Exploiting Deep Generative Prior for Versatile Image Restoration and Manipulation

Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, *Senior Member, IEEE*, and Ping Luo

Abstract—Learning a good image prior is a long-term goal for image restoration and manipulation. While existing methods like deep image prior (DIP) capture low-level image statistics, there are still gaps toward an image prior that captures rich image semantics including color, spatial coherence, textures, and high-level concepts. This work presents an effective way to exploit the image prior captured by a generative adversarial network (GAN) trained on large-scale natural images. As shown in Fig. 1, the deep generative prior (DGP) provides compelling results to restore missing semantics, e.g., color, patch, resolution, of various degraded images. It also enables diverse image manipulation including random jittering, image morphing, and category transfer. Such highly flexible restoration and manipulation are made possible through relaxing the assumption of existing GAN inversion methods, which tend to fix the generator. Notably, we allow the generator to be fine-tuned on-the-fly in a progressive manner regularized by feature distance obtained by the discriminator in GAN. We show that these easy-to-implement and practical changes help preserve the reconstruction to remain in the manifold of nature images, and thus lead to more precise and faithful reconstruction for real images. Code is available at https://github.com/XingangPan/deep-generative-prior.

Index Terms—Image prior, generative adversarial networks, image processing

1 Introduction

Learning image prior models is essential to solve various tasks of image restoration and manipulation, such as *image colorization* [1], [2], *image inpainting* [3], *super-resolution* [4], [5], and *adversarial defense* [6]. In the past decades, many image priors [7], [8], [9], [10], [11] have been proposed to capture certain statistics of natural images. Despite their successes, these priors often serve a dedicated purpose. For instance, markov random field [7], [8], [9] is often used to model the correlation among neighboring pixels, while dark channel prior [10] and total variation [11] are developed for dehazing and denoising respectively.

There is a surge of interest to seek for more general priors that capture richer statistics of images through deep learning models. For instance, the seminal work on deep image prior (DIP) [12] showed that the structure of a randomly initialized Convolutional Neural Network (CNN) implicitly captures texture-level image prior, thus can be used in image restoration by fine-tuning it to reconstruct a corrupted image. SinGAN [13] further shows that a randomly-initialized generative adversarial network (GAN) model is able to capture rich patch statistics after being trained from a single image. These priors have shown impressive results on some low-level image restoration and manipulation tasks like super-resolution and harmonizing. In both the representative works, the CNN and GAN are trained from a single image of interest from scratch.

In this study, we are interested to go one step further, examining how we could leverage a GAN [14] trained on large-scale natural images for richer priors beyond a single image. GAN is a good approximator for natural image manifold. By learning from large image datasets, it captures rich knowledge on natural images including color, spatial coherence, textures, and high-level concepts, which are useful for broader image restoration and manipulation effects. Specifically, we take a collapsed image (e.g., gray-scale image) as a partial observation of the original natural image, and reconstruct it in the observation space (e.g., gray-scale space) with the GAN, the image prior of the GAN would tend to restore the missing semantics (e.g., color) in a faithful way to match natural images. Despite its enormous potentials, it remains a challenging task to exploit a GAN as a prior for general image restoration and manipulation. The key challenge lies in the needs in coping with arbitrary images from different tasks with distinctly different natures. The reconstruction also needs to produce sharp and faithful images obeying the natural image manifold.

A plausible option for our problem is GAN inversion [15], [16], [17], [18]. Existing GAN inversion methods typically reconstruct a target image by optimizing over the latent code, i.e., $\mathbf{z}^* = \arg\min_{\mathbf{z}\in\mathbb{R}^d} \mathcal{L}(\mathbf{x}, G(\mathbf{z};\boldsymbol{\theta}))$, where $\mathbf{x}$ is the target

- *Xingang Pan, Xiaohang Zhan, and Dahua Lin are with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong. E-mail: {px117, dhlin}@ie.cuhk.edu.hk, xiaohangzhan@outlook.com.*
- *Bo Dai and Chen Change Loy are with S-Lab, Nanyang Technology University, Singapore 639798, Singapore. E-mail: {bo.dai, ccloy}@ntu.edu.sg.*
- *Ping Luo is with the Department of Computer Science, The University of Hong Kong, Hong Kong, Hong Kong. E-mail: pluo@cs.hku.hk.*

Manuscript received 13 Feb. 2021; revised 24 July 2021; accepted 21 Sept. 2021. Date of publication 24 Sept. 2021; date of current version 3 Oct. 2022.
*This work was supported in part by A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant, in part by the HK General Research Fund under Grant 27208720, and in part by the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).*
(Corresponding authors: Xingang Pan and Ping Luo.)
Recommended for acceptance by W. Zuo.
Digital Object Identifier no. 10.1109/TPAMI.2021.3115428

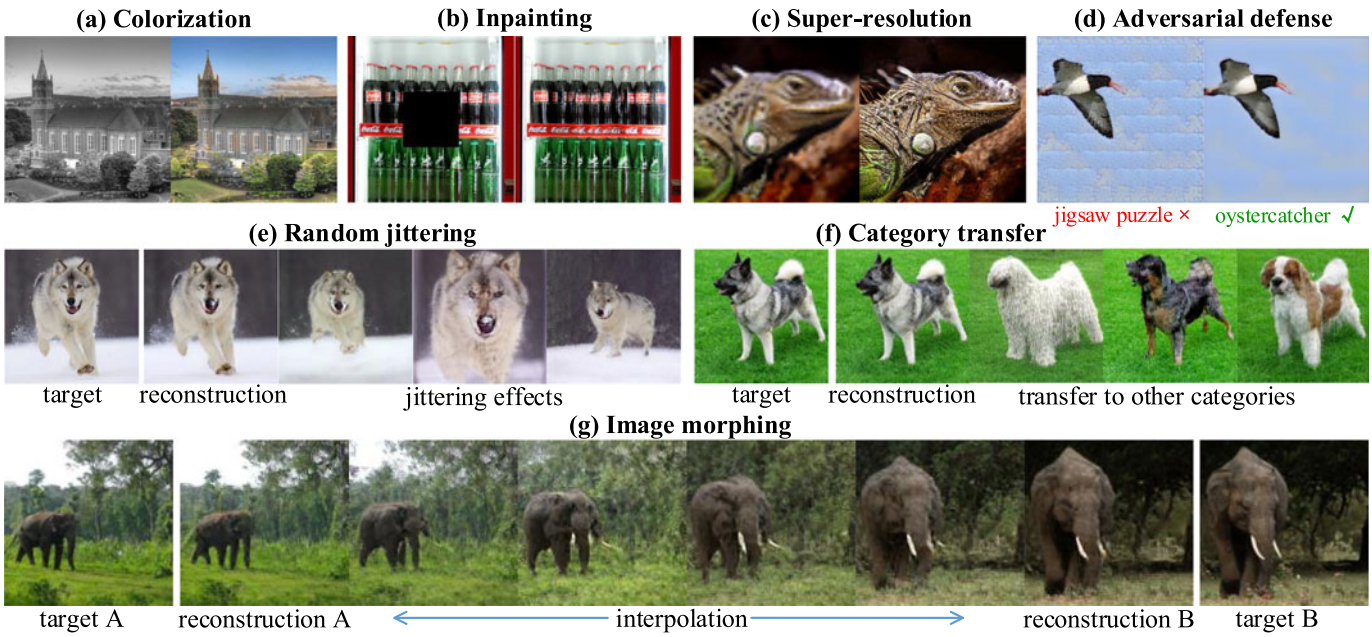

Fig. 1. These image restoration (a)(b)(c)(d) and manipulation (e)(f)(g) effects are achieved by merely leveraging the rich generative prior of a GAN without task-specific modeling. The GAN does not see these images during training.

image, G is a fixed generator, $\mathbf{z}$ and θ are the latent code and generator parameters respectively. In practice, we found that this strategy fails in dealing with complex real-world images. In particular, it often results in mismatched reconstructions, whose details (e.g., objects, texture, and background) appear inconsistent with the original images, as Figs. 2b and 2c show. On one hand, existing GAN inversion methods still suffer from the issues of mode collapse [19] and limited generator capacity, affecting their capability in capturing the desired data manifold. On the other hand, perhaps a more crucial limitation is that when a generator is fixed, the GAN is inevitably limited by the training distribution and its inversion cannot faithfully reconstruct unseen and complex images. It is infeasible to carry such assumptions while using a GAN as prior for general image restoration and manipulation.

Despite the gap between the approximated manifold and the real one, the GAN generator still captures rich statistics of natural images. In order to make use of these statistics while avoiding the aforementioned limitations, in this paper we present a relaxed and more practical reconstruction formulation for mining the priors in GAN. Our first reformulation is to allow the generator parameters to be fine-tuned on the target image on-the-fly, i.e., $\theta^*, \mathbf{z}^* = \arg\min_{\theta,\mathbf{z}} \mathcal{L}(\mathbf{x}, G(\mathbf{z};\theta))$. This lifts the constraint of confining the reconstruction within the training distribution. Relaxing the assumption with fine-tuning, however, is still not sufficient to ensure good reconstruction quality for arbitrary target images. We find that fine-tuning using a standard loss such as perceptual loss [20] or mean squared error (MSE) in DIP could risk wiping out the originally rich priors. Consequently, the reconstruction may become increasingly unnatural during the reconstruction of a degraded image. Fig. 2d shows an example, suggesting that a new loss and reconstruction strategy is needed.

Thus, in our second reformulation, we devise an effective reconstruction strategy that consists of two components:

1) *Feature matching loss from the coupled discriminator* - we make full use of the discriminator of a trained GAN to regularize the reconstruction process. Note that during training, the generator is optimized to mimic massive natural images via gradients provided by the discriminator. It is reasonable to still adopt the discriminator in guiding the generator to match a single image, as the discriminator preserves the original parameter structure of the generator better than other distance metrics. Thus deriving a feature matching loss from the discriminator helps maintain the reconstruction to remain in the natural image space. Although the discriminator feature matching loss is not new in the literature [21], its significance to GAN reconstruction has not been investigated before.

2) *Progressive reconstruction* - we observe that a joint fine-tuning of all parameters of the generator could lead to *'information lingering'*, where missing semantics (e.g., color) do not naturally change along with the content when reconstructing a degraded image. This is because the deep layers of the generator start to match the low-level textures before the high-level configurations are aligned. To address this issue, we propose a progressive reconstruction strategy that fine-tunes the generator gradually from the shallowest layers to the deepest layers. This allows the reconstruction to start with matching high-level configurations and gradually shift its focus on low-level details.

Thanks to the proposed techniques that enable faithful reconstruction while maintaining the generative prior, our approach, namely Deep Generative Prior (DGP), generalizes well to various kinds of image restoration and manipulation tasks, despite that our method is not specially designed for each task. When reconstructing a corrupted image in a task-dependent observation space, DGP tends to restore the missing information, while keeping existing semantic information unchanged. As shown in Figs. 1a, 1b, and 1c, color, missing patches, and details of the given images are well restored, respectively. As illustrated in Figs. 1e and 1f, we

Fig. 2. Comparison of various methods in reconstructing a gray image under the gray-scale observation space using a GAN. Conventional GAN inversion strategies like (b) [15] and (c) [18] produce imprecise reconstruction for the existing semantics. In this work, we relax the generator so that it can be fine-tuned on-the-fly, achieving more accurate reconstruction as in (d)(e)(f), of which optimization is based on (d) VGG perceptual loss, (e) discriminator feature matching loss, and (f) combined with progressive reconstruction, respectively. We highlight that discriminator is important to preserve the generative prior so as to achieve better restoration for the missing information (i.e., color). The proposed progressive strategy eliminates the 'information lingering' artifacts as in the red box in (e).

can manipulate the content of an image by tweaking the latent code or category condition of the generator. Fig. 1g shows that image morphing is possible by interpolating between the parameters of two fine-tuned generators and the corresponding latent codes of these images. To our knowledge, it is the first time these jittering and morphing effects are achieved on images with complex structures like ImageNet [22]. We show more interesting examples in the experiments and the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3115428.

This paper extends our earlier conference version [23] on several aspects: 1) It provides more intuition, visualization, and explanation for the presented method, illustrating how our approach works and its comparisons with prior works. 2) It contains more thorough experimental results, including its application to more GAN architectures [24], [25], comparisons with more recent approaches [26], [27], evaluation on new datasets, and more qualitative results. 3) It provides more technical details and analysis of our method.

2 Related Work

Image Prior. Image priors play an important role in image restoration and editing. Priors that describe various statistics of natural images have been widely developed and adopted in computer vision, including markov random fields [7], [8], [9], dark channel prior [10], and total variation regularizer [11], *etc*. These traditional hand-crafted priors often capture certain statistics and serve dedicated purposes.

In recent years, the seminal work of deep image prior (DIP) [12] shows that the structure of deep convolutional neural networks implicitly captures image statistics, which could also be used as a prior to restore corrupted images. SinGAN [13] fine-tunes a randomly initialized GAN on patches of a single image, achieving various image editing or restoration effects. As DIP and SinGAN are trained from scratch, they have limited access to image statistics beyond the input image, which restrains their applicability in tasks such as image colorization. There are also other deep priors developed for low-level restoration tasks like deep denoiser prior [28], [29], TNRD [30], and LCM [31], but competing with them is not our goal. Instead, our goal is to study and exploit the prior that is captured in GAN for versatile restoration as well as manipulation tasks. Existing attempts that use a pre-trained GAN as a source of image statistics include [32] and [33], which respectively applies to image manipulation, e.g., editing partial areas of an image, and image restoration, e.g., compressed sensing and super-resolution for human faces. As we will show in our experiments, by using a discriminator based distance metric and a progressive fine-tuning strategy, DGP can better preserve image statistics learned by the GAN and thus allows richer restoration and manipulation effects.

Recently, a concurrent work of multi-code GAN prior [27] also conducts image processing by solving the GAN inversion problem. It uses multiple latent codes to reconstruct the target image and keeps the generator fixed, while our method makes the generator image-adaptive by allowing it to be fine-tuned on-the-fly. Another concurrent work PULSE [34] achieves super-resolution on human face using a pre-trained StyleGAN. We will show that our method is task-agnostic and is applicable to more diverse images.

Image Restoration and Manipulation. In this paper, we demonstrate the effect of applying DGP to multiple tasks of image processing, including image colorization [1], image inpainting [3], super-resolution [4], [5], adversarial defence [6], and semantic manipulation [15], [35], [36]. While many task-specific models and loss functions have been proposed to pursue a better performance on a specific restoration task [1], [2], [3], [4], [5], [6], [37], there are also works that apply GAN and design task-specific pipelines to achieve various image manipulation effects [21], [32], [35], [36], [38], such as CycleGAN [35] and StarGAN [36]. Another line of work simply adopts a GAN pre-train for image synthesis to perform image manipulation [15], [39], [40], [41], [42], [43], but is restricted to synthetic images of the GAN itself or real images of limited complexity, e.g., human faces.

In this work we are interested in uncovering the potential of exploiting the GAN prior as a *task-agnostic* solution for real complex images, where we propose several techniques to achieve this goal. Moreover, as shown in Figs. 1e and 1g, with an improved reconstruction process, we successfully achieve image jittering and morphing on ImageNet, while previous methods are insufficient to handle these effects on such complex data.

GAN Inversion. A natural way to utilize generative prior is to conduct image reconstruction via GAN inversion. GAN inversion aims at finding a vector in the latent space that best reconstructs a given image, where the GAN generator is typically fixed. Previous attempts include optimizing the latent code directly via gradient back-propagation [16],

[17], leveraging an additional encoder mapping images to latent codes [44], [45], or a hybrid method of them [15], [46]. Bau, *et al*. [18] further propose to add small perturbations to shallow blocks of the generator to ease the inversion task. While these methods could handle datasets with limited complexities or synthetic images sampled by the GAN itself, we empirically found in our experiments they may produce imprecise reconstructions for complex real scenes, e.g., images in the ImageNet [22]. Recently, the work of StyleGAN [24] enables a new way for GAN inversion by operating in the relaxed intermediate latent spaces [26], [46], but noticeable mismatches are still observed and the inversion for vanilla GAN (e.g., BigGAN [47]) is still challenging. In this paper, instead of directly applying standard GAN inversion, we devise a more practical way to reconstruct a given image using the generative prior, which is shown to achieve better reconstruction results.

3 Method

We first provide some preliminaries on DIP and GAN before discussing how we exploit DGP for image restoration and manipulation.

Deep Image Prior. Ulyanov *et al*. [12] show that image statistics are implicitly captured by the structure of CNN. These statistics can be seen as a kind of image prior, which can be exploited in various image restoration tasks by tuning a randomly initialized CNN on the degraded image: $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} E(\hat{\mathbf{x}}, f(\mathbf{z};\boldsymbol{\theta})), \mathbf{x}^* = f(\mathbf{z};\boldsymbol{\theta}^*)$, where E is a task-dependent distance metric, $\mathbf{z}$ is a randomly sampled latent code, and f is a CNN with $\boldsymbol{\theta}$ being its parameters. $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ are the degraded image and restored image respectively. One limitation of DIP is that the restoration process mainly resorts to existing statistics in the input image, it is thus infeasible to apply DIP on tasks that require more general statistics, such as image colorization [1] and manipulation [15].

Generative Adversarial Networks (GANs). GANs are widely used for modeling complex data such as natural images [14], [24], [48], [49]. In GAN, the underlying manifold of natural images is approximated by the combination of a parametric generator G and a prior latent space $\mathcal{Z}$, so that an image can be generated by sampling a latent code $\mathbf{z}$ from $\mathcal{Z}$ and applying G as $G(\mathbf{z})$. GAN jointly trains G with a parametric discriminator D in an adversarial manner, where D is supposed to distinguish generated images from real ones. Although extensive efforts have been made to improve the power of GAN, there inevitably exists a gap between GAN's approximated manifold and the actual one, due to issues such as insufficient capacity and mode collapse.

3.1 Deep Generative Prior

Suppose $\hat{\mathbf{x}}$ is obtained via $\hat{\mathbf{x}} = \phi(\mathbf{x})$, where $\mathbf{x}$ is the original natural image and ϕ is a degradation transform. e.g., ϕ could be a graying transform that turns $\mathbf{x}$ into a grayscale image. Many tasks of image restoration can be regarded as recovering $\mathbf{x}$ given $\hat{\mathbf{x}}$. A common practice is learning a mapping from $\hat{\mathbf{x}}$ to $\mathbf{x}$, which often requires task-specific training for different ϕs. Alternatively, we can also employ statistics of $\mathbf{x}$ stored in some prior, and search in the space of $\mathbf{x}$ for an optimal $\mathbf{x}$ that best matches $\hat{\mathbf{x}}$, viewing $\hat{\mathbf{x}}$ as partial observations of $\mathbf{x}$.

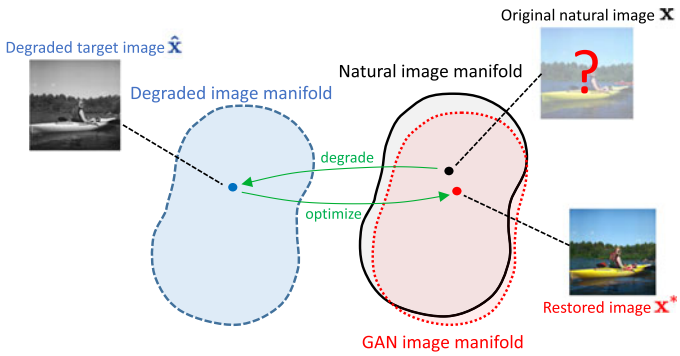

Fig. 3. Image restoration with GAN as a prior. For a degraded target image, we view the GAN image manifold as the approximated natural image manifold, and search for the image that matches the target image in the observation space, which becomes the restored image.

While various priors have been proposed [7], [12], [13] in the second line of research, in this paper we are interested in studying a more generic image prior, i.e., a GAN generator trained on large-scale natural images for image synthesis. Specifically, a straightforward realization is a reconstruction process based on GAN inversion, which optimizes the following objective:

$$\begin{aligned}\mathbf{z}^* &= \arg\min_{\mathbf{z}\in\mathbb{R}^d} E(\hat{\mathbf{x}}, G(\mathbf{z};\boldsymbol{\theta})), \qquad \mathbf{x}^* = G(\mathbf{z}^*;\boldsymbol{\theta}),\\ &= \arg\min_{\mathbf{z}\in\mathbb{R}^d} \mathcal{L}(\hat{\mathbf{x}}, \phi(G(\mathbf{z};\boldsymbol{\theta}))),\end{aligned} \tag{1}$$

where $\mathcal{L}$ is a distance metric such as the L2 distance, G is a GAN generator parameterized by $\boldsymbol{\theta}$ and trained on natural images. Ideally, if G is sufficiently powerful that the data manifold of natural images is well captured in G, the above objective will drag $\mathbf{z}$ in the latent space and locate the optimal natural image $\mathbf{x}^* = G(\mathbf{z}^*;\boldsymbol{\theta})$, which contains the missing semantics of $\hat{\mathbf{x}}$ and matches $\hat{\mathbf{x}}$ under ϕ. For example, if ϕ is a graying transform, $\mathbf{x}^*$ will be an image with a natural color configuration subject to $\phi(\mathbf{x}^*) = \hat{\mathbf{x}}$, as shown in Fig. 3. However, in practice it is not always the case.

As the GAN generator is fixed in Eq. (1) and its improved versions, e.g., adding an extra encoder [15], [44], these reconstruction methods based on the standard GAN inversion suffer from an intrinsic limitation, i.e., there is a gap between the approximated manifold of natural images and the actual one. On one hand, due to issues including mode collapse and insufficient capacity, the GAN generator cannot perfectly grasp the training manifold represented by a dataset of natural images. On the other hand, the training manifold itself is also an approximation of the actual one. Such two levels of approximations inevitably lead to a gap. Consequently, a sub-optimal $\mathbf{x}^*$ is often retrieved, which often contains significant mismatches to $\hat{\mathbf{x}}$, especially when the original image $\mathbf{x}$ is a complex image, e.g., ImageNet [22] images, or an image located outside the training manifold. See Fig. 2 and existing literature [18], [44] for an illustration.

A Relaxed GAN Reconstruction Formulation. Despite the gap between the approximated manifold and the real one, a well trained GAN generator still covers rich statistics of natural images. In order to make use of these statistics while avoiding the aforementioned limitation, we propose a relaxed GAN reconstruction formulation by allowing parameters $\boldsymbol{\theta}$

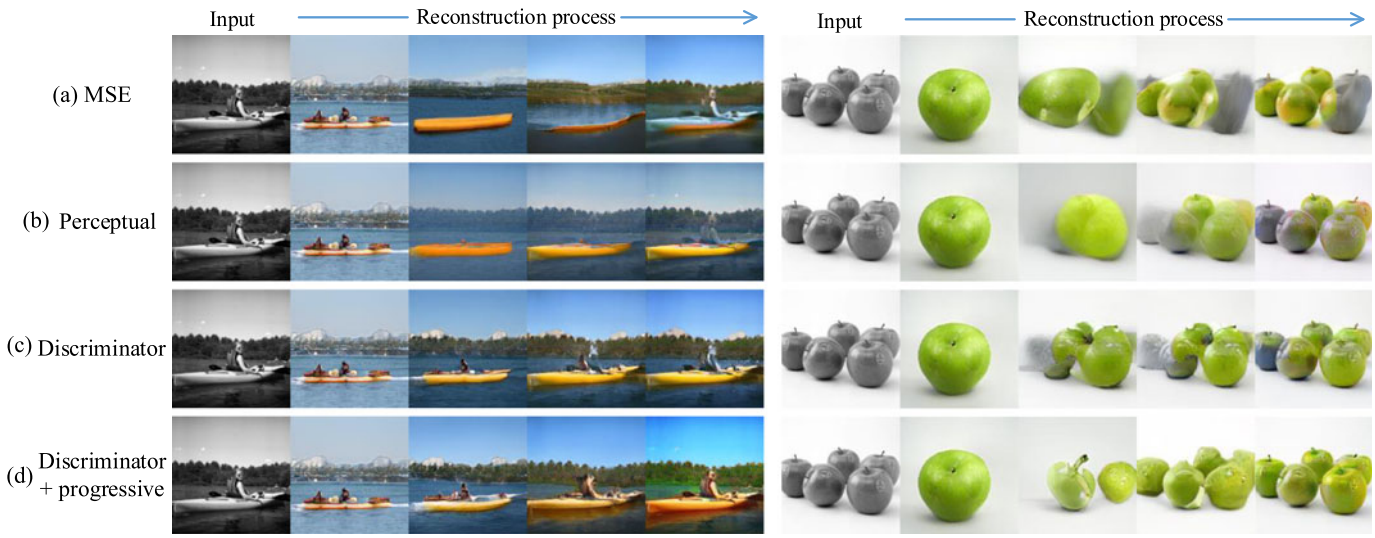

Fig. 4. Comparison of different loss types when fine-tuning the generator to reconstruct the image.

of the generator to be moderately fine-tuned along with the latent code $\mathbf{z}$. Such a relaxation on $\boldsymbol{\theta}$ gives rise to an updated objective:

$$\boldsymbol{\theta}^*, \mathbf{z}^* = \arg\min_{\boldsymbol{\theta},\mathbf{z}} \mathcal{L}(\hat{\mathbf{x}}, \phi(G(\mathbf{z};\boldsymbol{\theta}))), \quad \mathbf{x}^* = G(\mathbf{z}^*;\boldsymbol{\theta}^*). \tag{2}$$

We refer to this updated objective as Deep Generative Prior (DGP). With this relaxation, DGP significantly improves the chance of locating an optimal $\mathbf{x}^*$ for $\hat{\mathbf{x}}$, as fitting the generator to a single image is much more achievable than fully capturing a data manifold. Note that the generative prior buried in G, e.g., its ability to output faithful natural images, might be deteriorated during the fine-tuning process. The key to preserve the generative prior lies in the design of a good distance metric $\mathcal{L}$ and a proper optimization strategy.

3.2 Discriminator Guided Progressive Reconstruction

To fit the GAN generator to the input image $\hat{\mathbf{x}}$ while retaining a natural output, in this section we introduce a discriminator based distance metric, and a progressive fine-tuning strategy.

Discriminator Matters. Given an input image $\hat{\mathbf{x}}$, DGP will start with an initial latent code $\mathbf{z}_0$. In practice, we obtain $\mathbf{z}_0$ by randomly sampling a few hundreds of candidates from the latent space $\mathcal{Z}$ and selecting the one that its corresponding image $G(\mathbf{z};\boldsymbol{\theta})$ best resembles $\hat{\mathbf{x}}$ under the metric $\mathcal{L}$ we used in Eq. (2). As shown in Fig. 4, the choice of $\mathcal{L}$ significantly affects the optimization of Eq. (2). Existing literature often adopts the Mean-Squared-Error (MSE) [12] or the AlexNet/VGGNet based Perceptual loss [15], [20] as $\mathcal{L}$, which respectively emphasize the pixel-wise appearance and the low-level/mid-level texture. However, we empirically found using these metrics in Eq. (2) often cause unfaithful outputs at the beginning of optimization, leading to sub-optimal results at the end. We thus propose to replace them with a discriminator-based distance metric, which measures the L1 distance in the *discriminator feature space*

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i\in\mathcal{I}} \|D(\mathbf{x}_1, i), D(\mathbf{x}_2, i)\|_1, \tag{3}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two images, corresponding to $\hat{\mathbf{x}}$ and $\phi(G(\mathbf{z};\boldsymbol{\theta}))$ in Eqs. (1) and (2), and D is the discriminator that is coupled with the generator. $D(\mathbf{x}, i)$ returns the feature of $\mathbf{x}$ at i-block of D, and $\mathcal{I}$ is the index set of used blocks. Compared to the AlexNet/VGGNet based perceptual loss, the discriminator D is trained along with G, instead of being trained for a separate task. D, being a distance metric, thus is less likely to break the parameter structure of G, as they are well aligned during the pre-training. Moreover, we found the optimization of DGP using such a distance metric visually works like an image morphing process. e.g., as shown in Fig. 4, the person on the boat is preserved and all intermediate outputs are all vivid natural images. It is worth pointing out again while the feature matching loss is not new, this is the first time it serves as a regularizer during GAN reconstruction.

Progressive Reconstruction. Typically, we will fine-tune all parameters of $\boldsymbol{\theta}$ simultaneously during the optimization of Eq. (2). However, we observe an adverse effect of *'information lingering'*, where missing semantics (e.g. color) do not shift along with existing context. Taking Fig. 4c as an example, the leftmost apple fails to inherit the green color of the initial apple when it emerges. One possible reason is deep blocks of the generator G start to match low-level textures before high-level configurations are completely aligned. To overcome this problem, we propose a progressive reconstruction strategy for some restoration tasks.

Specifically, as illustrated in Fig. 5, we first fine-tune the shallowest block of the generator, and gradually continue with blocks at deeper depths, so that DGP can control the global configuration at the beginning and gradually shift its attention to details at lower levels. A demonstration of the proposed strategy is included in Fig. 4d, where DGP splits the apple from one to two at first, then increases the number to five, and finally refines the details of apples. Compared to the non-progressive counterpart, such a progressive strategy better preserves the consistency between missing and existing semantics.

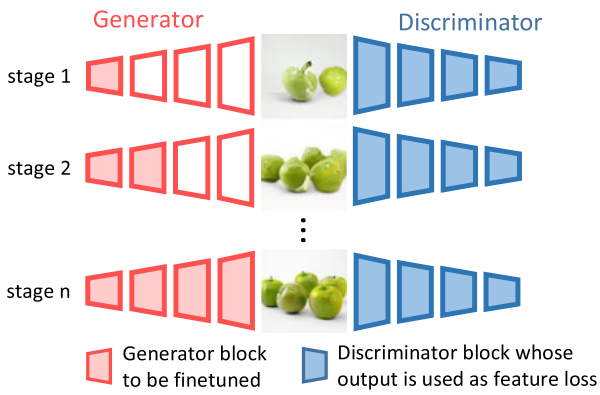

Fig. 5. Progressive reconstruction of the generator can better preserves the consistency between missing and existing semantics in comparison to simultaneous fine-tuning on all the parameters at once. Here the list of images shown in the middle are the outputs of the generator in different fine-tuning stages.

3.3 Technical Details

Architectures. We adopt the BigGAN [47] architectures of 128^2 and 256^2 resolutions in most experiments. BigGAN is selected due to its excellent performance in image generation of diverse object categories. We also study other GANs in experiments. For the 128^2 resolution, we use the best setting of [47], which has a channel multiplier of 96 and a batch size of 2048. As for the 256^2 resolution, the channel multiplier and batch size are respectively set to 64 and 1920 due to limited GPU resources. We train the GANs on the ImageNet training set, and the 128^2 and 256^2 versions have Inception scores of 103.5 and 94.5 respectively. Our experiments are conducted based on PyTorch [50].

Initialization. In order to ease the optimization goal of Eq. (4) in the paper, it is a good practice to start with a latent vector $\mathbf{z}$ that produces an approximate reconstruction. Therefore, we randomly sample 500 images using the GAN, and select the nearest neighbor of the target image under the discriminator feature metric as the starting point. Since encoder based methods tend to fail for degraded input images, they are not used in this work.

Note that in BigGAN, a class condition is needed as input. Therefore, in order to reconstruct an image, its class condition is required. This image classification problem could be solved by training a corresponding deep network classifier and is not the focus of this work, hence we assume the class label is given except for the adversarial defense task. For adversarial defense and images whose classes are not given, both the latent vector $\mathbf{z}$ and the class condition are randomly sampled.

Fine-Tuning. With the above pre-trained BigGAN and initialized latent vector $\mathbf{z}$, we fine-tune both the generator and the latent vector to reconstruct a target image. As the batch size is only 1 during fine-tuning, we use the tracked global statistics (i.e., running mean and running variance) for the batch normalization (BN) [51] layers to prevent inaccurate statistic estimation. The discriminator of BigGAN is composed of a number of residual blocks (6 blocks and 7 blocks for 128^2 and 256^2 resolution versions respectively). The output features of these blocks are used as the discriminator loss, as described in Eq. (6) of the paper. In order to prevent the latent vector from deviating too much from the prior Gaussian distribution, we add an additional L2 loss to the latent vector $\mathbf{z}$ with a loss weight of 0.02. We adopt the ADAM optimizer [52] in all our experiments. The detailed training settings for various tasks are provided in the supplementary material, available online. For inpainting and super-resolution, we use a weighted combination of discriminator loss and MSE loss, as the MSE loss is beneficial for the PSNR metric. For super-resolution, we study two settings, with one biased towards MSE loss while the other biased towards discriminator loss. Our quantitative results on adversarial defense are based on the 256^2 resolution model, while those for other tasks are based on the 128^2 resolution models.

TABLE 1
Comparison With Other GAN Inversion Methods, Including (a) Optimizing Latent Code [16], [17], (b) Learning an Encoder [15], (c) A Combination of (a)(b) [15], (d) Adding Small Perturbations to Early Stages Based on (c) [18], and (e) Using Multiple Latent Codes [27]

	(a)	(b)	(c)	(d)	(e)	Ours
PSNR↑	15.97	11.39	16.46	22.49	20.76	**32.89**
SSIM↑	46.84	32.08	47.78	73.17	63.28	**95.95**
MSE↓ (×e-3)	29.61	85.04	28.32	6.91	13.70	**1.26**

We reported PSNR, SSIM, and MSE of image reconstruction. The results are evaluated on the 1k ImageNet validation set.

4 Applications

We first compare our method with other GAN inversion methods for reconstruction, and then show the application of DGP in a number of image restoration and image manipulation tasks. Following the technical details described in Section 3.3, for most experiments we adopt a BigGAN [47] to progressively reconstruct given images based on discriminator feature loss. For dataset, we use the ImageNet [22] validation set that has not been observed by BigGAN. To quantitatively evaluate our method on image restoration tasks, we test on 1k images from the ImageNet validation set, where the first image for each class is collected to form the test set.

Comparison With Other GAN Inversion Methods. To begin with, we compare with other GAN inversion methods [15], [16], [17], [18], [27] for image reconstruction. As shown in Table 1, our method achieves a very high PSNR and SSIM scores, outperforming other GAN inversion methods by a large margin. It can be seen from Fig. 6 that conventional GAN inversion methods suffer from obvious mismatches between reconstructed images and the target one, where the details or even contents are not well aligned. In contrast, the reconstruction error of DGP is almost visually imperceptible. In the following sections we show that our method also well exploits the generative prior in various applications.

4.1 Image Restoration

Colorization. Image colorization aims at restoring a grayscale image $\hat{\mathbf{x}} \in \mathbb{R}^{H \times W}$ to a colorful image with RGB channels $\mathbf{x} \in \mathbb{R}^{3 \times H \times W}$. To obtain $\hat{\mathbf{x}}$ from the colorful image $\mathbf{x}$, the degradation transform ϕ is a graying transform that only preserves the brightness of $\mathbf{x}$. By taking this degradation transform to Eq. (2), the goal becomes finding the colorful

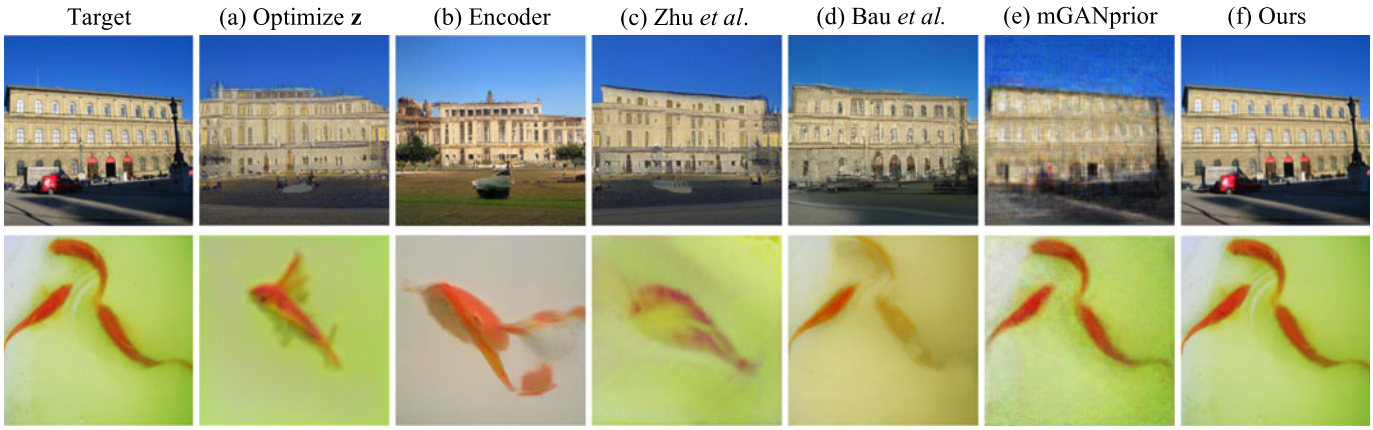

Fig. 6. *Image reconstruction*. We compare our method with other GAN inversion methods including (a) optimizing latent code [16], [17], (b) learning an encoder [15], (c) a combination of (a)(b) [15], (d) adding small perturbations to early stages based on (c) [18], and (e) using multiple latent codes [27].

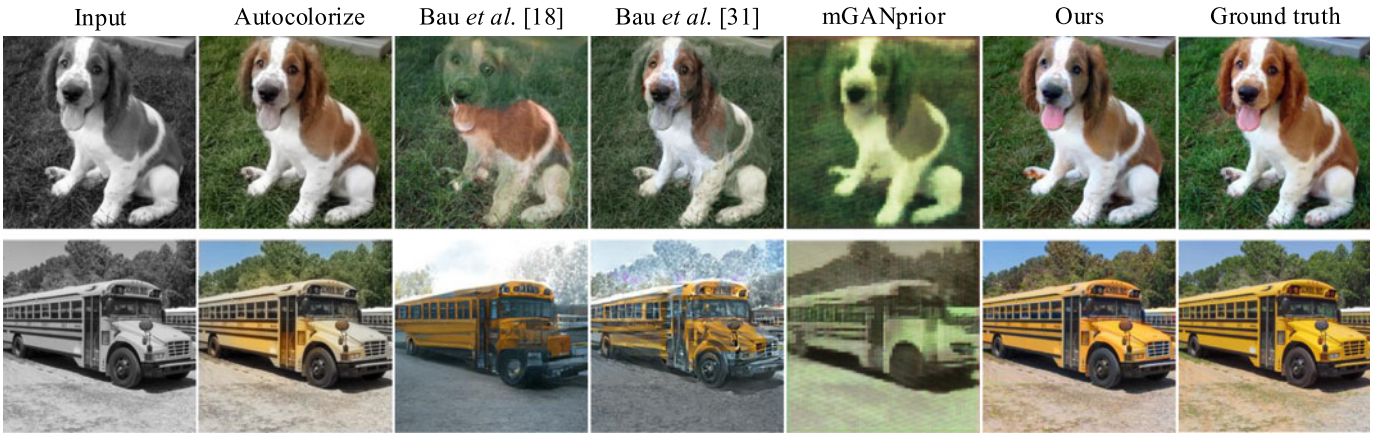

Fig. 7. *Colorization*. Qualitative comparison of Autocolorize [1], other GAN inversion methods [18], [32], mGANprior [27], and our DGP.

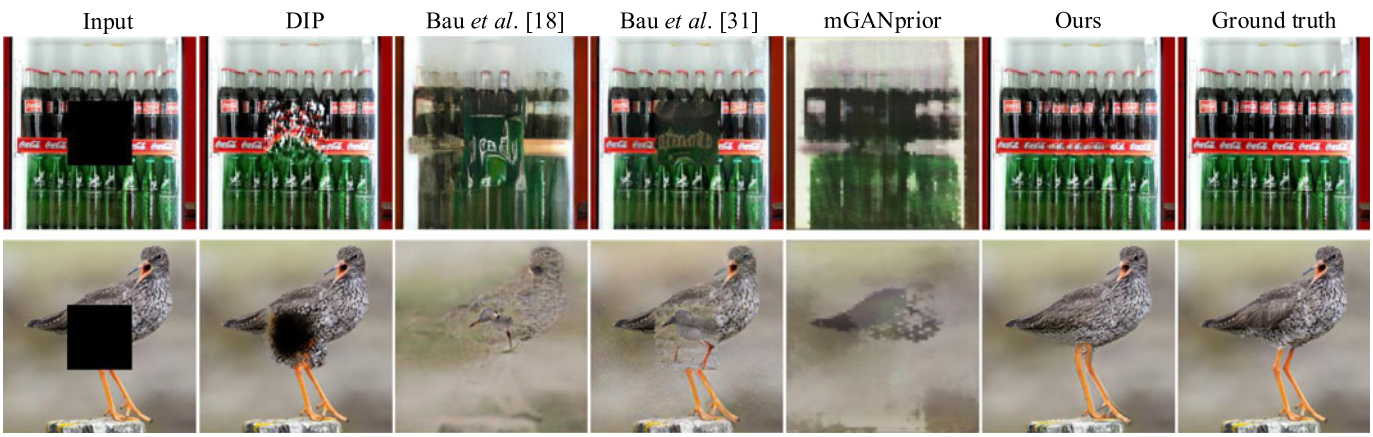

Fig. 8. *Inpainting*. Compared with DIP, [18], [32], and mGANprior, the proposed DGP could preserve the spatial coherence in image inpainting with large missing regions.

image $\mathbf{x}^*$ whose gray-scale image is the same as $\hat{\mathbf{x}}$. We optimize Eq. (2) using back-propagation and the progressive discriminator based reconstruction technique in Section 3.2. Fig. 4d shows the reconstruction process. Note that the colorization task only requires to predict the "ab" dimensions of the Lab color space. Therefore, we transform $\mathbf{x}^*$ to the Lab space, and adopt its "ab" dimensions as well as the given brightness dimension $\hat{\mathbf{x}}$ to produce the final colorful image.

Fig. 7 presents the qualitative comparisons with Autocolorize [1] and other methods. Note that Autocolorize is directly optimized to predict color from gray-scale images while our method does not adopt such task-specific training. Despite so, our method is visually better or comparable to Autocolorize. To evaluate the colorization quality, we report the classification accuracy of a ResNet50 [53] model on the colorized images. The ResNet50 accuracy for Autocolorize [1], Bau *et al*. [18], Bau *et al*. [32], mGANprior [27], and

TABLE 2
Inpainting Evaluation

	DIP	Zhu[15]	Bau[18]	Bau[32]	mcode[27]	Ours
PSNR↑	14.58	13.70	15.01	14.33	15.96	**16.97**
SSIM↑	29.37	33.09	33.95	30.60	42.54	**45.89**

We reported PSNR and SSIM of the inpainted area. The results are evaluated on the 1k ImageNet validation set.

TABLE 3
Super-Resolution ($\times 4$) Evaluation

	DIP	SinGAN	Bau[32]	mcode[27]	Ours (MSE)	Ours (D)
NIQE↓	6.03	6.28	5.05	6.53	5.30	**4.90**
PSNR↑	23.02	20.80	19.89	21.79	**23.30**	22.00
RMSE↓	17.84	19.78	25.42	20.54	**17.40**	20.09

We reported widely used NIQE [54], PSNR, and RMSE scores. The results are evaluated on the 1k ImageNet validation set. (MSE) and (D) indicate which kind of loss DGP is biased to use.

ours are 51.5%, 56.2%, 56.0%, 7.3% and 62.8% respectively, showing that DGP outperforms other baselines on this perceptual metric.

Inpainting. The goal of image inpainting is to recover the missing pixels of an image. The corresponding degradation transform is to multiply the original image with a binary mask $\mathbf{m}$: $\phi(\mathbf{x}) = \mathbf{x} \odot \mathbf{m}$, where $\odot$ is Hadamard's product. As before, we put this degradation transform to Eq. (2), and reconstruct target images with missing boxes. Thanks to the generative image prior of the generator, the missing part tends to be recovered in harmony with the context, as illustrated in Fig. 8. In contrast, the absence of a learned image prior would result in messy inpainting results, as in DIP. Quantitative results indicate that DGP outperforms DIP and other GAN inversion methods by a large margin, as Table 2 shows. It is also observed that the mGANprior approach does not fit well with the BigGAN model in our implementation, as it is originally devised for PGGAN. We would add comparisons based on PGGAN in later experiments.

Super-Resolution. In this task, one is given with a low-resolution image $\hat{\mathbf{x}} \in \mathbb{R}^{3\times H\times W}$, and the purpose is to generate the corresponding high-resolution image $\mathbf{x} \in \mathbb{R}^{3\times fH\times fW}$, where f is the upsampling factor. In this case, the degradation transform ϕ is to downsample the input image by a factor f. Following DIP [12], we adopt the Lanczos downsampling operator in this work.

Fig. 9 and Table 3 show the comparison of DGP with DIP, SinGAN, Bau *et al*. [32] and mGANprior [27]. Our method achieves sharper and more faithful super-resolution results than its counterparts. For quantitative results, we could trade off between perceptual quality like NIQE and commonly used PSNR score by using different combination ratios of discriminator loss and MSE loss at the final fine-tuning stage. For instance, when using higher MSE loss, DGP has excellent PSNR and RMSE performance, and outperforms other counterparts in all the metrics involved. And the perceptual quality NIQE could be further improved by biasing towards discriminator loss.

Fig. 12 illustrates the reconstruction process of the above image restoration tasks. Despite the mismatch at the beginning, the reconstructions finally match the target images and restore the missing semantics.

Flexibility of DGP. The generic paradigm of DGP provides more flexibility in restoration tasks. For example, an image of gray-scale bird may have many possibilities when restored in the color space. Since the BigGAN used in our method is a conditional GAN, we could achieve diversity in colorization by using different class conditions when restoring the image, as Fig. 10a shows. Furthermore, our method allows hybrid restoration, i.e., jointly conducting colorization, inpainting, and super-resolution. This could naturally be achieved by using a composite of degrade transform $\phi(\mathbf{x}) = \phi_a(\phi_b(\phi_c(\mathbf{x})))$, as shown in Fig. 10b.

Generalization of DGP. We also test our method on images not belonging to ImageNet. As Fig. 11 shows, DGP restores the color and missed patches of these images reasonably well. Particularly, compared with DIP, DGP fills the missed patches to be well aligned with the context. This indicates that DGP does capture the '*spatial coherence*' prior of natural images, instead of memorizing the ImageNet dataset. We further evaluate our method on 500 images of the Places [55] validation set for inpainting. The PSNR of our method, DIP, and [18] are 16.52, 14.17, and 14.59 respectively, which is consistent with the results on ImageNet.

Such generalization capacity attributes to several aspects. First, the ImageNet dataset used for training is very diverse and covers large-scale object categories. Second, convolutional neural networks are known to have an inductive bias

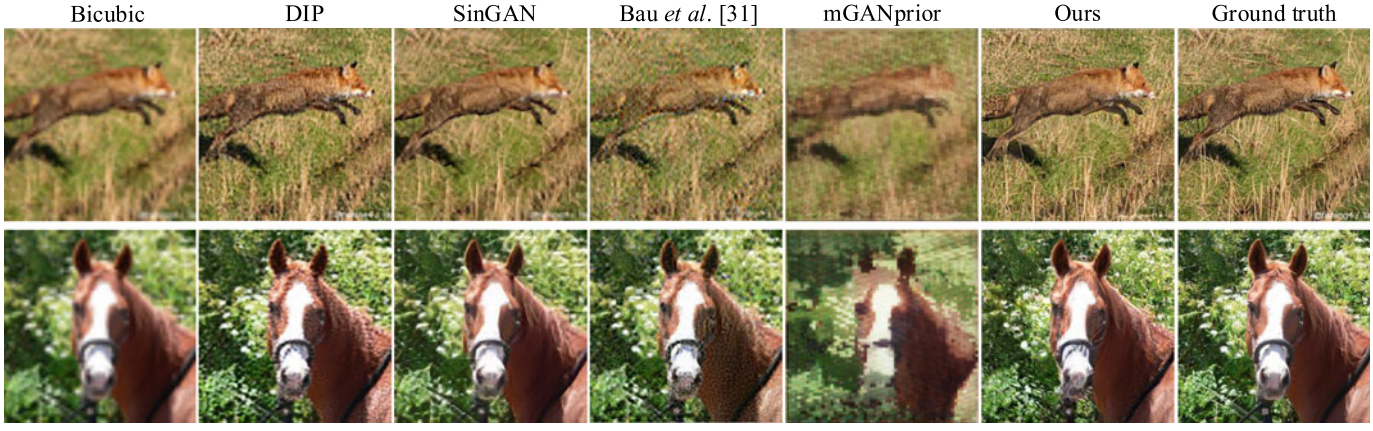

Fig. 9. *Super-resolution ($\times 4$)* on 64×64 size images. The comparisons of our method with DIP, SinGAN, [32], and mGANprior are shown, where DGP produces sharper super-resolution results.

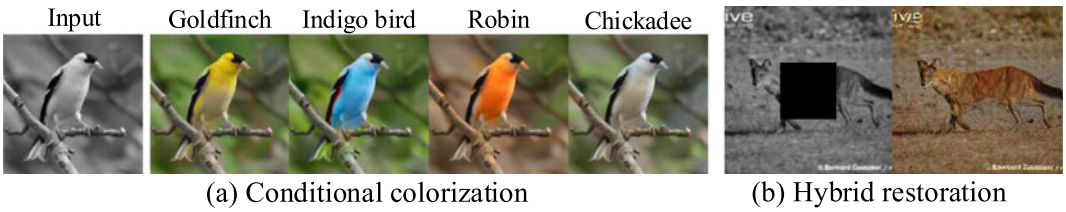

Fig. 10. (a) Colorizing an image under different class conditions. (b) Simultaneously conduct colorization, inpainting, and super-resolution ($\times 2$).

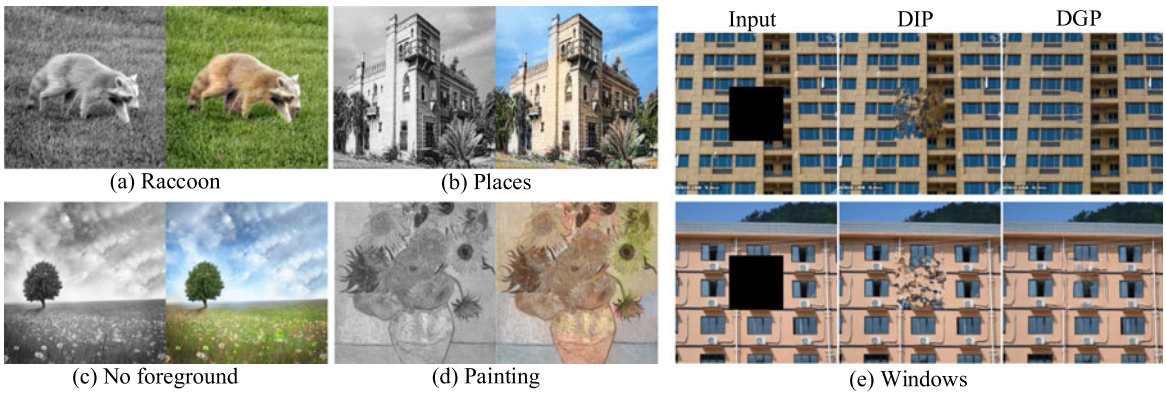

Fig. 11. Evaluation of DGP on non-ImageNet images, including (a) 'Raccoon', a category not belonging to ImageNet categories, (b) an image from Places dataset [55], (c) an image without foreground object, (d) a painting, and (e) windows. (a)(c)(d)(e) are scratched from Internet.

to generalize on images. Third, some statistical properties of images, like the spatial coherence, are shared across images of different domains, as evidenced by our results in Fig. 11e. However, like many data-driven approaches, DGP would also notice performance drop on radically different domains, like colorization on a painting as shown in Fig. 11d. Such cases may require more careful treatment like transfer learning or domain adaptation.

Ablation Study. To validate the effectiveness of the proposed discriminator guided progressive reconstruction method, we compare different fine-tuning strategies in Table 4. There is a clear improvement of discriminator feature matching loss over MSE and perceptual loss, and the combination of the progressive reconstruction further boosts the performance. Fig. 14 provides qualitative comparisons. The results show that the progressive strategy effectively eliminates the 'information lingering' artifacts.

Fig. 12. The reconstruction process of DGP in various image restoration tasks.

Application to StyleGAN and PGGAN. Our method is not limited to a specific GAN model. Here we apply our method to StyleGAN [24] and PGGAN [25], and compare with Image2StyleGAN [26] and mGANprior[27], which are the state-of-the-art GAN inversion methods for StyleGAN and PGGAN respectively. The results for image inpainting are shown in Fig. 13 and Table 5, where the quantitative results are calculated on the data samples in the mGANprior repository.[1] Compared with other baselines, our method achieves more accurate reconstruction in both masked areas and unmasked areas. The results are consistent with those of BigGAN.

Adversarial Defense. Adversarial attack methods aim at fooling a CNN classifier by adding a certain perturbation $\Delta\mathbf{x}$ to a target image $\mathbf{x}$ [56]. In contrast, adversarial defense aims at preventing the model from being fooled by attackers. Specifically, the work of DefenseGAN [6] proposed to restore a perturbed image to a natural image by reconstructing it with a GAN. It works well for simple data like MNIST, but would fail for complex data like ImageNet due to poor reconstruction. Here we show the potential of DGP in adversarial defense under a black-box attack setting [57], where the attacker does not have access to the classifier and defender.

For adversarial attack, the degradation transform is $\phi(\mathbf{x}) = \mathbf{x} + \Delta\mathbf{x}$, where $\Delta\mathbf{x}$ is the perturbation generated by the attacker. Since calculating $\phi(\mathbf{x})$ is generally not differentiable, here we adopt DGP to directly reconstruct the adversarial image $\hat{\mathbf{x}}$. To prevent $\mathbf{x}^*$ from overfitting to $\hat{\mathbf{x}}$, we stop the reconstruction when the MSE loss reaches 5e-3. We adopt the adversarial transformation networks attacker [57] to produce the adversarial samples.[2]

1. *https://github.com/genforce/mganprior*
2. We use the code at *https://github.com/pfnet-research/nips17-adversarial-attack*

TABLE 4
Comparison of Different Loss Type and Fine-Tuning Strategy

Task	Metric	MSE	Perceptual	Discriminator	Discriminator +Progressive
Colorization	ResNet50↑	49.1	53.9	56.8	**62.8**
SR	NIQE↓	6.54	6.27	6.06	**4.90**
	PSNR↑	21.24	20.30	21.58	**22.00**

As Fig. 15 shows, the generated adversarial image contains unnatural perturbations, leading to misclassification for a ResNet50 [53]. After reconstructing the adversarial samples using DGP, the perturbations are largely alleviated, and the samples are thus correctly classified. The comparisons of our method with DefenseGAN and DIP are shown in Table 6. DefenseGAN yields poor defense performance due to inaccurate reconstruction. And DGP outperforms DIP, thanks to the learned image prior that produces more natural restored images.

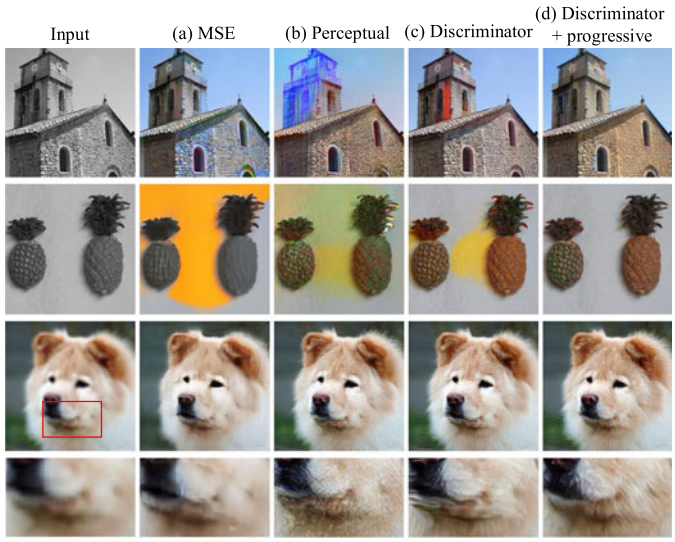

Fig. 14. Comparison of different loss types and optimization techniques in colorization and super-resolution.

4.2 Image Manipulation

Since DGP enables precise GAN reconstruction while preserving the generative property, it becomes straightforward to apply the fascinating capabilities of GAN to real images like random jittering, image morphing, and category transfer. In this section, we show the application of our method in these image manipulation tasks.

Random Jittering. We show the random jittering effects of DGP, and compare it with SinGAN. Specifically, after reconstructing a target image using DGP, we add Gaussian noise to the latent code $\mathbf{z}^*$ and see how the output changes. As shown in Fig. 17, the wolf in the image changes in pose, action, and size, where each variant looks like a natural shift of the original image. For SinGAN, however, the jittering effects seem to preserve some texture, but losing the concept of 'wolf'. This is because it cannot learn a valid representation of wolf by looking at only one wolf. In contrast, in DGP the generator is fine-tuned in a moderate way such that the structure of image manifold captured by the generator is well preserved. Therefore, perturbing $\mathbf{z}^*$ corresponds to shifting the image in the natural image manifold. We show more random jittering effects in Fig. 16.

TABLE 5
Comparison With Other Methods on StyleGAN and PGGAN for Image Inpainting

GAN	Method	Church		Bedroom		Conference	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
StyleGAN	z	14.75	57.60	15.36	55.06	15.53	53.33
	w	19.77	72.01	19.38	67.64	18.55	63.50
	w+	21.65	82.13	20.27	80.86	21.87	79.78
	Ours	**23.91**	**90.82**	**22.81**	**89.27**	**24.71**	**89.90**
PGGAN	z	18.11	65.67	17.04	58.41	16.58	56.38
	mcode	22.17	83.06	22.16	83.98	22.50	83.77
	Ours	**23.60**	**91.16**	**25.19**	**91.22**	**23.36**	**89.05**

'z' indicates optimizing the original latent code, while 'w' indicates optimizing the intermediate latent code of StyleGAN. 'w+' is the relaxed version of 'w', which corresponds to the Image2StyleGAN method [26]. 'mcode' is to use multiple latent codes as in [27].

Image Morphing. The purpose of image morphing is to achieve a visually sound transition from one image to another. Given a GAN generator G and two latent codes $\mathbf{z}_A$

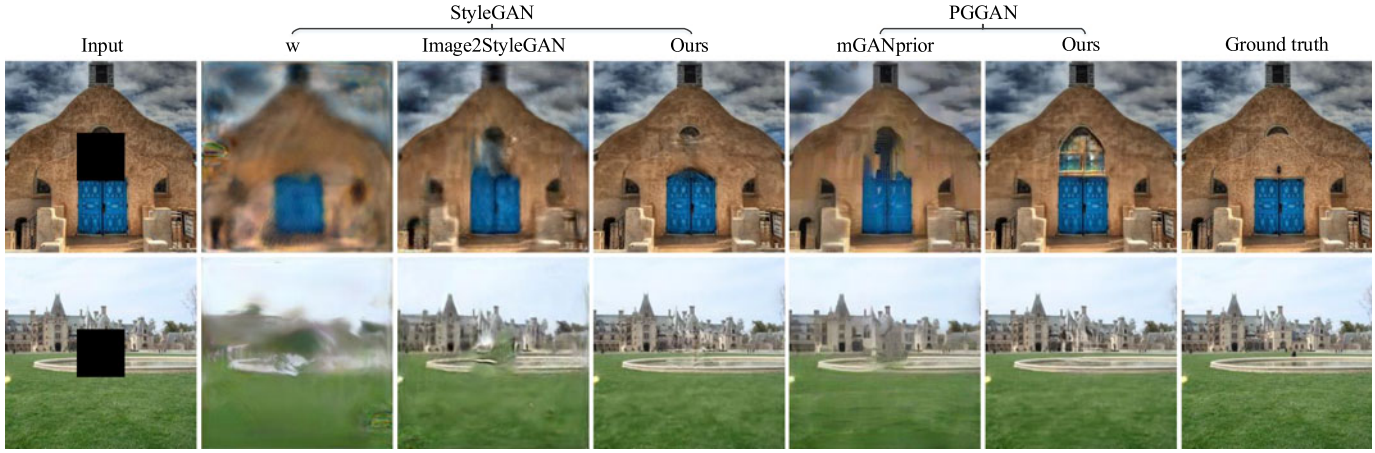

Fig. 13. Application of our method on StyleGAN and PGGAN for image inpainting. Our method shows better reconstruction in both masked and unmasked areas.

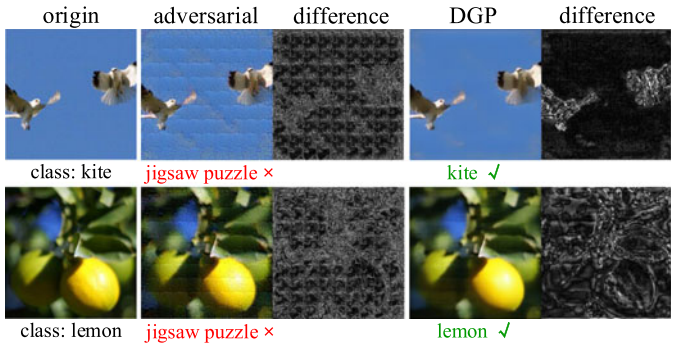

Fig. 15. *Adversarial defense.* DGP is capable of filtering out unnatural perturbations in the adversarial samples by reconstructing them.

and $\mathbf{z}_B$, morphing between $G(\mathbf{z}_A)$ and $G(\mathbf{z}_B)$ could naturally be done by interpolating between $\mathbf{z}_A$ and $\mathbf{z}_B$. In the case of DGP, however, reconstructing two target images $\mathbf{x}_A$ and $\mathbf{x}_B$ would result in two generators G_{θ_A} and G_{θ_B}, and the corresponding latent codes $\mathbf{z}_A$ and $\mathbf{z}_B$. Inspired by [58], to morph between $\mathbf{x}_A$ and $\mathbf{x}_B$, we apply linear interpolation to both the latent codes and the generator parameters: $\mathbf{z} = \lambda\mathbf{z}_A + (1-\lambda)\mathbf{z}_B, \boldsymbol{\theta} = \lambda\boldsymbol{\theta}_A + (1-\lambda)\boldsymbol{\theta}_B, \lambda \in (0,1)$, and generate images with the new $\mathbf{z}$ and $\boldsymbol{\theta}$, as shown in Fig. 18. Since the generator is moderately fine-tuned, θ_A and θ_B are similar. Thus, their interpolations are still valid generator parameters that could produce reasonable output images.

As Fig. 19 shows, our method enables highly photo-realistic image morphing effects. Despite the existence of complex backgrounds, the imagery contents shift in a natural way. To quantitatively evaluate image morphing quality, we apply image morphing to every consecutive image pairs for each class in the ImageNet validation set, and collect the intermediate images where $\lambda = 0.5$. For 50k images with 1k classes, this would create 49k generated images. We evaluate the image quality using Inception Score (IS) [19], and compare DGP with DIP, which adopts a similar network interpolation strategy. Finally, DGP achieves a satisfactory IS, 59.9, while DIP fails to create valid morphing results, leading to only 3.1 of IS.

TABLE 6
Adversarial Defense Evaluation

method	clean image	adversarial	DefenceGAN	DIP	Ours
top1 acc. (%)	74.9	1.4	0.2	37.5	41.3
top5 acc. (%)	92.7	12.0	1.4	61.2	65.9

We reported the classification accuracy of a ResNet50. The results are evaluated on the 1k ImageNet validation set.

We also provide qualitative comparison of our approach with DIP and other GAN inversion methods in Fig. 21. It is observed that DGP performs notably better than other counterparts, showing that the discriminator feature matching loss better preserves the property of the GAN generator.

Category Transfer. In conditional GAN, the class condition controls the content to be generated. So after reconstructing a given image via DGP, we can manipulate its content by tweaking the class condition. Figs. 1f and 20 present examples of transferring the object category of given images. Our method can transfer the dog and bird to various other categories without changing the pose, size, and image configurations.

5 Analysis and Discussions

The Effects of Fine-Tuning Generator. While our method allows the generator parameters to be adapted to a single target image, it would be interesting to investigate *how the generator is changed after fine-tuning, and how such change affects other samples.* Given a target image $\mathbf{x}$, we show both the reconstructed image $G(\mathbf{z}^*;\boldsymbol{\theta}^*)$ and the result using the original generator $G(\mathbf{z}^*;\boldsymbol{\theta})$, as shown in Fig. 22. We observe that $G(\mathbf{z}^*;\boldsymbol{\theta}^*)$ and $G(\mathbf{z}^*;\boldsymbol{\theta})$ are similar in the high-level concept and overall layout (e.g., a wolf standing on a snowfield), but differ mainly in some mid-level and low-level positions and textures, like the exact positions of its legs, whether its mouth is open, the pattern of its fur, *etc*. This shows that the latent code could match the overall layout but is insufficient to align

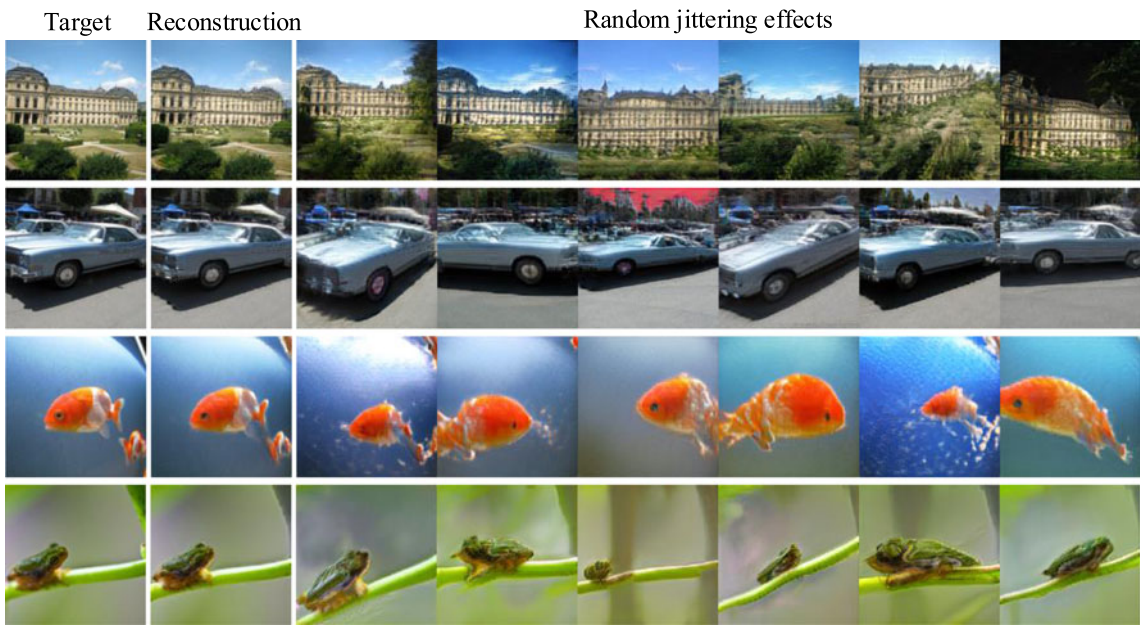

Fig. 16. *Random jittering.* After reconstruction, we could add random noise to the latent code to obtain diverse random jittering effects.

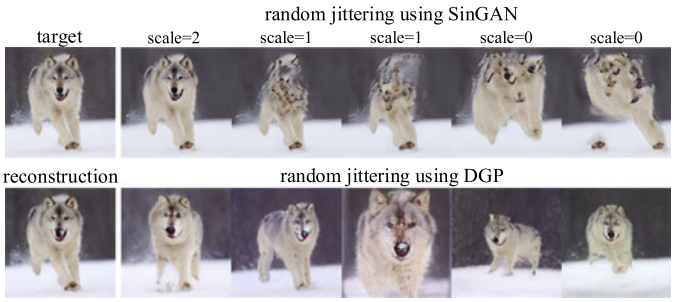

Fig. 17. Comparison of random jittering using SinGAN (above) and DGP (below).

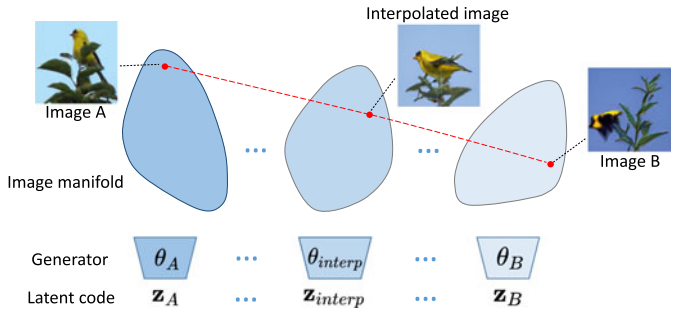

Fig. 18. Image morphing approach. To interpolate between two images, we reconstruct them using DGP, then interpolate between both the latent codes and generator parameters. Interpolating between generator parameters corresponds to creating a serious of image manifolds, which also produce natural images.

fine details, and that is why we still need to tune the generator weights accordingly. Besides, we further visualize the difference of θ and θ^* for other random latent codes ($\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ in Fig. 22). Interestingly, the effects of generator fine-tuning in these new samples inherit some common aspects with the difference between $G(\mathbf{z}^*;\theta^*)$ and $G(\mathbf{z}^*;\theta)$. For instance, in the first example, the ground is getting lighter while the background is getting darker, and the wolf is getting farther from the camera. And for the second example, the nose and face of the dog are getting lighter. These results are evidence that the semantic shift of the generator is *generalizable* instead of being instance-specific. That is to say, by adapting the generator to a target image, the generator learns some properties of this image that also generalize to other samples, showing that the generative property of the generator still preserves.

DGP is External Learning + Internal Learning. As DIP and SinGAN learn from the target image only, they fall into the

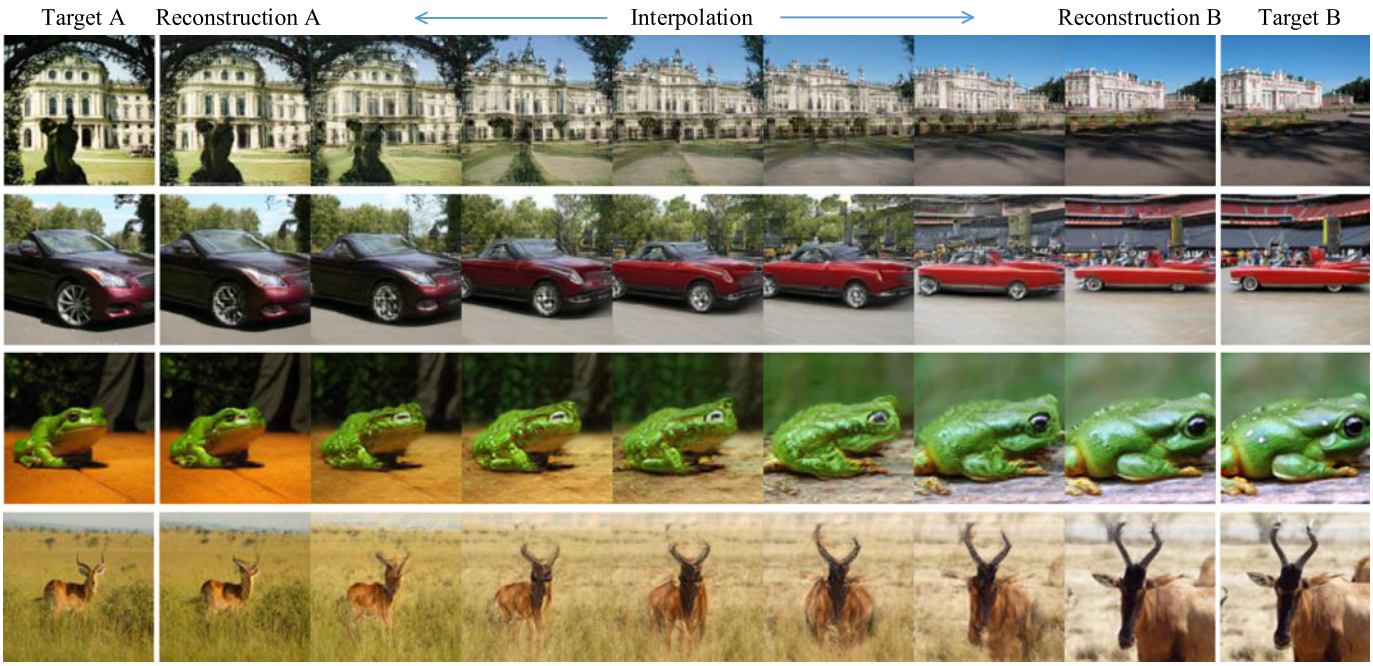

Fig. 19. *Image morphing.* Our method achieves visually realistic image morphing effects by interpolating between both the latent codes and generators.

Fig. 20. *Category transfer.* DGP enables the editing of semantics of objects in images by changing the input class condition.

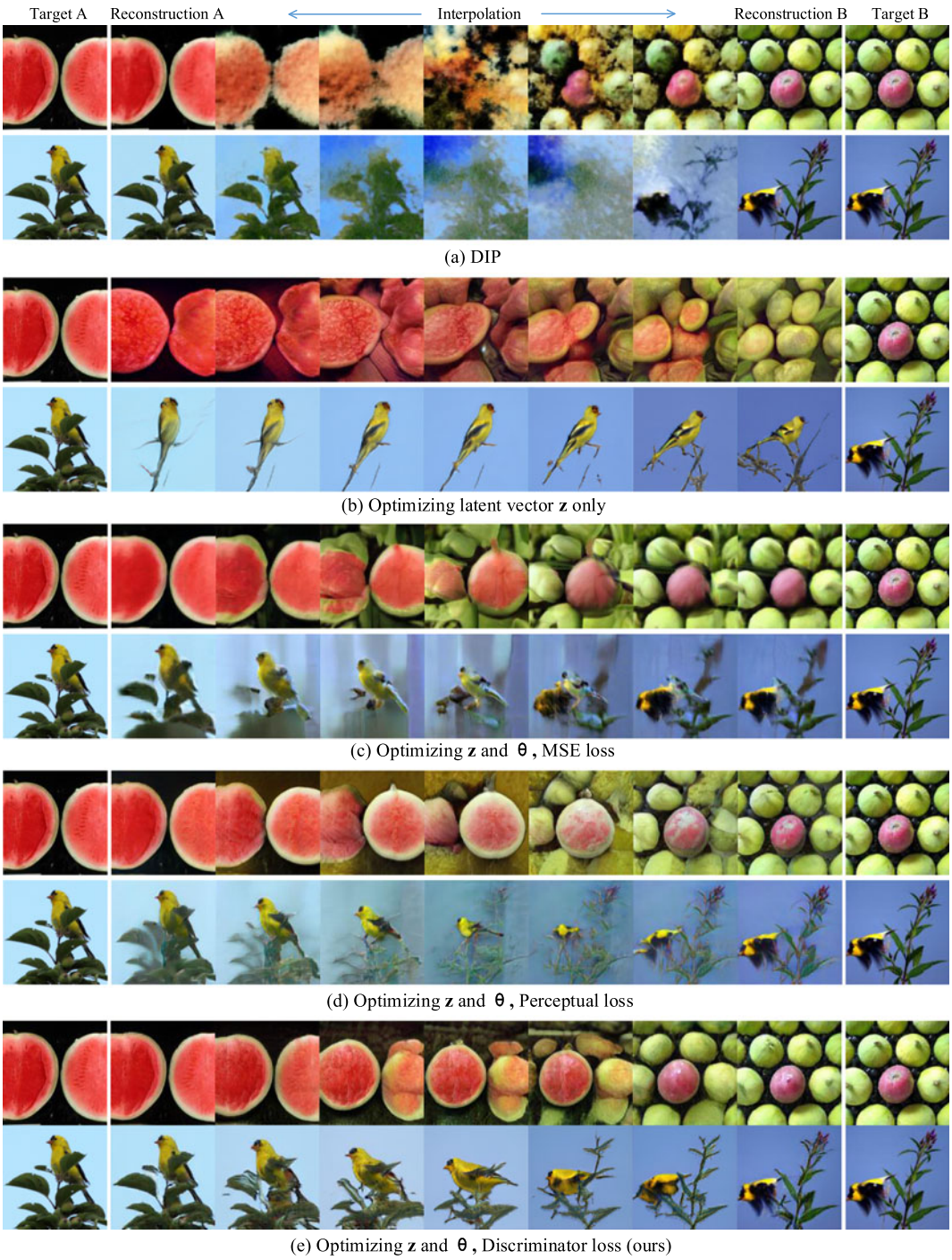

Fig. 21. Comparison of various methods in image morphing, including (a) using DIP, (b) optimizing the latent vector $\mathbf{z}$ of the pre-trained GAN, and (c) (d)(e) optimizing both $\mathbf{z}$ and the generator parameter θ with (c) MSE loss, (d) perceptual loss with VGG network, and (e) discriminator feature matching loss. (b) fails to produce accurate reconstruction while (a)(c)(d) could not obtain realistic interpolation results. In contrast, our results in (e) are much better.

internal learning [59] paradigm. In contrast, traditional GAN inversion methods belong to *external learning*, as they use fixed pre-trained GANs. In DGP, the pre-trained GAN also learns from the target image, thus can be viewed as an integration of both external and internal learning. The advantage of such a strategy is depicted in Fig. 23. In many cases, internal learning methods may not be sufficient to regularize the results in natural image manifold, e.g., for colorization. While GAN inversion could keep the solutions look natural, they often fail to match the target images in the observation space, as the GAN manifold is much smaller than the real image manifold. By integrating external learning with internal learning, the solution reaches the space $\mathbf{x} : E(\hat{\mathbf{x}}, \mathbf{x}) = 0$ while staying in or close to the natural image manifold. As shown in Fig. 11d and other experiments, external learning often helps internal learning, and their integration outperforms any individual of them.

Failure Cases. Although our method has shown compelling results as a generic image prior, negative results are also

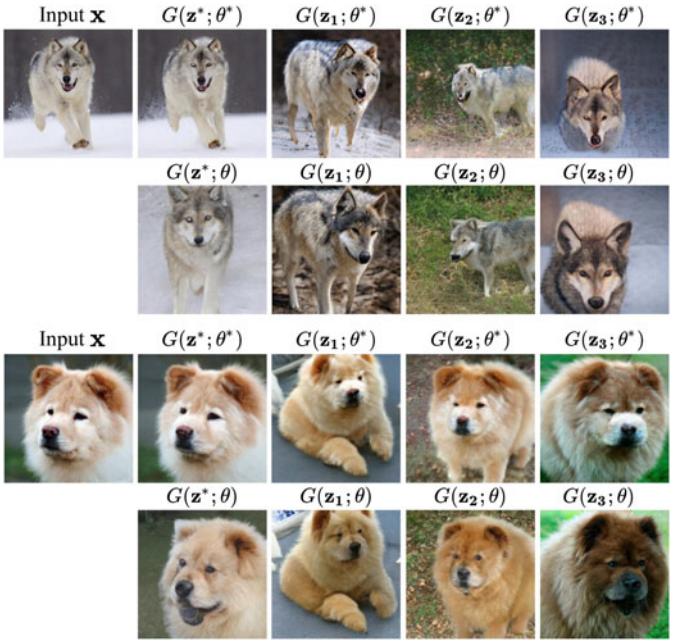

Fig. 22. *Visualizing the effects of fine-tuning generator*. For each example, we show both the reconstructed image $G(\mathbf{z}^*;\theta^*)$ and the result of the original generator $G(\mathbf{z}^*;\theta)$. We also show results on randomly sampled latent codes ($\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$). The latent code matches the overall layout while fine details are aligned by fine-tuning the generator.

observed for some challenging cases. Fig. 24 shows some typical failure cases. For instance, when there are multiple objects or small faces in the image, some of them may not be correctly restored, e.g., some vegetables in the first example do not get the correct colors. A possible reason is that the image manifold of the GAN may miss these data 'modes', due to the well known 'mode collapse' issue of GANs [19]. This could possibly be alleviated by techniques that improve mode coverage of GANs [60], [61].

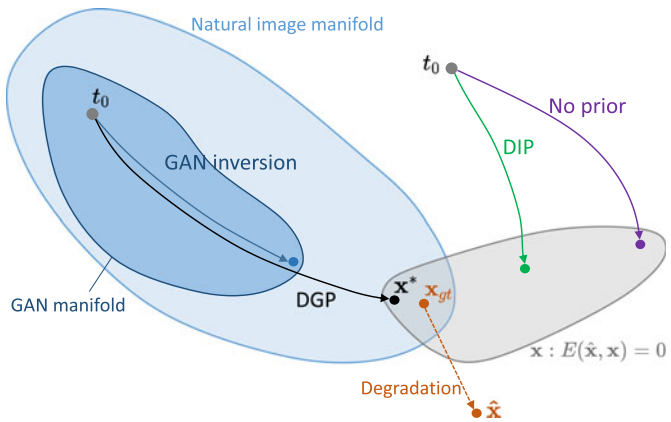

Fig. 23. *Restoration with priors - image space visualization*. We illustrate the restoration process for different priors when reconstructing an image $\mathbf{x}_{gt}$ from a degraded observation $\hat{\mathbf{x}}$ (e.g., a gray image). The ground-truth solution $\mathbf{x}_{gt}$ should belong to a manifold that have null energy $\mathbf{x} : E(\hat{\mathbf{x}}, \mathbf{x}) = 0$ (shown in gray) as in Eq. (1), and also belong to the natural image manifold. DIP typically results in a solution closer to $\mathbf{x}_{gt}$ than no prior, but may not be sufficient to regularize the result in the natural image manifold,e.g., for colorization. In contrast, GAN inversion regularizes the solution in GAN manifold, but often does not reach $\mathbf{x} : E(\hat{\mathbf{x}}, \mathbf{x}) = 0$. The proposed DGP combines the advantages of both, resulting in a solution closer to $\mathbf{x}_{gt}$.

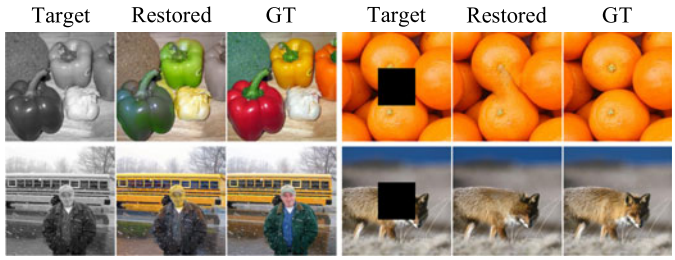

Fig. 24. Failure cases of our approach on colorization and inpainting.

6 Conclusion

To summarise, we have shown that a GAN generator trained on massive natural images can be used as a generic image prior, namely deep generative prior (DGP). Embedded with rich knowledge on natural images, DGP is capable of restoring the missing or corrupted information of a degraded image by progressively reconstructing it under the discriminator metric. Notably, such a reconstruction strategy addresses the challenge of precise image reconstruction via GANs, achieving multiple visually realistic image manipulation effects. Our results uncover the potential of a universal image prior captured by a GAN in image restoration and manipulation.

There are several possible directions for future work. First, the current optimization-based approach typically takes around one minute to process an image. We could accelerate this by training a degradation-agnostic encoder that maps the target image to the latent code. We may also learn to predict the change of the generator parameters to reduce the number of iterations needed for fine-tuning. Second, we also hope to improve its performance in challenging cases by enlarging the mode coverage of GANs. This could possibly be achieved by scaling up training data and the GAN model, or by leveraging algorithms that explicitly motivate mode seeking [60], [61]. Third, our approach now assumes known degradation transform for restoration. How to extend DGP to blind restoration settings where the degradation transform is not given remains an interesting problem. Finally, we believe the image prior of GAN has not been fully explored and might be extended to broader tasks like image deblur, image harmonizing, and image enhancement.

Acknowledgments

This research was conducted in collaboration with SenseTime.

References

[1] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 577–593.
[2] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.
[3] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson , and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6882–6890.
[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[5] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 105–114.

[6] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Representations*, 2018.

[7] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 860–867.

[8] S. C. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 11, pp. 1236–1250, Nov. 1997.

[9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[10] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[11] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D, Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.

[12] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9446–9454.

[13] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4569–4579.

[14] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[15] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, 597–613.

[16] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1967–1974, Jul. 2019.

[17] M. Albright and S. McCloskey , "Source generator attribution via inversion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 96–103.

[18] D. Bau *et al.*, "Seeing what a GAN cannot generate," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4502–4511.

[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[20] J. Johnson, A. Alahi, and L. Fei-Fei , "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[21] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei , "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[23] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, "Exploiting deep generative prior for versatile image restoration and manipulation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 262–277.

[24] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4396–4405.

[25] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018.

[26] R. Abdal, Y. Qin, and P. Wonka, "Image2styleGAN: How to embed images into the styleGAN latent space?," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4431–4440.

[27] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code GAN prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3009–3018.

[28] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3929–3938.

[29] S. A. Bigdeli, M. Zwicker, P. Favaro, and M. Jin, "Deep mean-shift priors for image restoration," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 763–772.

[30] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[31] S. Athar, E. Burnaev, and V. Lempitsky, "Latent convolutional models," in *Proc. Int. Conf. Learn. Representations*, 2018.

[32] D. Bau *et al.*, "Semantic photo manipulation with a generative image prior," *ACM Trans. Graph.*, vol. 38, no. 4, 2019, Art. no. 59.

[33] S. A. Hussein, T. Tirer, and R. Giryes, "Image-adaptive GAN based reconstruction," 2019, *arXiv: 1906.05284*.

[34] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2434–2442.

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[36] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8789–8797.

[37] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.

[38] C. Yang, Y. Shen, and B. Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," 2019, *arXiv:1911.09267*.

[39] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9240–9249.

[40] R. Abdal, Y. Qin, and P. Wonka, "Image2styleGAN++: How to edit the embedded images?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8293–8302.

[41] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," 2020, *arXiv:2007.06600*.

[42] E. Hrknen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANspace: Discovering interpretable GAN controls," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020.

[43] X. Pan, B. Dai, Z. Liu, C. C. Loy, and P. Luo, "Do 2D GANs know 3D shape? Unsupervised 3D shape reconstruction from 2D image GANs," in *Proc. Int. Conf. Learn. Representations*, 2021.

[44] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[45] M. Huh, R. Zhang, J.-Y. Zhu, S. Paris, and A. Hertzmann, "Transforming and projecting images into class-conditional generative networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 17–34.

[46] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 592–608.

[47] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. Int. Conf. Learn. Representations*, 2019.

[48] Y. Xiangli, Y. Deng, B. Dai, C. C. Loy, and D. Lin, "Real or not real, that is the question," in *Proc. Int. Conf. Learn. Representations*, 2020.

[49] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional GAN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2989–2998.

[50] A. Paszke *et al.* "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[54] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a completely blind image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[55] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, pp. 1452–1464, Jun. 2018.

[56] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[57] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, *arXiv:1703.09387*.

[58] X. Wang, K. Yu, C. Dong, X. Tang, and C. C. Loy, "Deep network interpolation for continuous imagery effect transition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1692–1701.
[59] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3118–3126.
[60] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1429–1437.
[61] N. Yu, K. Li, P. Zhou, J. Malik, L. Davis, and M. Fritz, "Inclusive GAN: Improving data and minority coverage in generative models," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 377–393.

Xingang Pan received the BE degree from Tsinghua University in 2016 and the PhD degree from the Department of Information Engineering, The Chinese University of Hong Kong, in 2021. His research interests include 2D/3D visual synthesis and inverse graphics.

Xiaohang Zhan received the BE degree from Tsinghua University in 2016 and the PhD degree from the Department of Information Engineering, The Chinese University of Hong Kong, in 2020. His research interests include unsupervised learning, self-supervised learning, and face analysis.

Bo Dai received the PhD degree in information engineering from The Chinese University of Hong Kong in 2018. He is currently a research assistant professor with S-Lab, Nanyang Technological University, Singapore. His research interests include generative models, video analysis, and cross-modality analysis.

Dahua Lin received the PhD degree from MIT in 2012. He is currently an associate professor with the Department of Information Engineering, The Chinese University of Hong Kong, the director of CUHK-SenseTime Joint Lab, and a co-director of the Centre of Perceptual and Interactive Intelligence. From 2012 to 2014, he was a research assistant professor with Toyota Technological Institute at Chicago. He has authored or coauthored more than 130 papers in top conferences and journals, including the ICCV, CVPR, ECCV, NIPS, and the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. His research interests include computer vision, deep learning, and large-scale data analytics. In recent years, he primarily focuses on high-level visual understanding, video analytics, and cross-modality modeling, significantly pushing forward the state of the art in these areas. He was the recipient of the Best Student Paper Award in NIPS 2010 and the Outstanding Reviewer Awards in ICCV 2009 and ICCV 2011. He was also the recipient of multiple awards in ImageNet, ActivityNet, and COCO. He has supervised or co-supervised the CUHK team in international competitions. He is on the editorial board of *International Journal of Computer Vision* and is an area chair of major conferences, such as CVPR, ECCV, ACM Multimedia, and AAAI for multiple times.

Chen Change Loy (Senior Member, IEEE) received the PhD degree in computer science from the Queen Mary University of London in 2010. He is currently an associate professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU). He is also the co-associate director of S-Lab, NTU. From 2013 to 2018, he was as a research assistant professor with the Department of Information Engineering, The Chinese University of Hong Kong. His research interests include computer vision and deep learning. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *International Journal of Computer Vision.* He was the area chair of CVPR 2021, CVPR 2019, ECCV 2018, AAAI 2021, and BMVC 2018-2020.

Ping Luo received the PhD degree in information engineering from The Chinese University of Hong Kong (CUHK) in 2014, supervised by professor Xiaoou Tang and professor Xiaogang Wang. He is currently an assistant professor with the Department of Computer Science, The University of Hong Kong. From 2014 to 2016, he was a postdoctoral fellow with CUHK. From 2017 to 2018, he was a principal research scientist with SenseTime Research. His research interests include machine learning and computer vision. He has authored or coauthored more than 100 peer-reviewed articles in top-tier conferences and journals, such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, ICML, ICLR, CVPR, and NIPS. His work has high impact with 13,000 citations according to Google Scholar. He has won a number of competitions and awards, such as the First Runner-up in 2014 ImageNet ILSVRC Challenge, the First Place in 2017 DAVIS Challenge on Video Object Segmentation, a Gold medal in 2017 Youtube 8M Video Classification Challenge, the First Place in 2018 Drivable Area Segmentation Challenge for Autonomous Driving, the 2011 HK PhD Fellow Award, and the 2013 Microsoft Research Fellow Award (ten PhDs in Asia). He is named as one of the Young Innovators under 35 by MIT Technology Review (TR35) Asia Pacific.