectors $w^+ \in \mathbb{R}^{18\times512}$ of latent space $\mathcal{W}^+$ [1] separately for corresponding 18 style-modulation layers of the StyleGAN. Nevertheless, these approaches ignore the "gapping" issue, resulting in color inconsistency and semantic misalignment.

Difference with Previous Studies. In this paper we focus on encoder-based GAN inversion to improve generation fidelity for image inpainting. The proposed InvertFill is related to but significantly different from previous studies. In specific, InvertFill is relevant to the methods in [28,39,50] where encoder-based architecture is adopted. However, differing from them, we introduce a new $\mathcal{F}\&\mathcal{W}^+$ latent space to explicitly handle the "gapping" issue which is ignored in previous algorithms. Our method also shares similar spirit with the works of [46,47] that adopt GAN for image inpainting. The difference is that these approaches may suffer from ambiguity when filling the large corruptions, while the proposed InvertFill exploits the priors of a large-scale generator and can achieve image inpainting with high-fidelity semantic.

3 The Proposed Method

Given an original image $\mathbf{I}$ and its corrupted image $\mathbf{I}_m = \mathbf{I} \odot (1 - \mathbf{M})$, where $\mathbf{M}$ is a binary mask and $\odot$ denotes element-wise product. The value of pixels in masked region $\mathbf{M}$ equal to 1 indicates invisible. We aim to produce a visually realistic reconstructed image $\mathbf{O}$ with the input of corrupted images $\mathbf{I}_m$.

3.1 $\mathcal{F}\&\mathcal{W}^+$ Latent Space

Our architecture mainly consists of three components: (i) A feature pyramid-based [19] encoder E that extracts input images and provides hierarchical reconstructed RGB images, (ii) the mapping networks with pre-modulation module, and (iii) a StyleGAN2 generator that takes in the style vectors as well as the input image $\mathbf{I}_m$ to generate a image. The details of InvertFill are shown in Fig. 2.

Specifically, we attach three RGB heads to the encoder E for generating reconstructed RGB images $\mathbf{O}_E^r = \{\mathbf{O}_E^1, \mathbf{O}_E^2, \mathbf{O}_E^3\}$ in correspondence to three different scale. We follow the *map2style* [28] for the mapping network and cut down the network number from 18 to 3, each of which corresponds to the disentanglement level of image representation (i.e., coarse, middle and fine [13,28]). Three *map2style* networks encode the output feature map of the encoder into the intermediate code $\mathbf{w}' \in \mathbb{R}^{3\times 512}$. Similarly, we replicate *map2style* as *map2structure* to project reconstructed RGB images $\mathbf{O}_E^r$ gradually into structure vector $\mathbf{S}_r = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$.

Before executing the style modulation in the generator, we perform L pre-modulation networks to project the semantic structure $\mathbf{S}_r$ into the style vector $\mathbf{w}^*$ in latent space $\mathcal{F}\&\mathcal{W}^+$, i.e., $\mathbf{w}^* = E(\mathbf{I}_m), \mathbf{w}^* \in \mathbb{R}^{L\times 512}$. $L = \log_2(s)\cdot 2 - 2$ denotes number of style-modulation layers of StyleGAN2 generator, and is adjusted by the image resolution s on the generator side. As Fig. 2(c) demonstrates, we adopt Instance Normalization (IN) [33] to regularize the $\mathbf{w}'$ latent code, then carry out denormalization according to multi-scale structure vector $\mathbf{S}_r$,

$$\mathbf{w}_l^* = \gamma \odot \mathrm{IN}\left(\mathbf{w}_r'\right) + \beta, \tag{1}$$

where $l \in \{1, 2, \ldots, L\}$ denotes the index of style vectors, $r \in [1, 3]$ indicates three vectors $\mathbf{w}'$ correspond to level of coarse to fine, (γ, β) is a pair of the affine transformation parameters learned by networks shown in Fig. 2(c). Different than previous methods in only using intermediate latent code from a network, the proposed pre-modulation module is a lightweight network and novel in applying more discriminative multi-scale features to help latent code perceive uncorrupted prior and better guide image generation.

The GAN is initially fed with a stochastic vector $z \in \mathcal{Z}$, and previous works [1,6,28,50] invert the source images into the intermediate latent space $\mathcal{W}$ or $\mathcal{W}^+$, which is a less entangled representation than latent space $\mathcal{Z}$. The style vectors $w \in \mathcal{W}$ or $w^+ \in \mathcal{W}^+$ are sent to the style-modulation layers of pre-trained StyleGAN2 to synthesize target images. These approaches can be formulated mathematically as follows,

$$\mathbf{O}_G = G(E(\mathbf{I}_m)), E(\mathbf{I}_m) \sim W^+, \tag{2}$$

where $E(\cdot)$ and $G(\cdot)$ represent the encoder that maps source images into latent space and the pre-trained GAN generator, respectively.

Nevertheless, the above formulation in Eq. (2) may encounter the "gapping" issue in image translation tasks with hard constraint, *e.g.*, image inpainting. The hard constraint requires that parts of the source and recovered image remain the

same. We formally defined the hard constraint in image inpainting as $\mathbf{I} \odot (1 - \mathbf{M}) \equiv \mathbf{O} \odot (1 - \mathbf{M})$. Intuitively, we argue that the "gapping" issue is caused by that the GAN model cannot directly access pixels of the input image but the intermediate latent code. To avoid the semantic inconsistency and color discrepancy caused by this problem, we utilize the corrupted image $\mathbf{I}_m$ as one of the inputs to assist with the GAN generator inspired by skip connection of U-Net [29]. In detail, $\mathbf{I}_m$ is fed into the RGB branch as shown in Fig. 2(b), the feature map between RGB branch and the generator are connected by element-wise addition. Hence, the previous formulation in Eq. (2) is updated as:

$$\mathbf{O}_G = G(E(\mathbf{I}_m), \mathbf{I}_m), E(\mathbf{I}_m) \sim \mathcal{F}\&\mathcal{W}^+ \tag{3}$$

.

3.2 Soft-update Mean Latent

Pixels closer to the mask boundary are more accessible to inpainting, but conversely the model is hard to predict specific content missing. We find that the encoder learns a trick to averaging textures to reconstruct the region away from unmasked region. It causes blurring or mosaic in some areas of the output image, mainly located away from the mask borders, as shown in Fig. 7. Drawing inspiration from L2 regularization and motivated by the intuition that fitting diverse domains works better than fitting a preset static domain, a feasible solution is to make style code $\mathbf{w}^*$ be bounded by the mean latent code of pre-trained GAN.

The mean latent code is obtained from abundant random samples that restrict the encoder outputs to the average style hence lossy the diversity of output distribution of encoder. In addition, it introduces additional hyperparameters and a static mean latent code that requires loading when training the model.

We adopt dynamic mean latent code instead of static one by stochastically fluctuating the mean latent code while training. Further, we smooth the effect of fluctuating variance for convergence inspired by a reinforcement learning [18]. For initialization, target mean latent code $\overline{\mathbf{w}}_t$ and online mean latent code $\overline{\mathbf{w}}_o$ are sampled. $\overline{\mathbf{w}}_o$ is used in image generation instead of $\overline{\mathbf{w}}_t$, which is fixed until $\overline{\mathbf{w}}_o = \overline{\mathbf{w}}_t$ and then resampled. Between two successive sampled mean latent codes, $\overline{\mathbf{w}}_o$ is updated by $\overline{\mathbf{w}}_t \leftarrow \tau\overline{\mathbf{w}}_o + (1-\tau)$ per iteration during training, where τ denotes updating factor and $\overline{\mathbf{w}}_t$ for soft updating target mean latent code. The soft-update mean latent degraded to static mean latent [28] when the parameter τ of soft-update mean latent approaching zero.

3.3 Optimization

Following prior work in inpainting [16,20], our architecture is supervised by regular inpainting loss $\mathcal{L}_{\text{ipt}}$, which consists of the pixel-wise Euclidean norm of valid and hole regions, the perceptual loss perc, the style loss style, and the total variation loss tv:

$$\mathcal{L}_{\text{ipt}} = \mathcal{L}_{\text{valid}} + \mathcal{L}_{\text{hole}} + \mathcal{L}_{\text{perc}} + \mathcal{L}_{\text{style}} + \mathcal{L}_{\text{tv}}, \tag{4}$$

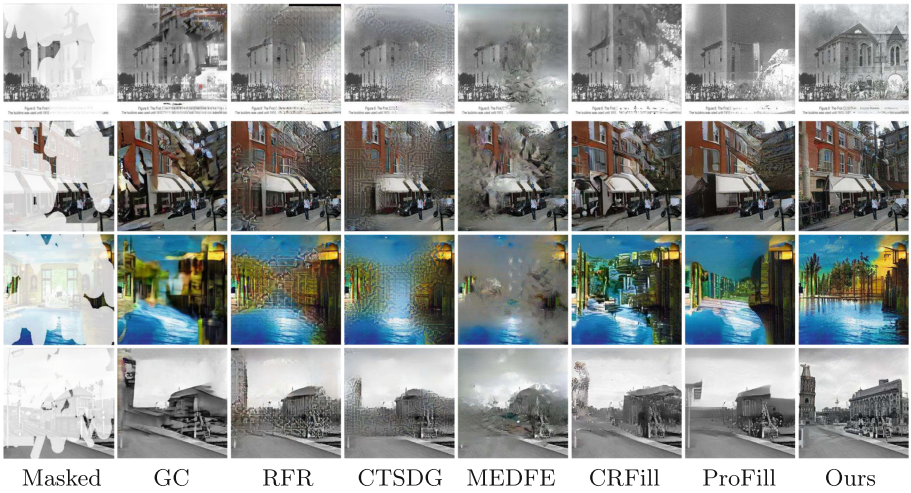

Fig. 3. Qualitative results on Places2 dataset.

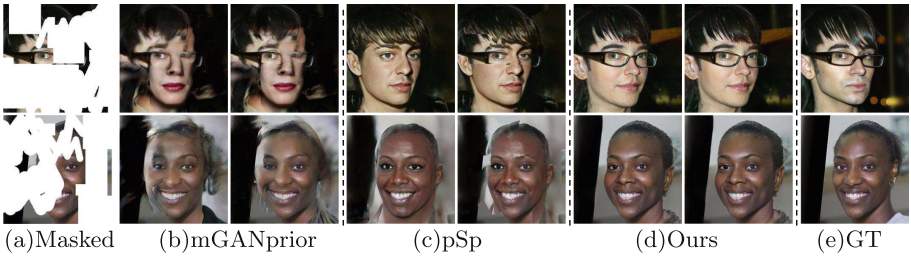

Fig. 4. Qualitative results on CelebA-HQ dataset. Two columns of (b-d) show the original model output and composition output, from left to right, respectively. The output of the GAN-inversion-based method (pSp [28] and mGANprior [6]) is inconsistency at the edge of the mask. Zoom-in to see the details.

where all above distance are calculated between $\mathbf{I}$ and $\mathbf{O}_G$. $\mathcal{L}_{\text{valid}}$ and $\mathcal{L}_{\text{hole}}$ are ℓ_1 norm on the known and masked region respectively. The perceptual loss $\mathcal{L}_{\text{perc}}$ and the style loss $\mathcal{L}_{\text{style}}$ are based on a pre-trained VGG-16 network. More details can be found in [16].

To directly optimize our encoder, the multi-scale reconstruction loss $\mathcal{L}_{\text{msr}}$ is utilized to penalize the deviation of $\mathbf{O}_E^r$ at each scale:

$$\mathcal{L}_{\text{msr}} = \sum_{r=1}^{3} (\mathcal{L}_{\text{perc}}^r + \mathcal{L}_{\text{style}}^r + \mathcal{L}_{\text{rec}}^r), \tag{5}$$

where $\mathcal{L}_{\text{rec}}$ is represented as mean-squared loss between $\mathbf{I}$ and $\mathbf{O}_E$. The multi-scale reconstruction loss $\mathcal{L}_{\text{msr}}$ contains three different losses including perceptual ($\mathcal{L}_{\text{perc}}^r$) [4], style ($\mathcal{L}_{\text{style}}^r$) [20] and mean-square ($\mathcal{L}_{\text{rec}}^r$) losses. The role of $\mathcal{L}_{\text{msr}}$ is

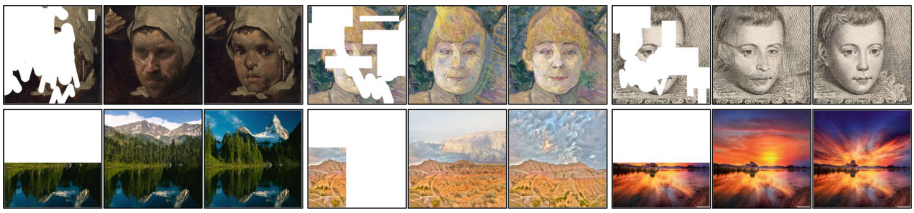

Fig. 5. The visual effect of our method for processing input images from unseen domain. The 1st row shows inpainting results of Metfaces, and the 2nd row shows outpainting results of Scenery. Each instance of results is laid out as the masked image, the model output, and the original image.

to supervise the generated image from decoder and make final generation close to the original image.

The soft-update mean latent is utilized to prevent the encoder from falling into the trick way. We adopt the following fidelity loss $\mathcal{L}_{\text{fid}}$ for improving the quality and diversity of output images:

$$\mathcal{L}_{\text{fid}} = \|\mathbf{w}^* - \overline{\mathbf{w}}\|_2, \ (\mathbf{w}^*, \overline{\mathbf{w}}) \in \mathbb{R}^{L \times 512}. \tag{6}$$

The fidelity loss $\mathcal{L}_{\text{fid}}$ is designed as a mean squared loss of style vectors $\mathbf{w}^*$ and mean latent code $\bar{\mathbf{w}}$. Its role is to improve the quality and diversity of the output images.

Overall, the loss of our networks is defined as the weighted sum of the inpainting loss, the multi-scale reconstruction loss, and the fidelity loss.

$$\mathcal{L} = \mathcal{L}_{\text{ipt}} + \lambda_{\text{msr}}\mathcal{L}_{\text{msr}} + \lambda_{\text{fid}}\mathcal{L}_{\text{fid}}, \tag{7}$$

where λ_{msr} and λ_{fid} are the balancing factors for the multi-scale reconstruction loss and the fidelity loss, respectively.

4 Experiments

We perform extensive validating experiments aiming to answer the following research questions:

- **RQ1:** How does our approach perform, compared with existing methods, especially the fidelity when the input is large-scale masked images.
- **RQ2:** Can our approach resolve the "gapping" issue?
- **RQ3:** Can our approach handle input from unseen domain by reusing the well-trained generator while only retraining a lightweight encoder?
- **RQ4:** How do different components (e.g., soft-update mean latent, pre-modulation) affect our approach?

Table 1. Quantitative comparison with the mainstream inpainting approaches on Places2 and CelebA-HQ datasets. *Hard*, *Extreme*, *All* masks denote the mask with coverage ratio of 50%–60%, 70%–90%, and 10%–90%, respectively. ↑ Higher is better, and ↓ lower is better. Best and second best results are **highlighted**.

			GC	RFR	MEDFE	ProFill	CTSDG	CRFill	Ours
Places2	*hard*	SSIM↑	0.624	0.645	0.598	**0.664**	0.651	0.629	0.641
		FID↓	22.05	27.77	44.38	21.49	35.77	22.46	**12.44**
		LPIPS↓	0.246	0.235	0.294	0.240	0.272	0.250	**0.232**
	extreme	SSIM↑	0.363	0.382	0.323	**0.409**	0.393	0.360	0.366
		FID↓	51.35	71.19	111.85	46.44	95.50	51.26	**21.08**
		LPIPS↓	0.407	0.395	0.495	0.402	0.438	0.413	**0.386**
	all	SSIM↑	0.734	0.750	0.714	**0.764**	0.755	0.738	0.761
		FID↓	14.19	16.26	26.15	13.81	21.36	14.44	**9.29**
		LPIPS↓	0.178	0.170	0.217	0.173	0.199	0.182	**0.155**
CelebA-HQ	*hard*	SSIM↑	0.790	**0.825**	0.781	–	0.818	0.810	0.812
		FID↓	17.38	9.98	21.97	–	15.13	13.78	**9.89**
		LPIPS↓	0.170	0.128	0.192	–	0.151	0.139	**0.121**
	extreme	SSIM↑	0.589	0.641	0.552	–	0.616	0.639	**0.652**
		FID↓	41.70	22.07	55.52	–	33.89	30.19	**13.21**
		LPIPS↓	0.297	0.241	0.359	–	0.281	0.275	**0.214**
	all	SSIM↑	0.852	**0.878**	0.846	–	0.875	0.859	0.867
		FID↓	11.78	7.96	15.52	–	10.32	11.94	**7.71**
		LPIPS↓	0.128	0.092	0.142	–	0.110	0.114	**0.089**

4.1 Experimental Settings

Datasets. Experiments for RQ1, RQ2 and RQ4 are conducted on two datasets, Places2 [49] and CelebA-HQ [11]. CelebA-HQ contains 30,000 high-resolution celebrity faces, and we follow [42,43] to split this dataset for training and testing. Places2 contains real-world photos, including more significant objects, such as streets, cars, houses, which is better suited for verifying models on large-scale masks than CelebA-HQ. Based on the official *train*/*val*/*test* split, we train the model on *train* plus *test* about 200,000 images, evaluate the model on first 5,000 images of *val*. With regard to RQ3, we utilize two datasets Scenery [41] and MetFaces [12]. Scenery dataset is a common benchmark for recent image outpainting tasks and contains 6,040 landscape photographs. We follow [41] to use about 5,000 images as training set and the remaining 1,000 images as test set. MetFaces consists of 1,336 human faces extracted from works of art, and we randomly select 1,000 images as training set and other images as test set. Our model and all baselines adopt the same training and test strategies to ensure experimental fairness.

Evaluation Metrics. We use three metrics following prior works to measure the quality and fidelity of inpainting results. SSIM [35] modeling image distortion by structure, luminance, and contrast, is a pixel-level objective metric similar to PSNR, and their drawbacks cause inconsistent evaluation results with the human eye. Despite that, they are classical metrics for image evaluation, one of which SSIM we selected for quantitative comparison. FID [9] is a deep metric

Fig. 6. Comparison with pSp [28] and pSp+Blend [36] that post-processing by image blending. The 1st row shows the color discrepancy that image blending is sufficient to resolve satisfactorily. The 2nd row shows that the semantic inconsistency is still reserved, except for our method. (Color figure online)

and closer to human perception. It measures the distribution distance with a pre-trained inception model, which better captures distortions. LPIPS [48] is another learned perceptual metric and commonly used to score the intra-conditioning diversity of models output. Following previous works [16,26,43], We calculate these quantitative metrics on original images $\mathbf{I}$ and composition images $\mathbf{I} \odot (1 - \mathbf{M}) + \mathbf{O}_G \odot \mathbf{M}$.

Baselines. We carefully select baseline methods mainly from two perspectives: UNet style methods and Inversion style methods to demonstrate our approach's characteristics and superiority. First, for the sake of validating the ability of InvertFill in filling images under large-scale masks, we compare it with the previous approaches including EC [26], GC [42], RFR [16], MEDFE [21], ProFill [47], CTSDG [8] and CRFill [46]. Second, we compare with the latest GAN inversion-based inpainting methods mGANprior [6] and pSp [28].

4.2 Implementation Details

We utilize eight A100 GPUs for pre-training the GAN generator, and one TITAN RTX GPU for optimizing the encoder and other experiments. Following [16], we scale the image size of all datasets to 256×256 as the input. In the light of the mask coverage, we classify the test masks into three difficulty levels: *Hard*/*Extreme*/*All*, indicates the mask with coverage ratio of 50%–60%, 70%–90%, 10%–90%, respectively. During testing, for a fair comparison, we use the same image-mask pair for all approaches. More details of implementation are shown in the supplementary.

4.3 Result Analysis

RQ1. We reproduce all the above baselines by utilizing their official implementations. Concerning Places2 dataset, we utilize the pre-trained weights officially released by the baselines. On CelebA-HQ dataset, EC [26], GC [43], mGANprior [6] offer pre-trained weights, we thereby carefully retrain other baselines through the official source codes. Because ProFill only offers Web API on Places2, we use placeholder '–' in Table 1 for ProFill on CelebA-HQ.

From Table 1, our method achieves the best or comparable performance among advanced inpainting approaches. In terms of the FID metric, our method

Table 2. Comparison with previous GAN inversion-based and diffusion-based approaches on CelebA-HQ dataset.

	FID↓	LPIPS↓	SSIM↑
Score-SDE [32]	24.76	0.337	0.428
mGANprior [6]	29.57	0.273	0.608
pSp [28]	25.61	0.248	0.594
pSp + Blend [36]	21.96	0.240	0.602
Ours	**13.21**	**0.214**	**0.652**

Table 3. Comparison with previous outpainting approaches and inpainting baselines on Scenery dataset.

	FID↓	LPIPS↓	SSIM↑
RFR [16]	138.31	0.455	0.376
pSp [28]	49.62	0.379	0.392
Boundless [15]	45.05	0.368	0.413
NS-outpaint [41]	38.95	0.342	0.410
Ours	**20.90**	**0.294**	**0.439**

Table 4. Comparison with previous inpainting methods on Metfaces. In this experimental setting, the model/generator is only trained on CelebA-HQ.

Method	Easy			Extreme		
	SSIM↑	FID↓	LPIPS↓	SSIM↑	FID↓	LPIPS↓
RFR	0.93	18.89	0.069	0.52	58.24	0.315
CRFill	0.95	13.67	0.042	0.54	50.93	0.278
pSp	0.95	14.91	0.040	0.49	65.04	0.341
Ours	**0.97**	**8.64**	**0.033**	**0.60**	**38.85**	**0.227**

at most produces a notable margin of 54.60% and 40.14% on Places2 and CelebA-HQ datasets, respectively. And our method also outperforms the second-best approach 11.2% and 10.4% improvements on another perceptual metric LPIPS, which validates the superiority of our design.

Figure 3 and 4 provide several visual inpainting results on Places2 and CelebA-HQ datasets. Figure 3 reveals that the prior works still struggle to generate refined texture if the input image with large corruptions, while our approach has been able to create semantically rich objects such as windows, towers, and woods. In Fig. 4, mGANprior [6] progressively erases the color discrepancy rely on optimized-based inversion but is unable to bypass semantic inconsistency. The encoder-based inversion method pSp [28] could synthesize realistic pixels for corrupted regions based on the well-trained model, though it still has not resolved the "gapping" issue. The results indicate that our method produces consistent output while generating high-fidelity texture compared to existing methods.

RQ2. The "gapping" causes color discrepancy and semantic inconsistency, and we are counting on image post-processing to tackle this issue at the beginning of this study. Specifically, we adopt image blending [36], which is effective in eliminating the color discrepancy but helpful in remedying semantic inconsistency.

To further demonstrate the superiority of our method, we construct the pSp+Blend variant that introduces an image blending [36] method after generating output images. In Fig. 6, the first row shows the distinct gap at the stitching boundary in pSp output, and pSp+Blend fixes this color discrepancy problem. Even so, the second row shows pSp+Blend unable to assist with

semantic inconsistency problem given the glasses are still incomplete. Compared with the vanilla pSp and pSp+Blend, output images of our approach no longer suffer from color discrepancy or semantic inconsistency.

We conduct a comparison experiment on CelebA-HQ dataset with the *Extreme* level masks. As demonstrated in Table 2, our method performs better than a recent diffusion-based approach Score-SDE [32] w.r.t to FID, LPIPS and SSIM metrics. The results in Table 2 also show that our method performs best among the existing inversion-based inpainting approaches after resolving the "gapping" issue. Notably, our method does not require any image post-processing.

Fig. 7. The importance of soft-update mean latent.

RQ3. Concerning validating that our approach can reuse the pre-trained GAN generator as priors to tackle image from unseen domain, we conduct two extended tests that introduced images or masks from unseen domains and only required optimizing the lightweight encoder. The first is archaic photograph inpainting, and we use MetFaces [12] for optimizing the encoder, and remain the pre-trained weights of GAN generator of CelebA-HQ dataset. For the second one, we perform our approach with outpainting masks [15] on Scenery dataset. Similarly, the generator did not retrain on the Scenery dataset rather than remaining the weights for Places2.

The 1st row of Fig. 5 shows the inpainting results of archaic photograph inpainting. It demonstrates that our method enables the generator to synthesize semantically consistent style and patches, even in an unseen domain. From the 2nd row of Fig. 5, the outpainting results on the Scenery dataset show our approach still can synthesize realistic texture and significant objects, *e.g.*trees, mountains. To ensure the masks are unseen for the GAN generator, we only use the outpainting masks to train the encoder, not the GAN generator.

Furthermore, we quantitatively compare mainstream outpainting approaches as well as adopt RFR [16] and pSp [28] as additional baselines. As shown in Table 3, our model considerably outperforms the best outpainting baselines [15, 41] with respect to FID, LPIPS, and SSIM. Similarly, we conduct experiments compared with inpainting baselines on Metfaces, as show in Table 4. In summary, the results indicate that our proposed method is robust and extends to other tasks with out-of-domain inputs.

Due to limited space, please kindly refer to the supplementary material for more results.

Table 5. Ablation study comparison on Places2 dataset under *Extreme* mask setting.

$\mathcal{F}\&\mathcal{W}^+$	SML	PM	$\mathcal{W}^+$	FID	LPIPS	SSIM	PSNR
✓				35.37	0.395	0.357	13.85
✓	✓			24.73	0.389	0.358	13.99
✓	✓	✓		21.08	0.386	0.366	14.62
	✓	✓	✓	42.85	0.392	0.361	14.25

Fig. 8. Illustration of two failure cases of the proposed method.

4.4 Ablation Study (RQ4)

The ablation experiments are carried out on the Places2 dataset under the *Extreme* mask setting. In Table 5, we construct three variants to verify the contribution of proposed modules, in which PM and SML denote pre-modulation and soft-update mean latent. By learning from these modules, our method considerably outperforms the most naive variant w.r.t FID, LPIPS, SSIM, and PSNR.

The soft-update mean latent is motivated by the intuition that fitting diverse domains works better than fitting a preset static domain, especially when the training dataset contains various scenarios such as street and landscape. As shown in Fig. 7, when we use SML code that dynamically fluctuates during training, the masked region far away from the mask border tends to be reconstructed by explicitly learned semantics instead of repetitive patterns. Notably, 'w/o SML' represents using regular static mean latent code.

4.5 Failure Cases and Discussion

Figure 8 shows two failure cases. Even if the model can recognize the corrupted objects (our method tends to recover the human face in the left case of Fig. 8), it mistakenly locates them and produces severe artifacts. When lacking sufficient prior knowledge, our method fails to reconstruct details. This demonstrates that these situations are challenging for image inpainting and need further study.

5 Conclusion

In this paper, we propose an encoder-based GAN inversion method InvertFill for image inpainting. The encoder projects corrupted images into a latent space

$\mathcal{F}\&\mathcal{W}^+$ with pre-modulation for learning more discriminative representation. The novel latent space $\mathcal{F}\&\mathcal{W}^+$ resolves the "gapping" issue when applied to GAN inversion in image inpainting. In addition, the soft-update mean latent dynamically samples diverse in-domain patterns, leading to more realistic textures. Extensive quantitative and qualitative comparisons demonstrate the superiority of our model over previous approaches and can cheaply support the semantically consistent completion of images or masks from unseen domains.

Acknowledgment. This work was supported by the Key Research Program of Frontier Sciences, CAS, Grant No. ZDBS-LY-JSC038. Libo Zhang was supported by the CAAI-Huawei MindSpore Open Fund and Youth Innovation Promotion Association, CAS (2020111). Heng Fan and his employer received no financial support for the research, authorship, and/or publication of this article.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2styleGAN: how to embed images into the styleGAN latent space? In: ICCV, pp. 4431–4440 (2019)
2. Cheng, Y., Lin, C.H., Lee, H., Ren, J., Tulyakov, S., Yang, M.: In&out: diverse image outpainting via GAN inversion. CoRR abs/2104.00675 (2021)
3. Criminisi, A., Pérez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: CVPR, pp. 721–728 (2003)
4. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. CoRR abs/1508.06576 (2015)
5. Goodfellow, I.J., et al.: Generative adversarial nets. In: NeurIPS, pp. 2672–2680 (2014)
6. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code GAN prior. In: CVPR, pp. 3009–3018 (2020)
7. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: NeurIPS, pp. 5767–5777 (2017)
8. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: ICCV, pp. 14134–14143 (2021)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: NeurIPS, pp. 6626–6637 (2017)
10. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV, pp. 1510–1519 (2017)
11. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR (2018)
12. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: NeurIPS (2020)
13. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR, pp. 4401–4410 (2019)
14. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of styleGAN. In: CVPR, pp. 8107–8116 (2020)
15. Krishnan, D., et al.: Boundless: generative adversarial networks for image extension. In: ICCV, pp. 10520–10529 (2019)
16. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: CVPR, pp. 7757–7765 (2020)

17. Liao, L., Xiao, J., Wang, Z., Lin, C., Satoh, S.: Image inpainting guided by coherence priors of semantics and textures. In: CVPR, pp. 6539–6548 (2021)
18. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. In: ICLR (2016)
19. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, pp. 936–944 (2017)
20. Liu, G., Reda, F.A., Shih, K.J., Wang, T.-C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 89–105. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_6
21. Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 725–741. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_43
22. Liu, H., Jiang, B., Xiao, Y., Yang, C.: Coherent semantic attention for image inpainting. In: ICCV, pp. 4169–4178 (2019)
23. Ma, Y., Liu, X., Bai, S., Wang, L., He, D., Liu, A.: Coarse-to-fine image inpainting via region-wise convolutions and non-local correlation. In: IJCAI, pp. 3123–3129 (2019)
24. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: ICCV, pp. 2813–2821 (2017)
25. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: ICLR (2018)
26. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: EdgeConnect: structure guided image inpainting using edge prediction. In: ICCVW, pp. 3265–3274 (2019)
27. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: CVPR, pp. 2536–2544 (2016)
28. Richardson, E., et al.: Encoding in style: a styleGAN encoder for image-to-image translation. In: CVPR, pp. 2287–2296 (2021)
29. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
30. Shetty, R., Fritz, M., Schiele, B.: Adversarial scene editing: automatic object removal from weak supervision. In: NeurIPS, pp. 7717–7727 (2018)
31. Song, L., Cao, J., Song, L., Hu, Y., He, R.: Geometry-aware face completion and editing. In: AAAI, pp. 2506–2513 (2019)
32. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2021)
33. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: the missing ingredient for fast stylization. CoRR abs/1607.08022 (2016)
34. Wang, H.P., Yu, N., Fritz, M.: Hijack-GAN: unintended-use of pretrained, black-box GANs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7872–7881, June 2021
35. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**, 600–612 (2004)
36. Wu, H., Zheng, S., Zhang, J., Huang, K.: GP-GAN: towards realistic high-resolution image blending. In: MM, pp. 2487–2495 (2019)

37. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: disentangled controls for styleGAN image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12863–12872, June 2021
38. Xia, W., Zhang, Y., Yang, Y., Xue, J., Zhou, B., Yang, M.: GAN inversion: a survey. CoRR abs/2101.05278 (2021)
39. Xu, Y., Shen, Y., Zhu, J., Yang, C., Zhou, B.: Generative hierarchical features from synthesizing images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4432–4442, June 2021
40. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-Net: image inpainting via deep feature rearrangement. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_1
41. Yang, Z., Dong, J., Liu, P., Yang, Y., Yan, S.: Very long natural scenery image prediction by outpainting. In: ICCV, pp. 10560–10569 (2019)
42. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: CVPR, pp. 5505–5514 (2018)
43. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV, pp. 4470–4479 (2019)
44. Yu, T., et al.: Region normalization for image inpainting. In: AAAI, pp. 12733–12740 (2020)
45. Zeng, Y., Fu, J., Chao, H., Guo, B.: Aggregated contextual transformations for high-resolution image inpainting. CoRR abs/2104.01431 (2021)
46. Zeng, Y., Lin, Z., Lu, H., Patel, V.M.: CR-Fill: generative image inpainting with auxiliary contextual reconstruction. In: ICCV, pp. 14164–14173, October 2021
47. Zeng, Yu., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 1–17. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_1
48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR, pp. 586–595 (2018)
49. Zhou, B., Lapedriza, À., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. TPAMI **40**, 1452–1464 (2018)
50. Zhu, J., Shen, Y., Zhao, D., Zhou, B.: In-domain GAN inversion for real image editing. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12362, pp. 592–608. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58520-4_35
51. Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 597–613. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_36