INVITED PAPER

# Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications

*This article provides an overview of generative adversarial networks (GANs) with a special focus on algorithms and applications for visual synthesis.*

By Ming-Yu Liu, Xun Huang, Jiahui Yu, *Member IEEE*, Ting-Chun Wang, and Arun Mallya

**ABSTRACT** | The generative adversarial network (GAN) framework has emerged as a powerful tool for various image and video synthesis tasks, allowing the synthesis of visual content in an unconditional or input-conditional manner. It has enabled the generation of high-resolution photorealistic images and videos, a task that was challenging or impossible with prior methods. It has also led to the creation of many new applications in content creation. In this article, we provide an overview of GANs with a special focus on algorithms and applications for visual synthesis. We cover several important techniques to stabilize GAN training, which has a reputation for being notoriously difficult. We also discuss its applications to image translation, image processing, video synthesis, and neural rendering.

**KEYWORDS** | Computer vision; generative adversarial networks (GANs); image and video synthesis; image processing; neural rendering.

## I. INTRODUCTION

The generative adversarial network (GAN) framework is a deep learning architecture [59], [100] introduced by Goodfellow *et al.* [60]. It consists of two interacting neural networks—a generator network $G$ and a discriminator network $D$—which are trained jointly by playing a zero-sum game where the objective of the generator is to synthesize fake data that resembles real data and the objective of the discriminator is to distinguish between real and fake data. When the training is successful, the generator is an approximator of the underlying data generation mechanism in the sense that the distribution of the fake data converges to the real one. Due to the distribution matching capability, GANs have become a popular tool for various data synthesis and manipulation problems, especially in the visual domain.

GAN's rise also marks another major success of deep learning in replacing hand-designed components with machine-learned components in modern computer vision pipelines. As deep learning has directed the community to abandon hand-designed features, such as the histogram of oriented gradients (HOG) [36], for deep features computed by deep neural networks, the objective function used to train the networks remains largely hand-designed. While this is not a major issue for a classification task since effective and descriptive objective functions, such as the cross-entropy loss exist, this is a serious hurdle for a generation task. After all, how can one hand-design a function to guide a generator to produce a better cat image? How can we even mathematically describe "felineness" in an image?

GANs address the issue by deriving a functional form of the objective using training data. As the discriminator is trained to tell whether an input image is a cat image from the training data set or one synthesized by the generator, it defines an objective function that can guide the

Manuscript received August 4, 2020; revised November 26, 2020; accepted December 24, 2020. Date of publication February 1, 2021; date of current version April 30, 2021. *(All the authors contributed equally to this work.)* *(Corresponding author: Ming-Yu Liu.)*

**Ming-Yu Liu**, **Xun Huang**, **Ting-Chun Wang**, and **Arun Mallya** are with NVIDIA, Santa Clara, CA 95050-2519 USA (e-mail: mingyu.liu.tw@gmail.com). **Jiahui Yu** is with Google, Mountain View, CA 10011 USA.

Digital Object Identifier 10.1109/JPROC.2021.3049196

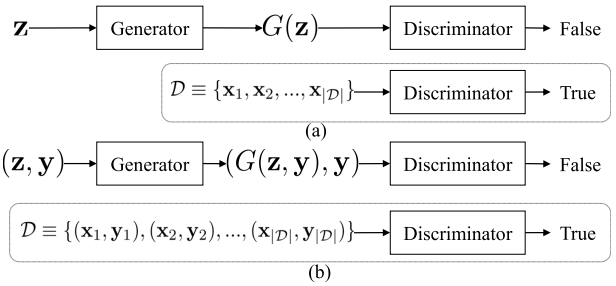



**Fig. 1.** *Unconditional versus conditional GANs. (a) In unconditional GANs, the generator converts a noise input $z$ to a fake image G($z$), where $z \sim \mathcal{Z}$, and $\mathcal{Z}$ is usually a Gaussian random variable. The discriminator tells apart real images $x$ from the training data set $\mathcal{D}$ and fake images from G. (b) In conditional GANs, the generator takes an additional input $y$ as the control signal, which could be another image (image-to-image translation), text (text-to-image synthesis), or a categorical label (label-to-image synthesis). The discriminator tells apart real from fake by leveraging the information in $y$. In both settings, the combination of the discriminator and real training data defines an objective function for image synthesis. This data-driven objective function definition is a powerful tool for many computer vision problems.*

generator in improving its generation based on its current network weights. The generator can keep improving as long as the discriminator can differentiate real and fake cat images. The only way that a generator can beat the discriminator is to produce images similar to the real images used for training. Since all the training images contain cats, the generator output must contain cats to win the game. Moreover, when we replace the cat images with dog images, we can use the same method to train a dog image generator. The objective function for the generator is defined by the training data set and the discriminator architecture. It is, thus, a very flexible framework to define the objective function for a generation task, as illustrated in Fig. 1.

However, despite its excellent modeling power, GANs are notoriously difficult to train because it involves chasing a moving target. Not only do we need to make sure that the generator can reach the target but also that the target can reach a desirable level of goodness. Recall that the goal of the discriminator is to differentiate real and fake data. As the generator changes, the fake data distribution also changes as well. This poses a new classification problem to the discriminator, distinguishing the same real but a new kind of fake data distribution, one that is presumably more similar to the real data distribution. As the discriminator is updated according to the new classification problem, it induces a new objective for the generator. Without careful control of the dynamics, a learning algorithm tends to experience failures in GAN training. Often, the discriminator becomes too strong and provides strong gradients that push the generator to a numerically unstable region. This is a well-recognized issue. Fortunately, over the years, various approaches, including better training algorithms, network architectures, and regularization techniques, have been proposed to stabilize GAN training. We will review several representative approaches in Section III. In Fig. 2, we illustrate the progress of GANs over the past few years.

In the original GAN formulation [60], the generator is formulated as a mapping function that converts a simple, unconditional distribution, such as a uniform distribution or a Gaussian distribution, to complex data distribution, such as a natural image distribution. We, now, generally refer to this formulation as the unconditional GAN framework. While the unconditional framework has several important applications on its own, the lack of controllability in the generation outputs makes it unfit for many applications. This has motivated the development of the conditional GAN framework. In the conditional framework, the generator additionally takes a control signal as input. The signal can take in many different forms, including category labels, texts, images, layouts, sounds, and even graphs. The goal of the generator is to produce outputs corresponding to the signal. In Fig. 1, we compare these two frameworks. This conditional GAN framework has led to many exciting applications. We will cover several representative ones through Sections IV–VII.

GANs have led to the creation of many exciting new applications. For example, it has been the core building block to semantic image synthesis algorithms that concern converting human-editable semantic representations, such as segmentation masks or sketches, to photorealistic images. GANs have also led to the development of many image-to-image translation methods that aim to translate an image in one domain to a corresponding image in a different domain. These methods find a wide range of applicability, ranging from image editing to domain adaptation. We will review some algorithms in Section IV.

We can, now, find GAN's footprint in many visual processing systems. For example, for image restoration, super-resolution (SR), and inpainting, where the goal is to transform an input image distribution to a target image distribution, GANs have been shown to generate results with much better visual quality than those produced with

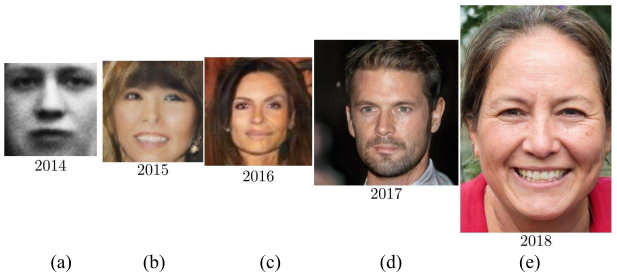



**Fig. 2.** *GAN progress on face synthesis. The progress of GANs on face synthesis over the years. From left to right, we have face synthesis results by (a) the original GAN [60], (b) DCGAN [163], (c) CoGAN [119], (d) PgGAN [84], and (e) StyleGAN [85]. This image was originally created and shared by Ian Goodfellow on Twitter.*

traditional methods. We will provide an overview of GAN methods in these image-processing tasks in Section V.

Video synthesis is another exciting area that GANs have shown promising results. Many research studies have utilized GANs to synthesize realistic human videos or transfer motions from one person to another for various entertainment applications, which we will review in Section VI. Finally, due to its great capability in generating photo-realistic images, GANs have played an important role in the development of neural rendering—using neural networks to boost the performance of the graphics rendering pipeline. We will cover GAN studies in Section VII.

## II. RELATED WORKS

Several GAN review articles exist, including the introductory article by Goodfellow [58]. The articles by Creswell *et al.* [35] and Pan *et al.* [154] summarize GAN methods prior to 2018. Wang *et al.* [223] provided a taxonomy of GANs. Our work differs from the prior studies in that we provide a more contemporary summary of GANs with a focus on image and video synthesis.

There are many different deep generative models or deep neural networks that model the generation process of some data. Besides GANs, other popular deep generative models include deep Boltzmann machines (DBMs), variational autoencoders (VAEs), deep autoregressive models (DARs), and normalizing flow models (NFMs). We compare these models in Fig. 3 and briefly review them in the following.

### A. Deep Boltzmann Machines

DBMs [45], [48], [68], [175] are energy-based models [101], which can be represented by undirected graphs. Let $\boldsymbol{x}$ denote the array of image pixels, often called visible nodes. Let $\boldsymbol{h}$ denote the hidden nodes. DBMs model the probability density function of data based on the Boltzmann (or Gibbs) distribution as

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{N(\boldsymbol{\theta})}\sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{x},\boldsymbol{h};\boldsymbol{\theta})) \tag{1}$$

where $E$ is an energy function modeling interactions of nodes in the graph, $N$ is the partition function, and $\boldsymbol{\theta}$ denotes the network parameters to be learned. Once a DBM is trained, a new image can be generated by applying Markov chain Monte Carlo (MCMC) sampling, ascending from a random configuration to one with high probability. While extensively expressive, the reliance on MCMC sampling on both training and generation makes DBMs scale poorly compared to other deep generative models since efficient MCMC sampling is itself a challenging problem, especially for large networks.

### B. Variational AutoEncoders

VAEs [93], [94], [168] are directed probabilistic graphic models, inspired by the Helmholtz machine [37]. They are also descendants of latent variable models, such as principal component analysis and autoencoders [18], which concerns representing high-dimensional data $\boldsymbol{x}$ using lower dimensional latent variables $\boldsymbol{z}$. In terms of structure, a VAE employs an inference model $q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\phi})$ and a generation model $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})p(\boldsymbol{z})$, where $p(\boldsymbol{z})$ is usually a Gaussian distribution, which we can easily sample from, and $q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\phi})$ approximates the posterior $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})$. Both the inference and generation models are implemented using feed-forward neural networks. VAE training is through maximizing the evidence lower bound (ELBO) of $\log p(\boldsymbol{x};\boldsymbol{\theta})$, and the nondifferentiability of the stochastic sampling is elegantly handled by the reparameterization trick [94]. One can also show that maximization of the ELBO is equivalent to minimizing the Kullback–Leibler (KL) divergence

$$KL\left(q(\boldsymbol{x})q(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\phi})||p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})\right) \tag{2}$$

where $q(x)$ is the empirical distribution of the data [94]. Once a VAE is trained, an image can be efficiently generated by first sampling $\boldsymbol{z}$ from the Gaussian prior $p(\boldsymbol{z})$ and then passing it through the feed-forward deep neural network $p(\boldsymbol{x}|\boldsymbol{z};\boldsymbol{\theta})$. VAEs are effective in learning useful latent representations [188]. However, they tend to generate blurry output images.

### C. Deep AutoRegressive Models

DARs [30], [153], [177], [207] are deep learning implementations of classical autoregressive models, which assumes an ordering to the random variables to be modeled and generates the variables sequentially based on the ordering. This induces a factorization form to the data distribution given by solving

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \prod_i p\left(x_i|\boldsymbol{x}_{<i};\boldsymbol{\theta}\right) \tag{3}$$

where $x_i$ are variables in $\boldsymbol{x}$, and $\boldsymbol{x}_{<i}$ are the union of the variables that are prior to $x_i$ based on the assumed ordering. DARs are conditional generative models where they generate a new portion of the signal based on what has been generated or observed so far. The learning is based on maximum likelihood learning

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\log p(x_i|\boldsymbol{x}_{<i};\boldsymbol{\theta})\right]. \tag{4}$$

DAR training is more stable compared to the other generative models. However, due to the recurrent nature, they are slow in inference. Also, while, for audio or text, a natural ordering of the variables can be determined based on the time dimension, such an ordering does not exist for images. One, hence, has to enforce an order prior that is an unnatural fit to the image grid.

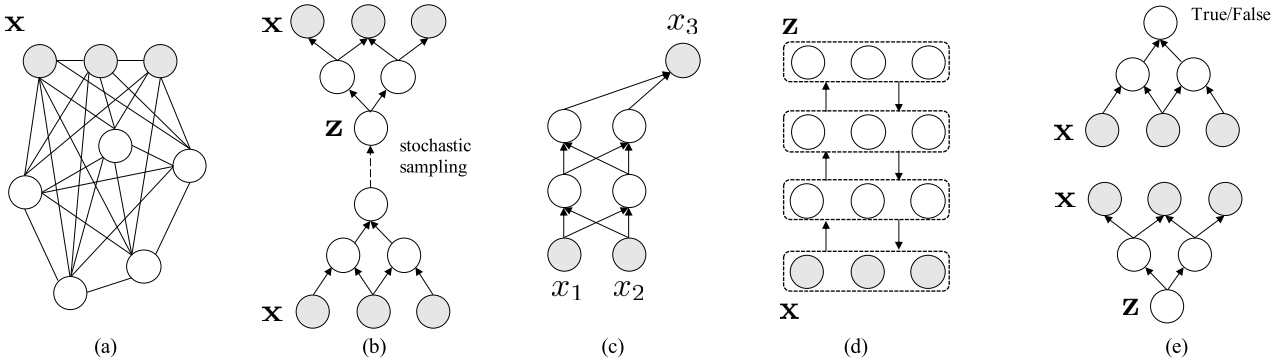



**Fig. 3.** ***Structure comparison of different deep generative models. Except for the DBM that is based on undirected graphs, other models are all based on directed graphs, which enjoys a faster inference speed. (a) Boltzmann machine. (b) VAE. (c) Autoregressive model. (d) NFM. (e) GAN.***

### D. Normalizing Flow Models

NFMs [40], [41], [92], [167] are based on the normalizing flow—a transformation of a simple probability distribution into a more complex distribution by a sequence of invertible and differentiable mappings. Each mapping corresponds to a layer in a deep neural network. With a layer design that guarantees invertibility and differentiability for all possible weights, one can stack many such layers to construct a powerful mapping because composition of invertible and differentiable functions is invertible and differentiable. Let $F = f^{(1)} \circ f^{(2)} \cdots \circ f^{(K)}$ be such a $K$-layer mapping that maps the simple probability distribution $Z$ to the data distribution $X$. The probability density of a sample $\boldsymbol{x} \sim X$ can be computed by transforming it back to the corresponding $\boldsymbol{z}$. Hence, we can apply maximum likelihood learning to train NFMs because the log-likelihood of the complex data distribution can be converted to the log-likelihood of the simple prior distribution subtracted by the Jacobians terms. This gives

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) = \log p(\boldsymbol{z};\boldsymbol{\theta}) - \sum_{i=1}^{K} \log \left| \det \frac{df^{(i)}}{d\boldsymbol{z}_{i-1}} \right| \tag{5}$$

where $\boldsymbol{z}_i = f^{(i)}(\boldsymbol{z}_{i-1})$. One key strength of NFMs is in supporting direct evaluation of probability density calculation. However, NFMs require an invertible mapping, which greatly limits the choices of applicable architectures.

## III. LEARNING

Let $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ be the learnable parameters in $G$ and $D$, respectively. GAN training is formulated as a minimax problem

$$\min_{\boldsymbol{\phi}} \max_{\boldsymbol{\theta}} V(\boldsymbol{\theta}, \boldsymbol{\phi}) \tag{6}$$

where $V$ is the utility function.

GAN training is challenging. Famous failure cases include mode collapse and mode dropping. In mode collapse, the generator is trapped to a certain local minimum where it only captures a small portion of the distribution. In mode dropping, the generator does not faithfully model the target distribution and misses some portion of it. Other common failure cases include checkerboard and water drop artifacts. In this article, we cover the basics of GAN training and some techniques invented to improve training stability.

### A. Learning Objective

The core idea in GAN training is to minimize the discrepancy between the true data distribution $p(\boldsymbol{x})$ and the fake data distribution $p(G(\boldsymbol{z};\boldsymbol{\theta}))$. As there are a variety of ways to measure the distance between two distributions, such as the Jensen–Shannon divergence, the KL divergence, and the integral probability metric, there are also a variety of GAN losses, including the saturated GAN loss [60], the nonsaturated GAN loss [60], the Wasserstein GAN loss [6], [64], the least-square GAN loss [134], the hinge GAN loss [112], [242], the f-divergence GAN loss [81], [150], and the relativistic GAN loss [80]. Empirically, the performance of a GAN loss depends on the application and the network architecture. As of the time of writing this survey article, there is no clear consensus on which one is absolutely better.

Here, we give a generic GAN learning objective formulation that subsumes several popular ones. For the discriminator update step, the learning objective is

$$\max_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\Big[f_D(D(\boldsymbol{x};\boldsymbol{\phi})\Big] + \mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}}\Big[f_G(D(G(\boldsymbol{z};\boldsymbol{\theta});\boldsymbol{\phi})\Big] \tag{7}$$

where $f_D$ and $f_G$ are the output layers that transform the results computed by the discriminator $D$ to the classification scores for the real and fake images, respectively. For the generator update step, the learning objective is

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}}\Big[g_G(D(G(\boldsymbol{z};\boldsymbol{\theta});\boldsymbol{\phi})\Big] \tag{8}$$

where $g_G$ is the output layer that transforms the result computed by the discriminator to a classification score for

**Table 1** Comparison of Different GAN Losses, Including Saturated [60], Nonsaturated [60], Wasserstein [6], Least Square [134], and Hinge [112], [242], in Terms of the Discriminator Output Layer Type in (7) and (8). We Maximize $f_D$ and $f_G$ for Training the Discriminator. As Shown in (7) and (8), We Minimize $g_G$ for Training the Generator. Note That $\sigma(x) = 1/1 + e^{-x}$ Is the Sigmoid Function

| Loss | $f_D(x)$ | $f_G(x)$ | $g_G(x)$ |
|---|---|---|---|
| Saturated | $\log\sigma(x)$ | $\log(1-\sigma(x))$ | $\log(1-\sigma(x))$ |
| Non-Saturated | $\log\sigma(x)$ | $\log(1-\sigma(x))$ | $-\log\sigma(x)$ |
| Wasserstein | $x$ | $-x$ | $-x$ |
| Least-Square | $-(x-1)^2$ | $-x^2$ | $(x-1)^2$ |
| Hinge | $\min(0, x-1)$ | $\min(0, -x-1)$ | $-x$ |

the fake image. In Table 1, we compare $f_D$, $f_G$, and $g_G$ for several popular GAN losses.

## B. Training

Two variants of stochastic gradient descent/ascent (SGD) schemes are commonly used for GAN training: the simultaneous update scheme and the alternating update scheme. Let $V_D(\boldsymbol{\theta}, \boldsymbol{\phi})$ and $V_G(\boldsymbol{\theta}, \boldsymbol{\phi})$ be the objective functions in (7) and (8), respectively. In the simultaneous update, each training iteration contains a discriminator update step and a generator update step given by

$$\boldsymbol{\phi}^{(t+1)} = \boldsymbol{\phi}^{(t)} + \alpha_D \frac{\partial V_D(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t)})}{\partial \boldsymbol{\phi}} \tag{9}$$

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha_G \frac{\partial V_G(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t)})}{\partial \boldsymbol{\theta}} \tag{10}$$

where $\alpha_D$ and $\alpha_G$ are the learning rates for the generator and discriminator, respectively. In the alternating update, each training iteration consists of one discriminator update step followed by a generator update step, given by

$$\boldsymbol{\phi}^{(t+1)} = \boldsymbol{\phi}^{(t)} + \alpha_D \frac{\partial V_D(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t)})}{\partial \boldsymbol{\phi}} \tag{11}$$

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha_G \frac{\partial V_G(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t+1)})}{\partial \boldsymbol{\theta}}. \tag{12}$$

Note that in the alternating update scheme, the generator update (12) utilizes the newly updated discriminator parameters $\boldsymbol{\theta}^{(t+1)}$, while, in the simultaneous update (10), it does not. These two schemes have their pros and cons. The simultaneous update scheme can be computed more efficiently, as a major part of the computation in the two steps can be shared. On the other hand, the alternating update scheme tends to be more stable as the generator update is computed based on the latest discriminator. Recent GAN studies [24], [64], [70], [118], [156] mostly use the alternating update scheme. Sometimes, the discriminator update (11) is performed several times before computing (12) [24], [64].

Among various SGD algorithms, ADAM [91], which is based on adaptive estimates of the first- and second-order moments, is very popular for training GANs. ADAM has several user-defined parameters. Typically, the first momentum is set to 0, while the second momentum is set to 0.999. The learning rate for the discriminator update is often set to two to four times larger than the learning rate for the generator update (usually set to 0.0001), which is called the two-time update scales (TTUR) [67]. We also note that RMSProp [201] is popular for GAN training [64], [84], [85], [118].

## C. Regularization

We review several popular regularization techniques available for countering instability in GAN training.

**Gradient penalty (GP)** is an auxiliary loss term that penalizes deviation of gradient norm from the desired value [64], [138], [169]. To use GP, one adds it to the objective function for the discriminator update, that is, (7). There are several variants of GP. Generally, they can be expressed as

$$\text{GP-}\delta = \mathbb{E}_{\hat{\boldsymbol{x}}}\Big[\, \|\nabla D(\hat{\boldsymbol{x}})\|_2 - \delta \Big]. \tag{13}$$

The two most common forms are GP-1 [64] and GP-0 [138].

GP-1 was first introduced by Gulrajani *et al.* [64]. It uses an imaginary data distribution

$$\hat{\boldsymbol{x}} = u\boldsymbol{x} + (1-u)G(\boldsymbol{z}), \quad u \sim \mathcal{U}(0,1) \tag{14}$$

where $u$ is a uniform random variable between 0 and 1. Basically, $\hat{\boldsymbol{x}}$ is neither real nor fake. It is a convex combination of a real sample and a fake sample. The design of the GP-1 is motivated by the property of an optimal $D$ that solves the Wasserstein GAN loss. However, GP-1 is also useful when using other GAN losses. In practice, it has the effect of countering vanishing and exploding gradients that occurred during GAN training.

On the other hand, the design of GP-0 is based on the idea of penalizing the discriminator deviating away from the Nash equilibrium. GP-0 takes a simpler form where they do not use imaginary sample distribution but use the real data distribution, that is, setting $\hat{\boldsymbol{x}} \equiv \boldsymbol{x}$. We find the use of GP-0 in several state-of-the-art GAN algorithms [85], [86].

**Spectral normalization (SN)** [140] is an effective regularization technique used in many recent GAN algorithms [24], [156], [170], [242]. SN is based on regularizing the spectral norm of the projection operation at each layer of the discriminator, by simply dividing the weight matrix by its largest eigenvalue. Let $W$ be the weight matrix of a layer of the discriminator network. With SN, the true weight that is applied is

$$W/\sqrt{\lambda_{\max}(W^T W)} \tag{15}$$

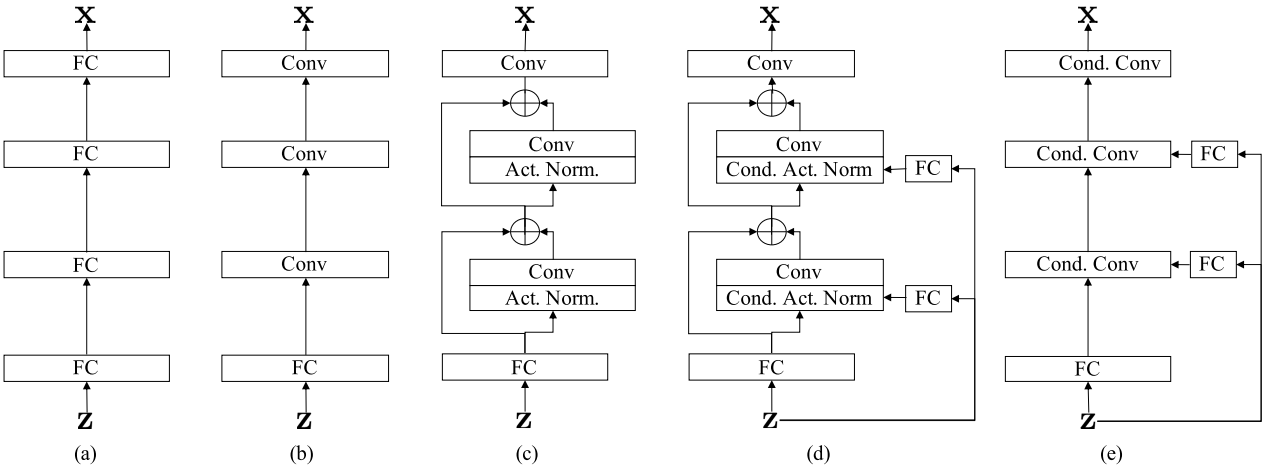



**Fig. 4.** ***Generator evolution. Since the debut of GANs [60], the generator architecture has continuously evolved. (a)–(c) One can observe the change from simple MLPs to deep convolutional and residual networks. Recently, conditional architectures, including (d) conditional ANs and (e) conditional convolutions, have gained popularity as they allow users to have more control on the generation outputs. (a) MLP. (b) Deep ConvNet. (c) Residual Net. (d) Residual Net. with Cond. Act. Norm. (e) Cond. ConvNet.***

where $\lambda_{\max}(A)$ extracts the largest eigenvalue from the square matrix $A$. In other words, each project layer has a projection matrix with a spectral norm equal to 1.

**Feature matching (FM)** provides a way to encourage the generator to generate images similar to real ones in some sense. Similar to GP, FM is an auxiliary loss. There are two popular implementations: one is batch-based [176], and the other is instance-based [99], [218]. Let $D^i$ be the $i$th layer of a discriminator $D$, that is, $D = D^d \circ \cdots \circ D^2 \circ D^1$. For the batch-based FM loss, it matches the moments of the activations extracted by the real and fake images, respectively. For the $i$th layer, the loss is

$$\left\| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[D^i \circ \cdots \circ D^1(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{Z}}[D^i \circ \cdots \circ D^1(G(\boldsymbol{z}))] \right\|. \tag{16}$$

One can apply the FM loss to a subset of layers in the generator and use the weighted sum as the final FM loss. The instance-based FM loss is only applicable to conditional generation models where we have the corresponding real image for a fake image. For the $i$th layer, the instance-based FM loss is given by

$$\left\| [D^i \circ \cdots \circ D^1(\boldsymbol{x}_i)] - [D^i \circ \cdots \circ D^1(G(\boldsymbol{z}, \boldsymbol{y}_i))] \right\| \tag{17}$$

where $\boldsymbol{y}_i$ is the control signal for $\boldsymbol{x}_i$.

*Perceptual loss [79]:* Often, when instance-based FM loss is applicable, one can additionally match features extracted from real and fake images using a pretrained network. Such a variant of FM losses is called the perceptual loss [79].

**Model average (MA)** can improve the quality of images generated by a GAN. To use MA, we keep two copies of the generator network during training, where one is the original generator with weight $\boldsymbol{\theta}$ and the other is the MA generator with weight $\boldsymbol{\theta}_{MA}$. At iteration $t$, we update $\boldsymbol{\theta}_{MA}$ based on

$$\boldsymbol{\theta}_{MA}^{(t)} = \beta\boldsymbol{\theta}^{(t)} + (1-\beta)\boldsymbol{\theta}_{MA}^{(t-1)} \tag{18}$$

where $\beta$ is a scalar controlling the contribution from the current model weight.

## D. Network Architecture

Network architectures provide a convenient way to inject inductive biases. Certain network designs often work better than others for a given task. Since the introduction of GANs, we have observed an evolution of the network architecture for both the generator and discriminator.

*1) Generator Evolution:* In Fig. 4, we visualize the evolution of the GAN generator architecture. In the original GAN paper [60], both the generator and the discriminator are based on the multilayer perceptron (MLP) [see Fig. 4(a)]. As an MLP fails to model the translational invariance property of natural images, its output images are of limited quality. In the DCGAN work [163], deep convolutional architecture [see Fig. 4(b)] is used for the GAN generator. As the convolutional architecture is a better fit for modeling image signals, the outputs produced by the DCGAN are often of better quality. Researchers also borrow architecture designs from discriminative modeling tasks. As the residual architecture [66] is proved to be effective for training deep networks, several GAN studies start to use the residual architecture in their generator design [see Fig. 4(c)] [6], [140].

A residual block used in modern GAN generators typically consists of a skip connection paired with a series of batch normalization (BN) [74], nonlinearity, and convolution operations. The BN is one type of activation norm (AN), a technique that normalizes the activation values

to facilitate training. Other AN variants have also been exploited for the GAN generator, including the instance normalization [206], the layer normalization [8], and the group normalization [228]. Generally, an activation normalization scheme consists of a whitening step followed by an affine transformation step. Let $\boldsymbol{h}_c$ be the output of the whitening step for $\boldsymbol{h}$. The final output of the normalization layer is

$$\gamma_c \boldsymbol{h}_c + \beta_c \tag{19}$$

where $\gamma_c$ and $\beta_c$ are scalars used to shift the postnormalization activation values. They are constants learned during training.

For many applications, it is required to have some way to control the output produced by a generator. This desire has motivated various conditional generator architectures [see Fig. 4(d)] for the GAN generator [24], [70], [156]. The most common approach is to use the conditional AN. In a conditional AN, both $\gamma_c$ and $\beta_c$ are data-dependent. Often, one employs a separate network to map input control signals to the target $\gamma_c$ and $\beta_c$ values. Another way to achieve such controllability is to use hypernetworks, basically, using an auxiliary network to produce weights for the main network. For example, we can have a convolutional layer where the filter weights are generated by a separate network. We often call such a scheme conditional convolutions [see Fig. 4(e)], and it has been used for several state-of-the-art GAN generators [86], [216].

*2) Discriminator Evolution:* GAN discriminators have also undergone an evolution. However, the change has mostly been on moving from the MLP to deep convolutional and residual architectures. As the discriminator is solving a classification task, new breakthroughs in architecture design for image classification tasks could influence future GAN discriminator designs.

*3) Conditional Discriminator Architecture:* There are several effective architectures for utilizing control signals (conditional inputs $\boldsymbol{y}$) in the GAN discriminator to achieve better image generation quality, as visualized in Fig. 5. This includes the auxiliary classifier (AC) [151], input concatenation (IC) [75], and the projection discriminator (PD) [141]. The AC and PD are mostly used for category-conditional image generation tasks, while the PD is common for image-to-image translation tasks.

*4) Neural Architecture Search:* As neural architecture search has become a popular topic for various recognition tasks, efforts have been made in trying to automatically find a performant architecture for GANs [56].

While this section and Section III have focused on introducing the GAN mechanism and various algorithms used to train them, Sections IV–VII focus on various applications of GANs in generating images and videos.

## IV. IMAGE TRANSLATION

This section discusses the application of GANs to image-to-image translation, which aims to map an image from one domain to a corresponding image in a different domain, for example, sketch to shoes, label maps to photos, and summer to winter. The problem can be studied in a supervised setting, where sample pairs of corresponding images are available, or an unsupervised setting, where such training data are unavailable, and we only have two independent sets of images. In Sections IV-A and B, we will discuss recent progress in both settings.

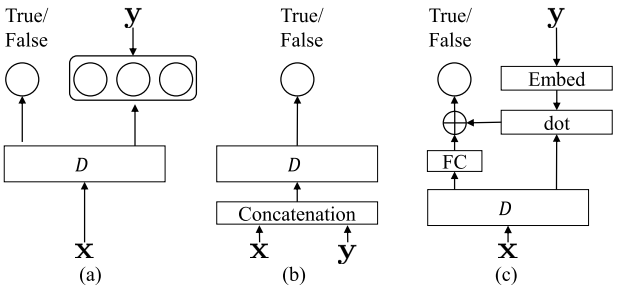



**Fig. 5.** ***Conditional discriminator architectures. There are several ways to leverage the user input signal $y$ in the GAN discriminator. (a) AC [151]. In this design, the discriminator is asked to predict the ground-truth label for the real image. (b) Input concatenation [75]. In this design, the discriminator learns to reason whether the input is real by learning a joint feature embedding of image and label. (c) PD [141]. In this design, the discriminator computes an image embedding and correlates it with the label embedding (through the dot product) to determine whether the input is real or fake.***

### A. Supervised Image Translation

Isola *et al.* [75] proposed the pix2pix framework as a general-purpose solution to image-to-image translation in the supervised setting. The training objective of pix2pix combines conditional GANs with the pixelwise $\ell_1$ loss between the generated image and the ground truth. One notable design choice of pix2pix is the use of patchwise discriminators (PatchGAN), which attempts to discriminate each local image patch rather than the whole image. This design incorporates the prior knowledge that the underlying image translation function we want to learn is local, assuming independence between pixels that are far away. In other words, translation mostly involves style or texture changes. It significantly alleviates the burden of the discriminator because it requires much less model capacity to discriminate local patches than whole images.

One important limitation of pix2pix is that its translation function is restricted to be one-to-one. However, many of the mappings that we aim to learn are one-to-many in nature. In other words, the distribution of possible outputs is multimodal. For example, one can imagine many shoes in different colors and styles that correspond to the same sketch of a shoe. Naively injecting a Gaussian noise latent code to the generator does not lead to many variations since the generator is free to ignore that latent code. BicycleGAN [255] explores approaches to encourage the generator to make use of the latent code to represent output variations, including applying a KL divergence loss

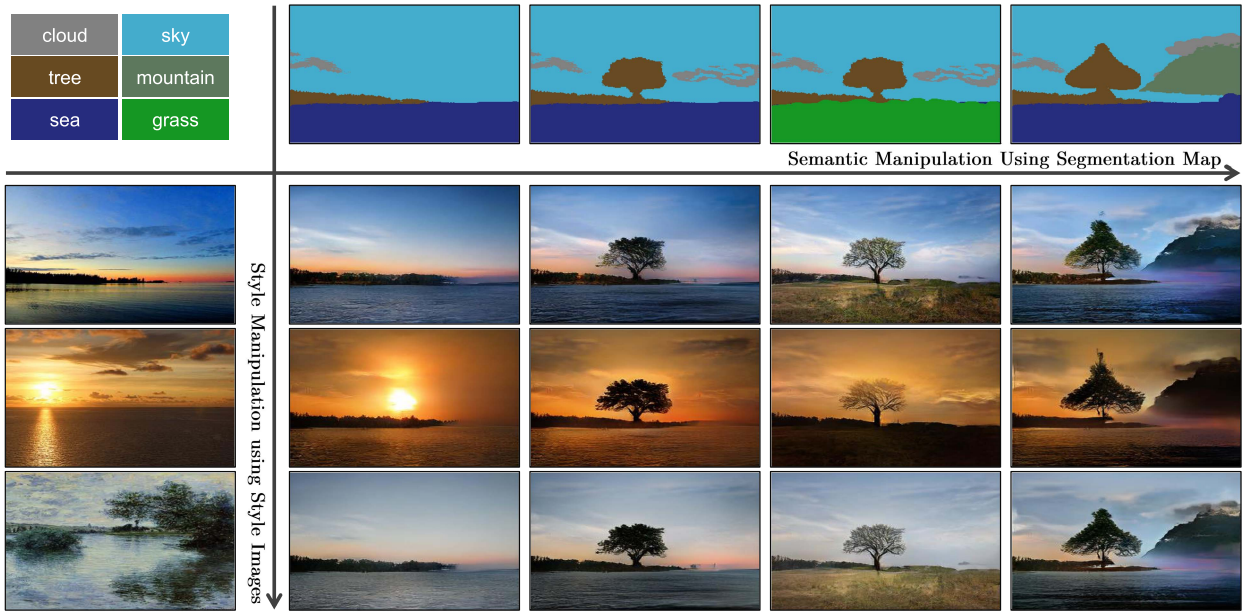



**Fig. 6.** ***Image translation examples of SPADE [156], which converts semantic label maps into photorealistic natural scenes. The style of the output image can also be controlled by a reference image (the leftmost column). Images are from Park et al. [156].***

to the encoded latent code and reconstructing the sampled latent code from the generated image. Other strategies to encourage diversity include using different generators to capture different output modes [54], replacing the reconstruction loss with maximum likelihood objective [106], [107], and directly encouraging the distance between output images generated from different latent codes to be large [120], [133], [234].

Besides, the quality of image-to-image translation has been significantly improved by some recent studies [104], [122], [156], [194], [218], [250]. In particular, pix2pixHD [218] is able to generate high-resolution (HR) images with a coarse-to-fine generator and a multiscale discriminator. SPADE [156] further improves the image quality with a spatially adaptive normalization layer. SPADE, in addition, allows a style image input for better control of the desired look of the output image. Some examples of SPADE are shown in Fig. 6.

## B. Unsupervised Image Translation

For many tasks, paired training images are very difficult to obtain [16], [32], [70], [90], [105], [117], [235], [254]. Unsupervised learning of mappings between corresponding images in two domains is a much harder problem but has wider applications than the supervised setting. CycleGAN [254] simultaneously learns mappings in both directions and employs a cycle consistency loss to enforce that, if an image is translated to the other domain and translated back to the original domain, the output should be close to the original image. UNIT [117] makes a shared latent space assumption [119] that a pair of corresponding images can be mapped to the same latent code in a shared latent space. It is shown that shared-latent space implies cycle consistency and imposes a stronger regularization. DistanceGAN [16] encourages the mapping to preserve the distance between any pair of images before and after translation. While the methods above need to train a different model for each pair of image domains, StarGAN [32] is able to translate images across multiple domains using only a single model.

In many unsupervised image translation tasks (e.g., horses to zebras and dogs to cats), the two-image domains mainly differ in the foreground objects, and the background distribution is very similar. Ideally, the model should only modify the foreground objects and leave the background region untouched. Some work [31], [137], [232] employs spatial attention to detect and change the foreground region without influencing the background. InstaGAN [142] further allows the shape of the foreground objects to be changed.

The early work mentioned above focuses on unimodal translation. On the other hand, recent advances [5], [57], [70], [105], [128], [133] have made it possible to perform multimodal translation, generating diverse output images given the same input. For example, MUNIT [70] assumes that images can be encoded into two disentangled latent spaces: a domain-invariant content space that captures the information that should be preserved during translation, and a domain-specific style space that represents the variations that are not specified by the input image. To generate diverse translation results, we can recombine the content code of the input image with different style codes sampled from the style space of the target domain. Fig. 7 compares MUNIT with existing unimodal translation

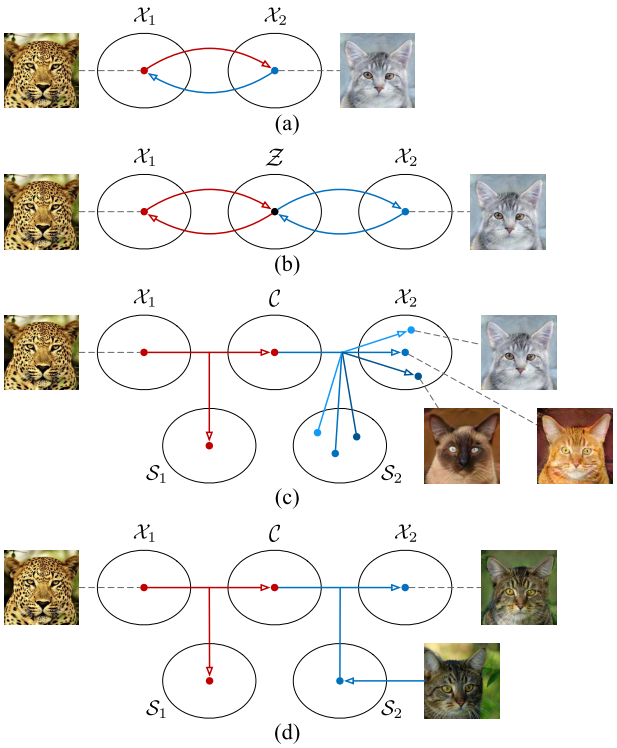



**Fig. 7.** ***Comparisons among unsupervised image translation methods (CycleGAN [254], UNIT [117], and MUNIT [70]). $\mathcal{X}_1$ and $\mathcal{X}_2$ are two different image domains (dogs and cats in this example). (a) CycleGAN enforces the learned mappings to be inverses of each other. (b) UNIT autoencodes images in both domains to a common latent space $\mathcal{Z}$. Both CycleGAN and UNIT can only perform unimodal translation. (c) MUNIT (randomly sampled) decomposes the latent space into a shared content space $\mathcal{C}$ and unshared style spaces $\mathcal{S}_1, \mathcal{S}_2$. Diverse outputs can be obtained by sampling different style codes from the target style space. (d) Style of the translation output can also be controlled by a guiding image in the target domain [MUNIT (example-guided)].***

methods including CycleGAN and UNIT. The disentangled latent space not only enables multimodal translation but also allows example-guided translation in which the generator recombines the domain-invariant content of an image from the source domain and the domain-specific style of an image from the target domain. The idea of using a guiding style image has also been applied to the supervised setting [156], [214], [244].

Although paired example images are not needed in the unsupervised setting, most existing methods still require access to a large number of unpaired example images in both source and target domains. Some studies seek to reduce the number of training examples without much loss of performance. Benaim and Wolf [17] focused on the situation where there are many images in the target domain but only a single image in the source domain. The work of Cohen and Wolf [34] enables translation in the opposite direction where the source domain has many images, but the target domain has only one. The above setting assumes that the source and target domain images, whether there are many or few, are available during training. Liu *et al.* [118] proposed FUNIT to address a different situation where there are many source domain images that are available during training but few target domain images that are available only at test time. The target domain images are used to guide translation similar to the example-guided translation procedure in MUNIT. Saito *et al.* [170] proposed a content-conditioned style encoder to better preserve the domain-invariant content of the input image. However, the above scenario [118], [170] still assumes access to the domain labels of the training images. Some recent work aims to reduce the need for such supervision by using few [221] or even no [9] domain labels. Very recently, some studies [15], [113], [155] are able to achieve image translation even when each domain only has a single image, inspired by recent advances that can train GANs on a single image [179].

Despite the empirical successes, the problem of unsupervised image-to-image translation is inherently ill-posed, even with constraints such as cycle consistency or shared latent space. Specifically, there exist infinitely many mappings that satisfy those constraints [38], [51], [231], yet most of them are not semantically meaningful. How do current methods successfully find meaningful mapping in practice? Galanti *et al.* [51] assume that meaningful mapping is of minimal complexity, and the popular generator architectures are not expressive enough to represent mappings that are highly complex. Bézenac *et al.* [38] further argued that the popular architectures are implicitly biased toward mappings that produce minimal changes to the input, which are usually semantically meaningful. In summary, the training objectives of unsupervised image translation alone cannot guarantee that the model can find semantically meaningful mappings and the inductive bias of generator architectures plays an important role.

## V. IMAGE PROCESSING

GAN's strength in generating realistic images makes it ideal for solving various image-processing problems, especially for those where the perceptual quality of image outputs is the primary evaluation criterion. This section will discuss some prominent GAN-based methods for several key image-processing problems, including image restoration and enhancement (SR, denoising, deblurring, and compression artifacts removal) and image inpainting.

### A. Image Restoration and Enhancement

The traditional way of evaluating algorithms for image restoration and enhancement tasks is to measure the distortion, the difference between the ground-truth images and restored images using metrics, such as the mean square error (MSE), the peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM). Recently, metrics for measuring perceptual quality, such as the no-reference (NR) metric [127], have been proposed, as the visual quality is arguably the most important factor for

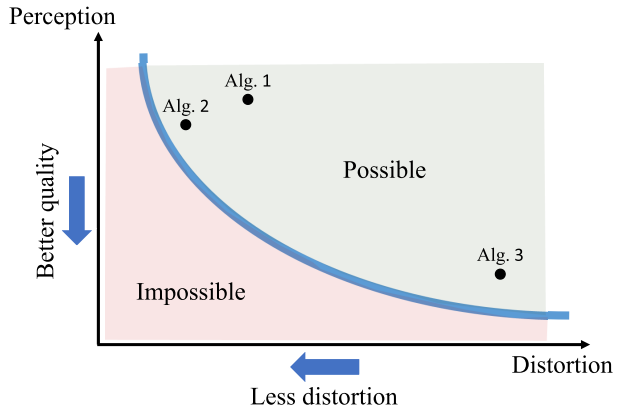



**Fig. 8.** ***Perception–distortion tradeoff [22]. Distortion metrics, including the MSE, PSNR, and SSIM, measure the similarity between the ground-truth image and the restored images. Perceptual quality metrics, including NR [127], measure the distribution distance between the recovered image distribution and the target image distribution. Blau and Michaeli [22] showed that an image restoration algorithm can be characterized by the distortion and perceptual quality tradeoff curve. The plot is from Blau and Michaeli [22].***

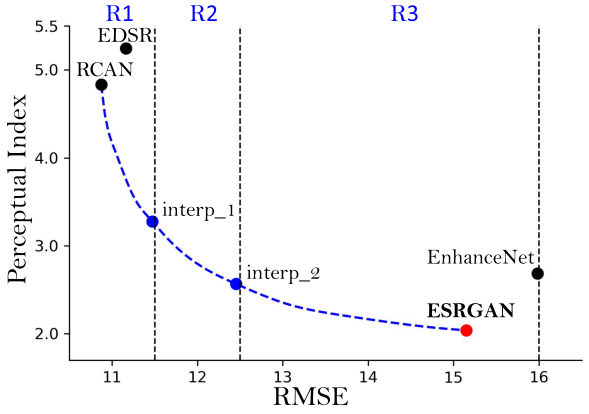



**Fig. 9.** ***Perception–distortion curve of the ESRGAN [220] on PIRM self-validation data set [21]. The curve also compares the ESRGAN with the EnhanceNet [174], the RCAN [246], and the EDSR [111]. The curve is from Wang et al. [220].***

the usability of an algorithm. Blau and Michaeli [22] proposed the perception–distortion tradeoff [22], which states that an image restoration algorithm can potentially improve only in terms of its distortion or in terms of its perceptual quality, as shown in Fig. 8. Blau and Michaeli [22] further demonstrated that GANs provide a principled way to approach the perception–distortion bound.

**Image SR** aims at estimating an HR image from its low-resolution (LR) counterpart. Deep learning has enabled faster and more accurate SR methods, including SRCNN [42], FSRCNN [43], ESPCN [182], VDSR [88], SRResNet [102], EDSR [111], SRDenseNet [203], MemNet [193], RDN [247], WDSR [236], and many others. However, the above SR approaches focus on improving the distortion metrics and pay little to no attention to the perceptual quality metrics. As a result, they tend to predict oversmoothed outputs and fail to synthesize finer high-frequency details.

Recent image SR algorithms improve the perceptual quality of outputs by leveraging GANs. The SRGAN [102] is the first of its kind and can generate photorealistic images with $4\times$ or higher upscaling factors. The quality of the SRGAN [102] outputs is mainly measured by the mean opinion score (MOS) over 26 raters. To enhance the visual quality further, Wang *et al.* [220] revisited the design of the three key components in the SRGAN: the network architecture, the GAN loss, and the perceptual loss. They proposed the enhanced SRGAN (ESRGAN), which achieves consistently better visual quality with more realistic and natural textures than the competing methods, as shown in Figs. 9 and 10. The ESRGAN is the winner of the 2018 Perceptual Image Restoration and Manipulation (PIRM) challenge [21] (region 3 in Fig. 9). Other GAN-based image SR methods and practices can be found in the 2018 PIRM challenge report [21].

The above image SR algorithms all operate in the supervised setting where they assume corresponding LR and HR pairs in the training data set. Typically, they create such a training data set by downsampling the ground-truth HR images. However, the downsampled HR images are very different from the LR images captured by a real sensor, which often contains noise and other distortion. As a result, these SR algorithms are not directly applicable to upsample LR images captured in the wild. Several methods have addressed the issue by studying image SR in the unsupervised setting where they only assume a data set of LR images captured by a sensor and a data set of HR images. Recently, Maeda [131] proposed a GAN-based image SR algorithm that operates in the unsupervised setting for bridging the gap.

**Image denoising** aims at removing noise from noisy images. The task is challenging since the noise distribution is usually unknown. This setting is also referred to as blind image denoising. DnCNN [243] is one of the

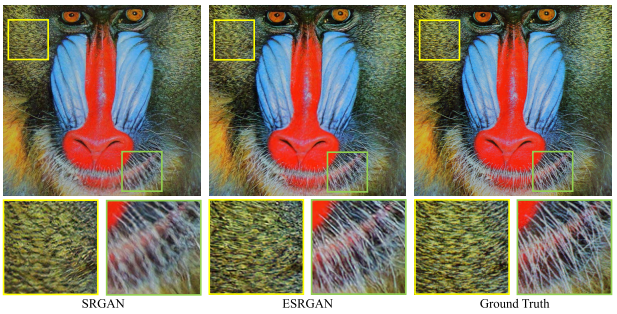



**Fig. 10.** ***Visual comparison between the ESRGAN [220] and the SRGAN [102]. Images are from Wang et al. [220].***

first approaches using feed-forward convolutional neural networks for image denoising. However, DnCNN [243] requires knowing the noise distribution in the noisy image and, hence, has limited applicability. To tackle blind image denoising, Chen *et al.* [27] proposed the GAN–CNN-based Blind Denoiser (GCBD) that consists of: 1) a GAN trained to estimate the noise distribution over the input noisy images to generate noise samples and 2) a deep CNN that learns to denoise on generated noisy images. The GAN training criterion of GCBD [27] is based on Wasserstein GAN [6], and the generator network is based on DCGAN [163].

**Image deblurring** sharpens blurry images that result from motion blur, out of focus, and possibly other causes. DeblurGAN [95] trains an image motion deblurring network using Wasserstein GAN [6] with the GP-1 loss and the perceptual loss (see Section III). Shen *et al.* [181] used a similar approach to deblur face image by using GAN and perceptual loss and incrementally training the deblurring network. Visual examples are shown in Fig. 11.

**Lossy image compression** algorithms (e.g., JPEG, JPEG2000, BPG, and WebP) can efficiently reduce image sizes but introduce visual artifacts in compressed images when the compression ratio is high. Deep neural networks have been widely explored for removing the introduced artifacts [4], [52], [204]. Galteri *et al.* [52] showed that a residual network trained with a GAN loss is able to produce images with more photorealistic details than MSE- or SSIM-based objectives for the removal of image compression artifacts. Tschannen *et al.* [204] further proposed distribution-preserving lossy compression using a new combination of Wasserstein GAN and Wasserstein autoencoder [202]. More recently, Agustsson *et al.* [4] built an extreme image compression system using unconditional and conditional GANs, outperforming all other codecs in the low bit-rate setting. Some compression visual examples [4] are shown in Fig. 12.

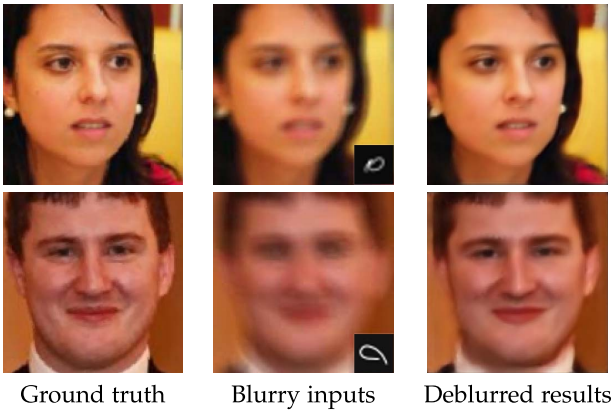



**Fig. 11.** *Face deblurring results with GANs [181]. Images are from Shen et al. [181].*

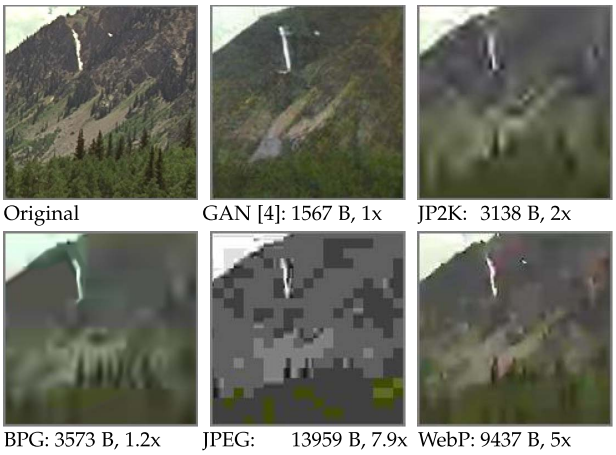



**Fig. 12.** *Image compression with GANs [4]. Comparing a GAN-based approach [4] for image compression to those obtained by the off-the-shelf codecs. Even with fewer than half the number of bytes, GAN-based compression [4] produces more realistic visual results. Images are from Agustsson et al. [4].*

## B. Image Inpainting

Image inpainting aims at filling missing pixels in an image such that the result is visually realistic and semantically correct. Image inpainting algorithms can be used to remove distracting objects or retouch undesired regions in photos and can be further extended to other tasks, including image uncropping, rotation, stitching, retargeting, recomposition, compression, SR, harmonization, and more.

Traditionally, patch-based approaches, such as the PatchMatch [12], copy background patches according to the low-level FM (e.g., Euclidean distance on pixel RGB values) and paste them into the missing regions. These approaches can synthesize plausible stationary textures but fail at nonstationary image regions, such as faces, objects, and complicated scenes. Recently, deep learning and GAN-based approaches [73], [78], [114], [145], [222], [229], [237], [238], [241], [249] open a new direction for image inpainting using deep neural networks learned on large-scale data in an end-to-end fashion. Compared with PatchMatch, these methods are more scalable and can leverage large-scale data.

The context encoder (CE) approach [157] is one of the first in using a GAN generator to predict the missing regions and is trained with the $\ell_2$ pixelwise reconstruction loss and a GAN loss. Iizuka *et al.* [73] further improved the GAN-based inpainting framework using both global and local GAN discriminators, with the global one operating on the entire image and the local one operating on only the patch in the hole. We note that the postprocessing techniques, such as image blending, are still required in these GAN-based approaches [73], [157] to reduce visual artifacts near the hole boundaries.

Yu *et al.* [237] proposed DeepFill, a GAN framework for end-to-end image inpainting without any postprocessing

**Fig. 13.** *Image inpainting results using the DeepFill [237]. Missing regions are shown in white. In each pair, the left is the input image, and the right is the direct output of trained GAN without any postprocessing. Images are from Yu et al. [237].*

step, which leverages a stacked network, consisting of a coarse network and a refinement network, to ensure the color and texture consistency between the in-filled regions and their surroundings. Moreover, as convolutions are local operators and less effective in capturing long-range spatial dependencies, the contextual attention layer [237] is introduced and integrated into the DeepFill to borrow information from distant spatial locations explicitly. Visual examples of the DeepFill [237] are shown in Fig. 13.

One common issue with the earlier GAN-based inpainting approaches [73], [157], [237] is that the training is performed with randomly sampled rectangular masks. While allowing easy processing during training, these approaches do not generalize well to free-form masks, that is, irregular masks with arbitrary shapes. To address the issue, Liu *et al.* [114] proposed the partial convolution layer where the convolution is masked and renormalized to utilize valid pixels only. Yu *et al.* [238] further proposed the gated convolution layer, generalizing the partial convolution by providing a learnable dynamic feature selection mechanism for each channel at each spatial location across all layers. In addition, as free-form masks may appear anywhere in images with any shape, global and local GANs [73] designed for a single rectangular mask are not applicable. To address this issue, Yu *et al.* [238] introduced a patch-based GAN loss, SNPatchGAN [238], by applying spectral-normalized discriminator on the dense image patches. Visual examples of the DeepFillV2 [238] with free-form masks are shown in Fig. 14.

Although capable of handling free-form masks, these inpainting methods perform poorly in reconstructing foreground details. This motivated the design of edge-guided image inpainting methods [145], [229]. These methods decompose inpainting into two stages. The first stage predicts edges or contours of foregrounds, and the second stage takes predicted edges to predict the final output. Moreover, for image inpainting, enabling user interactivity is essential as there are many plausible solutions for filling a hole in an image. User-guided inpainting methods [145], [229], [238] have been proposed to provide an option to take additional user inputs, for example, sketches, as guidance for image inpainting networks. An example of user-guided image inpainting is shown in Fig. 15.

**Fig. 14.** *Free-form image inpainting results using the DeepFillV2 [238]. From left to right, we have the ground-truth image, the free-form mask, and the DeepFillV2 inpainting result. Original images are from Yu et al. [238].*

Finally, we note that the image out-painting or extrapolation tasks are closely related to image inpainting [89], [195]. They can also be benefited from a GAN formulation.

## VI. VIDEO SYNTHESIS

Video synthesis focuses on generating video content instead of static images. Compared with image synthesis, video synthesis needs to ensure the temporal consistency of the output videos. This is usually achieved by using a temporal discriminator [205], flow-warping loss on neighboring frames [217], smoothing the inputs before processing [26], or a postprocessing step [98]. Each of them might be suitable for a particular task.

Similar to image synthesis, video synthesis can be classified into unconditional and conditional video synthesis. Unconditional video synthesis generates sequences using random noise inputs [33], [171], [205], [210]. Because such a method needs to model all the spatial and temporal content in a video, the generated results are often short or with very constrained motion patterns. For example, MoCoGAN [205] decomposes the motion and content parts of the sequence and uses a fixed latent code for the content and a series of latent codes to generate the motion.

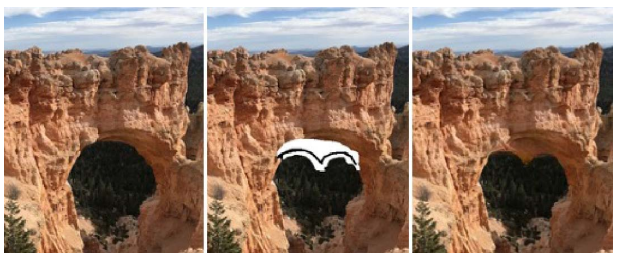


**Fig. 15.** *User-guided image inpainting results using the DeepFillV2 [238]. From left to right, we have the ground-truth image, the mask with user-provided edge guidance, and the DeepFillV2 inpainting result. Images are from Yu et al. [238].*


**Fig. 16.** *Face swapping versus reenactment [149]. Face swapping focuses on pasting the face region from one subject to another, while face reenactment concerns transferring the expressions and head poses from the target subject to the source image. Images are from Nirkin et al. [149].*

The synthesized videos are usually up to a few seconds on simple video content, such as facial motion.

On the other hand, conditional video synthesis generates videos conditioning on input content. A common category is future frame prediction [39], [47], [83], [103], [110], [125], [136], [191], [208], [212], [213], [230], which attempts to predict the next frame of a sequence based on the past frames. Another common category of conditional video synthesis is conditioning on an input video that shares the same high-level representation. Such a setting is often referred to as the video-to-video synthesis [217]. This line of studies has shown promising results on various tasks, such as transforming high-level representations to photorealistic videos [217], animating characters with new expressions or motions [26], [199], or innovating a new rendering pipeline for graphics engines [50]. Due to its broader impact, we will mainly focus on conditional video synthesis. Particularly, we will focus on its two major domains: face reenactment and pose transfer.

## A. Face Reenactment

Conditional face video synthesis exists in many forms. The most common forms include face swapping and face reenactment. Face swapping focuses on pasting the face region from one subject to another, whereas face reenactment concerns transferring the subject's expressions and head poses. Fig. 16 illustrates the difference. Here, we only focus on face reenactment. It has many applications in fields, such as gaming or the film industry, where the characters can be animated by human actors. Based on whether the trained model can only work for a specific person or is universal to all persons, face reenactment can be classified as subject-specific or subject-agnostic, as described in the following.

*1) Subject-Specific Model:* Traditional methods usually build a subject-specific model, which can only synthesize one predetermined subject by focusing on transferring the expressions without transferring the head movement [192], [197]–[199], [209]. This line of studies usually starts by collecting footage of the target person to be synthesized, either using an RGBD sensor [198] or an RGB sensor [199]. Then, a 3-D model of the target person is built for the face region [20]. At test time, given the new expressions, they can be used to drive the 3-D model to generate the desired motions, as shown in Fig. 17. Instead of extracting the driving expressions from someone else, they can also be directly synthesized from speech inputs [192]. Since 3-D models are involved, this line of studies typically does not use GANs.

Some follow-up studies take transferring head motions into account and can model both expressions and different head poses at the same time [11], [87], [227]. For example, RecycleGAN [11] extends CycleGAN [254] to incorporate temporal constraints, so it can transform

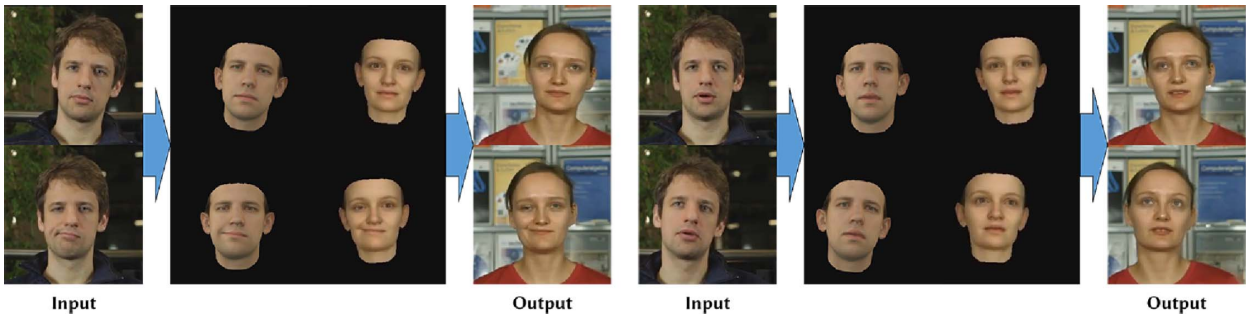



**Fig. 17.** *Face reenactment using 3-D face models [87]. These methods first construct a 3-D model for the person to be synthesized, so they can easily animate the model with new expressions. Images are from Kim et al. [87].*

videos of a particular person into another fixed person. On the other hand, ReenactGAN [227] can transfer movements and expressions from an arbitrary person to a fixed person. Still, the subject-dependent nature of these studies greatly limits their usability. One model can only work for one person, and generalizing to another person requires training a new model. Moreover, collecting training data for the target person may not be feasible at all times, which motivates the emergence of subject-agnostic models.

*2) Subject-Agnostic Model:* Several recent studies propose subject-agnostic frameworks, which focus on transferring the facial expressions without head movements [28], [29], [49], [53], [77], [144], [152], [159], [160], [190], [211], [251]. In particular, many studies only focus on the mouth region since it is the most expressive part of talking. For example, given an audio speech and one lip image of the target identity, Chen *et al.* [28] synthesized a video of the desired lip movements. Fried *et al.* [49] edited the lower face region of an existing video, so they can edit the video script and synthesize a new video corresponding to the change. While these studies have better generalization capability than the previous subject-specific methods, they usually cannot synthesize spontaneous head motions. The head movements cannot be transferred from the driving sequence to the target person.

Some studies can very recently handle both expressions and head movements using subject-agnostic frameworks [7], [62], [149], [184], [216], [219], [226], [239]. These frameworks only need a single 2-D image of the target person and can synthesize talking videos of this person given arbitrary motions. These motions are represented using either facial landmarks [7] or keypoints learned without supervision [184]. Since the input is only a 2-D image, many methods rely on warping the input or its extracted features and then fill in the unoccluded areas to refine the results. For example, Averbuch-Elor *et al.* [7] first warped the image and directly copied the teeth region from the driving image to fill in the holes in the case of an open mouth. Siarohin *et al.* [184] warped the extracted features from the input image, using motion fields estimated from sparse keypoints. On the other hand, Zakharov *et al.* [239] demonstrated that it is possible to achieve promising results using direct synthesis methods without any warping. To synthesize the target identity, they extract features from the source images and inject the information into the generator through the AdaIN [69] parameters. Similarly, the few-shot vid2vid [216] injects the information into their generator by dynamically determining the SPADE [156] parameters. Since these methods require only an image as input, they become particularly powerful and can be used in even more cases. For instance, several studies [7], [216], [239] demonstrate successes in animating paintings or graffiti instead of real humans, as shown in Fig. 18, which is not possible with the previous subject-dependent approaches. Recently, Wang *et al.* [219] demonstrated the use of a novel subject-agnostic face reenactment method for video conferencing, achieving an order of magnitude bandwidth saving over the H.264 standard. However, while these methods have achieved great results in synthesizing people talking under natural motions, they usually struggle to generate satisfying outputs under extreme poses or uncommon expressions, especially when the target pose is very different from the original one. Moreover, synthesizing complex regions, such as hair or background, is still hard. This is, indeed, a very challenging task that is still open to further research. A summary of different categories of face reenactment methods can be found in Table 2.

**Table 2** Categorization of Face Reenactment Methods. Subject-Specific Models Can Only Work on One Subject per Model, While Subject-Agnostic Models Can Work on General Targets. Among Each of Them, Some Frameworks Only Focus on the Inner Face Region, So They Can Only Transfer Expressions, While Others Can Also Transfer Head Movements. Studies With * Do Not Use GANs in Their Framework

| Target subject | Transferred region | Methods |
|---|---|---|
| Specific | Face only | [209]*, [198]*, [199]*, [192]*, [197]* |
| | Entire head | [87], [11], [227] |
| General | Face only | [152], [28], [53], [144], [159], [251], [29], [190], [77]*, [211], [49], [160] |
| | Entire head | [7]*, [226]*, [149], [239], [162]*, [216], [184]*, [65] |

## B. Pose Transfer

Pose transfer techniques aim at transferring the body pose of one person to another person. It can be seen as the whole body counterpart of face reenactment. In contrast to the talking head generation, which usually shares similar motions, body poses have more varieties and are, thus, much harder to synthesize. Early studies focus on simple pose transfers that generate low resolution and lower quality images. They work only on single images instead of videos. Recent studies have shown their capability to generate high-quality and HR videos for challenging poses but can only work on a particular person per model. Very recently, several studies attempt to perform subject-agnostic video synthesis. A summary of the categories is shown in Table 3. In the following, we introduce each category in more detail.

*1) Subject-Agnostic Image Generation:* Although we focus on video synthesis in this section, since most of the existing motion transfer approaches only focus on synthesizing images, we still briefly introduce them here (see [10], [44], [46], [61], [82], [108], [124], [129], [130], [146], [161], [164], [186], [189], [240], [248], and [257]). Ma *et al.* [129] adopted a two-stage coarse-to-fine approach using GANs to synthesize a person in a different pose, represented by a set of keypoints. In their follow-up work [130], the foreground, background, and poses in the image are further disentangled into different latent codes

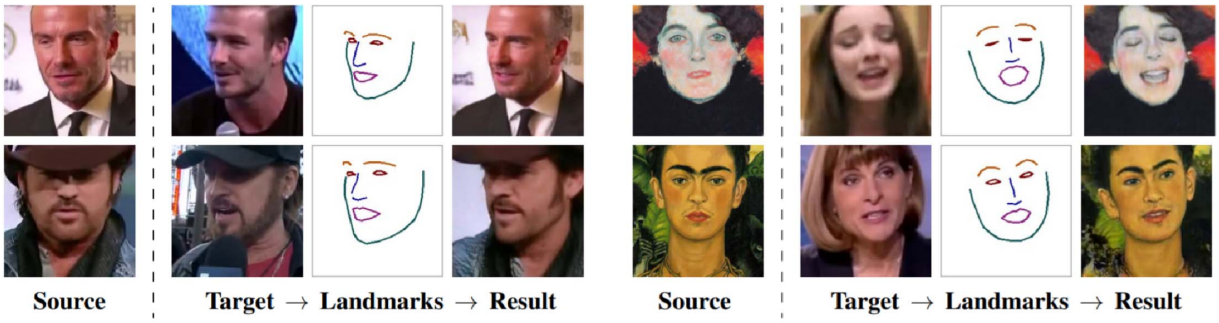



**Fig. 18.** *Few-shot face reenactment methods which require only a 2-D image as input [239]. The driving expressions are usually represented by facial landmarks or keypoints. Images are from Zakharov et al. [239].*

**Table 3** Categories of Pose Transfer Methods. Again, They Can Be Classified Depending on Whether One Model Can Work for Only One Person or Any Persons. Some of the Frameworks Only Focus on Generating Single Images, While Others Also Demonstrate Their Effectiveness on Videos. Studies With * Do Not Use GANs in Their Framework

| Target subject | Output type | Methods |
|---|---|---|
| Specific | Videos | [200]*, [217], [26], [3], [183]*, [252], [116], [115] |
| General | Images | [129], [130], [186], [46]*, [10], [161], [240]*, [82], [248], [146], [164], [44], [61], [108], [124], [189], [257] |
| | Videos | [233], [185], [224]*, [121], [184]*, [216], [166] |

to provide more flexibility and controllability. Later, Siarohin *et al.* [186] introduced deformable skip connections to move local features to the target pose position in a U-Net generator. Similarly, Balakrishnan *et al.* [10] decomposed different parts of the body into different layer masks and apply spatial transforms to each of them. The transformed segments are then fused together to form the final output.

The above methods work in a supervised setting where images of different poses of the same person are available during training. To work in the unsupervised setting, Pumarola *et al.* [161] rendered the synthesized image back to the original pose and applied cycle-consistency constraint on the back-rendered image. Lorenz *et al.* [124] decoupled the shape and appearance from images without supervision by adopting a two-stream auto-encoding architecture, so they can resynthesize images in a different shape with the same appearance.

Recently, instead of relying on 2-D keypoints solely, some frameworks choose to utilize 3-D or 2.5-D information. For example, Zanfir *et al.* [240] incorporated estimating 3-D parametric models into their framework to aid the synthesis process. Similarly, Li *et al.* [108] predicted 3-D dense flows to warp the source image by estimating 3-D models from the input images. Neverova *et al.* [146] adopted the DensePose [63] to help warp the input textures according to their UV-coordinates and inpaint the holes to generate the final result. Grigorev *et al.* [61] also mapped the input to texture space and inpainted the textures before warping them back to the target pose. Huang *et al.* [71] combined the SMPL models [123] with the implicit field estimation framework [172] to rig the reconstructed meshes with desired motions. While these methods work reasonably well in transferring poses, as shown in Fig. 19, directly applying them to videos will usually result in unsatisfactory artifacts, such as flickering or inconsistent results. In the following, we introduce methods specifically targeting video generation, which works on a one-person-per-model basis.

*2) Subject-Specific Video Generation:* For high-quality video synthesis, most methods employ a subject-specific model, which can only synthesize a particular person. These approaches start with collecting training data of the target person to be synthesized (e.g., a few minutes of a subject performing various motions) and then train a neural network or infer a 3-D model from it to synthesize the output. For example, Thies *et al.* [200] extended their previous face reenactment work [199] to include shoulders and part of the upper body to increase realism and fidelity. To extend to whole-body motion transfer, Wang *et al.* [217] extended their image synthesis framework [218] to videos

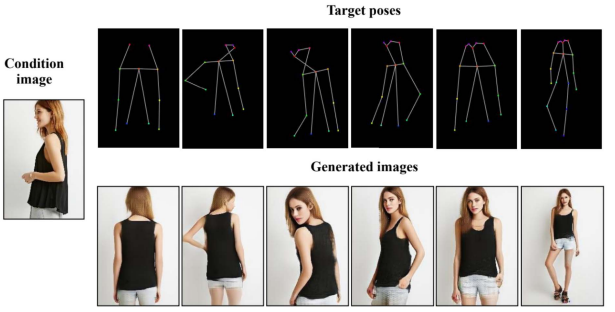



**Fig. 19.** *Subject-agnostic pose transfer examples [257]. Using only a 2-D image and the target pose to be synthesized, these methods can realistically generate the desired outputs. Images are from Zhu et al. [257].*

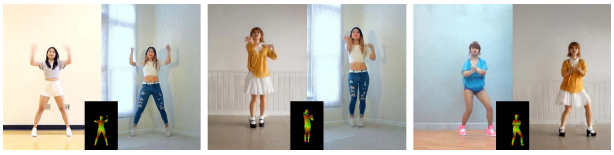


**Fig. 20.** *Subject-specific pose transfer examples for video generation [217]. For each image triplet, (left) driving sequence, (middle) intermediate pose representation, and (right) synthesized output. By using a model specifically trained on the target person, it can synthesize realistic output videos faithfully reflecting the driving motions. Images are from Wang et al. [217].*

and successfully demonstrated the transfer results on several dancing sequences, opening the era for a new application (see Fig. 20). Chan *et al.* [26] also adopted a similar approach to generate many dancing examples but using a simple temporal smoothing on the inputs instead of explicitly modeling temporal consistency by the network. Following these studies, many subsequent studies improve upon them [3], [115], [116], [183], [252], usually by combining the neural network with 3-D models or graphics engines. For example, instead of predicting RGB values directly, Shysheya *et al.* [183] predicted DensePose-like part maps and texture maps from input 3-D keypoints and adopted a neural renderer to render the outputs. Liu *et al.* [116] first constructed a 3-D character model of the target by capturing multiview static images and then trained a character-to-image translation network using a monocular video of the target. The authors later combined the constructed 3-D model with the monocular video to estimate dynamic textures, so they can use different texture maps when synthesizing different motions to increase the realism [115].

*3) Subject-Agnostic Video Generation:* Finally, the most general framework would be to have one model that can work universally regardless of the target identity. Early studies in this category synthesize videos unconditionally and do not have full control over the synthesized sequence (e.g., MoCoGAN [205]). Some other studies, such as [233], have control over the appearance and the starting pose of the person, but the motion generation is still unconditional. Due to these factors, the synthesized videos are usually shorter and of lower quality. Very recently, a few studies have shown the ability to render higher quality videos for pose transfer results [121], [166], [184], [185], [216], [224]. Weng *et al.* [224] reconstructed the SMPL model [123] from the input image and animated it with some simple motions, such as running. Liu *et al.* [121] proposed a unified framework for pose transfer, novel view synthesis, and appearance transfer all at once. Siarohin *et al.* [184], [185] estimated unsupervised keypoints from the input images and predicted a dense motion field to warp the source features to the target pose. Wang *et al.* [216] extended vid2vid [217] to the few-shot setting by predicting kernels in the SPADE [156] modules. Similarly, Ren *et al.* [166] also predicted kernels in their local attention modules using the input images to adaptively select features and warp them to the target pose. While these approaches have achieved better results than previous studies (see Fig. 21), their qualities are still not comparable to state-of-the-art subject-specific models. Moreover, most of them still synthesize lower resolution outputs (256 or 512). How to further increase the quality and resolution to the photorealistic level is still an open question.

## VII. NEURAL RENDERING

Neural rendering is a recent and upcoming topic in the area of neural networks, which combines classical rendering and generative models. Classical rendering can produce photorealistic images given the complete specification of the world. This includes all the objects in it, their geometry, material properties, the lighting, the cameras, and so on. Creating such a world from scratch is a laborious process that often requires expert manual input. Moreover, faithfully reproducing such data directly from images of the world can often be hard or impossible. On the other hand, as described in Sections IV–VI, GANs have had great success in producing photorealistic images given minimal semantic inputs. The ability to synthesize and learn material properties, textures, and other intangibles from training data can help overcome the drawbacks of classical rendering.

Neural rendering aims to combine the strengths of the two areas to create a more powerful and flexible framework. Neural networks can either be applied as a postprocessing step after classical rendering or as part of the rendering pipeline with the design of 3-D-aware and


**Fig. 21.** *Subject-agnostic pose transfer videos [216]. Given an example image and a driving pose sequence, the methods can output a sequence of the person performing the motions. Images are from Wang et al. [216].*

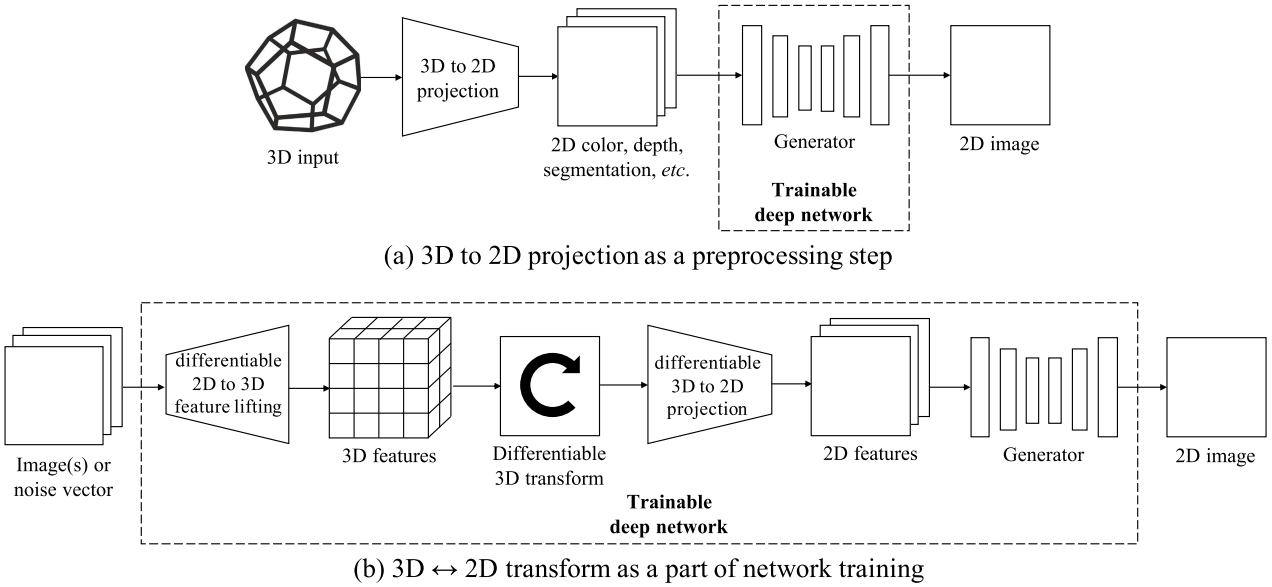



**Fig. 22.** ***Two common frameworks for neural rendering. (a) In the first set of studies [109], [132], [135], [139], [158], a neural network that purely operates in the 2-D domain is trained to enhance an input image, possibly supplemented with other information, such as depth or segmentation maps. (b) Second set of studies [147], [148], [178], [187], [225] introduces native 3-D operations that produce and transform 3-D features. This allows the network to reason in 3-D and produce view-consistent outputs. (a) 3-D to 2-D projection as a preprocessing step. (b) 3-D ↔ 2-D transform as a part of network training.***

differentiable layers. Sections VII-A and B discuss such approaches and how they use GAN losses to improve the quality of outputs. In this article, we focus on studies that use GANs to train neural networks and augment the classical rendering pipeline to generate images. For a general survey on the use of neural networks in rendering, refer the survey paper on neural rendering by Tewari *et al.* [196].

We divide the studies on GAN-based neural rendering into two parts: 1) studies that treat 3-D to 2-D projection as a preprocessing step and apply neural networks purely in the 2-D domain and 2) studies that incorporate layers that perform differentiable operations to transform features from 3-D to 2-D or vice versa (3-D ↔ 2-D) and learn some implicit form of geometry to provide 3-D understanding to the network.

## A. 3-D to 2-D Projection as a Preprocessing Step

A number of studies [109], [132], [135], [139], [158] improve upon traditional techniques by casting the task of rendering into the framework of image-to-image translation, possibly unimodal, multimodal, or conditional, depending on the exact use cases. Using given camera parameters, the source 3-D world is first projected to a 2-D feature map containing per-pixel information, such as color, depth, surface normals, and segmentation. This feature map is then fed as input to a generator, which tries to produce desired outputs, usually a realistic-looking RGB image. The deep neural network application happens in the 2-D space after the 3-D world is projected to the camera view, and no features or gradients are backpropagated to the 3-D source world or through the camera projection. A key advantage of this approach is that the traditional graphics rendering pipeline can be easily augmented to immediately take advantage of proven and mature techniques from 2-D image-to-image translation (as discussed in Section IV), without the need for designing and implementing differentiable projection layers or transformations that are part of the deep network during training. This type of framework is illustrated in Fig. 22(a).

Martin-Brualla *et al.* [135] introduced the notion of rerendering, where a deep neural network takes as input a rendered 2-D image and enhances it (improving colors, boundaries, resolution, and so on) to produce a rerendered image. The full pipeline consists of two steps—a traditional 3-D to 2-D rendering step and a trainable deep network that enhances the rendered 2-D image. The 3-D to 2-D rendering technique can be differentiable or nondifferentiable, but no gradients are backpropagated through this step. This allows one to use more complex rendering techniques. By using this two-step process, the output of a performance capture system, which might suffer from noise, poor color reproduction, and other issues, can be improved. In this particular work, they did not see an improvement from using a GAN loss, perhaps because they trained their system on the limited domain of people and faces, using carefully captured footage.

Meshry *et al.* [139] and Li *et al.* [109] extended this approach to the more challenging domain of unstructured photo collections. They produce multiple plausible views

of famous landmarks from noisy point clouds generated from Internet photo collections by utilizing Structure from Motion (SfM). Meshry *et al.* [139] generated a 2-D feature map containing per-pixel albedo and depth by splatting points of the 3-D point cloud onto a given viewpoint. The segmentation map of the expected output image is also concatenated to this feature representation. The problem is then framed as a multimodal image translation problem. A noisy and incomplete input has to be translated to a realistic image conditioned on a style code to produce desired environmental effects, such as lighting. Li *et al.* [109] used a similar approach but with multiplane images and achieve better photorealism. Pittaluga *et al.* [158] tackled the task of producing 2-D color images of the underlying scene given as input a sparse SfM point cloud with associated point attributes, such as color, depth, and SIFT descriptors. The input to their network is a 2-D feature map obtained by projecting the 3-D points to the image plane given the camera parameters. The attributes of the 3-D point are copied to the 2-D pixel location to which it is projected. Mallya *et al.* [132] precomputed the mapping of the 3-D world point cloud to the pixel locations in the images produced by cameras with known parameters and use this to obtain an estimate of the next frame, referred to as a "guidance image." They learn to output video frames consistent over time and viewpoints by conditioning the generator on these noisy estimates.

In these studies, the use of a generator coupled with an adversarial loss helps produce better-looking outputs conditioned on the input feature maps. Similar to applications of pix2pixHD [218], such as manipulating output images by editing input segmentation maps, Meshry *et al.* [139] are able to remove people and transient objects from images of landmarks and generate plausible inpainting. A key motivation of the work of Pittaluga *et al.* [158] was to explore if a user's privacy can be protected by techniques, such as discarding the color of the 3-D points. A very interesting observation was that discarding color information helps prevent accurate reproduction. However, the use of a GAN loss recovers plausible colors and greatly improves the output results, as shown in Fig. 23. GAN losses might also be helpful in cases where it is hard to manually define a good loss function, either due to the inherent ambiguity in determining the desired behavior or the difficulty in fully labeling the data.

**Fig. 23.** *Inverting images from 3-D point clouds and their associated depth and SIFT attributes [158]. The top row of images is produced by a generator trained without an adversarial loss, whereas the bottom row uses adversarial loss. Using an adversarial loss helps generates better details and more plausible colors. Images are from Pittaluga et al. [158].*

### B. 3-D ↔ 2-D Transform as a Part of Network Training

In the previous set of studies, the geometry of the world or object is explicitly provided, and neural rendering is purely used to enhance the appearance or add details to the traditionally rendered image or feature maps. The studies in this section [147], [148], [178], [187], [225] introduce native 3-D operations in the neural network used to learn from and produce images. These operations enable them to model the geometry and appearance of the scene in the feature space. The general pipeline of this line of studies is illustrated in Fig. 22(b). Learning a 3-D representation and modeling the process of image projection and formation into the network have several advantages: the ability to reason in 3-D, control the pose, and produce a series of consistent views of a scene. Contrast to this is the neural network shown in Fig. 22(a), which purely operates in the 2-D domain.

DeepVoxels [187] learns a persistent 3-D voxel feature representation of a scene given a set of multiview images and their associated camera intrinsic and extrinsic parameters. Features are first extracted from the 2-D views and then lifted to a 3-D volume. This 3-D volume is then integrated into the persistent DeepVoxels representation. These 3-D features are then projected to 2-D using a projection layer, and a new view of the object is synthesized using a U-Net generator. This generator network is trained with an $\ell_1$ loss and a GAN loss. The authors found that using a GAN loss accelerates the generation of high-frequency details, especially at earlier stages of training. Similar to DeepVoxels [187], visual object networks (VONs) [256] generate a voxel grid from a sample noise vector and use a differentiable projection layer to map the voxel grid to a 2.5-D sketch. Inspired by classical graphics rendering pipelines, this work decomposes image formation into three conditionally independent factors of shape, viewpoint, and texture. Trained with a GAN loss, their model synthesizes more photorealistic images, and the use of the disentangled representation allows for 3-D manipulations, which are not feasible with purely 2-D methods.

HoloGAN [147] proposes a system to learn 3-D voxel feature representations of the world and to render them

to realistic-looking images. Unlike VONs [256], HoloGAN does not require explicit 3-D data or supervision and can do so using unlabeled images (no pose, explicit 3-D shape, or multiple views). By incorporating a 3-D rigid-body transformation module and a 3-D-to-2-D projection module in the network, HoloGAN provides the ability to control the pose of the generated objects. HoloGAN employs a multiscale feature GAN discriminator, and the authors empirically observed that this helps prevent mode collapse. BlockGAN [148] extends the unsupervised approach of the HoloGAN [147] to also consider object disentanglement. BlockGAN learns 3-D features per object and the background. These are combined into 3-D scene features after applying appropriate transformations before projecting them into the 2-D space. One issue with learning scene compositionality without explicit supervision is the conflation of features of the foreground object and the background, which results in visual artifacts when objects or the camera moves. By adding more powerful "style" discriminators (feature discriminators introduced in [147]) to their training scheme, the authors observed that the disentangling of features improved, resulting in cleaner outputs.

SynSin [225] learns an end-to-end model for view synthesis from a single image, without any ground-truth 3-D supervision. Unlike the above studies that internally use a feature voxel representation, SynSin predicts a point cloud of features from the input image and then projects it to new views using a differentiable point cloud renderer. Two-dimensional image features and a depth map are first predicted from the input image. Based on the depth map, the 2-D features are projected to 3-D to obtain the 3-D feature point cloud. The network is trained adversarially with a discriminator based on the one proposed by Wang *et al.* [218].

One of the drawbacks of voxel-based feature representations is the cubic growth in the memory required to store them. To keep requirements manageable, voxel-based approaches are typically restricted to low resolutions. GRAF [178] proposes to use conditional radiance fields, which are a continuous mapping from a 3-D location and a 2-D viewing direction to an RGB color value, as the intermediate feature representation. They also use a single discriminator similar to PatchGAN [75], with weights that are shared across patches with different receptive fields. This allows them to capture the global context and refine local details.

As summarized in Table 4, the studies discussed in this section use a variety of 3-D feature representations and train their networks using paired input–output with known transformations or unlabeled and unpaired data. The use of a GAN loss is common to all these approaches. This is perhaps because traditional hand-designed losses, such as the $\ell_1$ loss or even perceptual loss, are unable fully to capture what makes a synthesized image look unrealistic. Furthermore, in the case where explicit task supervision is unavailable, BlockGAN [148] shows that a GAN loss can help in learning disentangled features by ensuring that the outputs after projection and rendering look realistic. The learnability and flexibility of the GAN loss to the task at hand help provide feedback, guiding how to change the generated image, and, thus, the upstream features so that it looks as if it were sampled from the distribution of real images. This makes the GAN framework a powerful asset in the toolbox of any neural rendering practitioner.

**Table 4** Key Differences Among 3-D-Aware Methods. Adversarial Losses Are Used by a Range of Methods That Differ in the Type of 3-D Feature Representation and Training Supervision

| 3D feature representation | Supervision | Methods |
|---|---|---|
| Radiance field | None | GRAF [178] |
| Voxel | None | HoloGAN [147] |
| | | BlockGAN [148] |
| | 3D supervision | VONs [256] |
| | Input-Output | DeepVoxels [187] |
| Point cloud | pose transformation | SynSin [225] |

## VIII. LIMITATIONS AND OPEN PROBLEMS

Despite the successful applications introduced above, there are still limitations of GANs needed to be addressed by future work.

### A. Evaluation Metrics

Evaluating and comparing different GAN models are difficult. The most popular evaluation metrics are perhaps inception score (IS) [176] and Fréchet inception distance (FID) [67], which both have many shortcomings. The IS, for example, is not able to detect intraclass mode collapse [23]. In other words, a model that generates only a single image per class can obtain a high IS. FID can better measure such diversity, but it does not have an unbiased estimator [19]. Kernel inception distance (KID) [19] can capture higher order statistics and has an unbiased estimator but has been empirically found to suffer from high variance [165]. In addition to the above measures that summarize the performance with a single number, there are metrics that separately evaluate fidelity and diversity of the generator distribution [97], [143], [173].

### B. Instability

Although the regularization techniques introduced in Section III-C have greatly improved the stability of GAN training, GANs are still much more unstable to train than supervised discriminative models or likelihood-based generative models. For example, even the state-of-the-art BigGAN model would eventually collapse in the late stage of training on ImageNet [24]. Also, the final performance is generally very sensitive to hyperparameters [96], [126].

### C. Interpretability

Despite the impressive quality of the generate images, there has been a lack of understanding of how GANs represent the image structure internally in the generator. Bau *et al.* [13] visualized the causal effect of different neurons on the output image. After finding the semantic meaning of individual neurons or directions in the latent space [55], [76], [180], one can edit a real image by inverting it to the latent space, edit the latent code according to the desired semantic change, and regenerate it with the generator. Finding the best way to encode an image to the latent space is, therefore, another interesting research direction [1], [2], [14], [72], [86], [253].

### D. Forensics

The success of GANs has enabled many new applications but also raised ethical and social concerns, such as fraud and fake news. The ability to detect GAN-generated images is essential to prevent malicious usage of GANs. Recent studies have found it possible to train a classifier to detect generated images and generalize to unseen generator architectures [25], [215], [245]. This cat-and-mouse game may continue, as generated images may become increasingly harder to detect in the future.

## IX. CONCLUSION

In this article, we present a comprehensive overview of GANs with an emphasis on algorithms and applications to visual synthesis. We summarize the evolution of the network architectures in GANs and the strategies to stabilize GAN training. We then introduce several fascinating applications of GANs, including image translation, image processing, video synthesis, and neural rendering. In the end, we point out some open problems for GANs, and we hope that this article would inspire future research to solve them. ■

## REFERENCES


[1] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to embed images into the StyleGAN latent space?" *ICCV*, 2019, pp. 4432–4441.

[2] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to edit the embedded images?" *CVPR*, 2020, pp. 8296–8305.

[3] K. Aberman, M. Shi, J. Liao, D. Liscbinski, B. Chen, and D. Cohen-Or, "Deep video-based performance cloning," *Comput. Graph. Forum*, vol. 38, no. 2, pp. 219–233, May 2019.

[4] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. ICCV*, 2019, pp. 221–231.

[5] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, "Augmented CycleGAN: Learning many-to-many mappings from unpaired data," in *Proc. ICML*, 2018, pp. 195–204.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proc. ICML*, 2017, pp. 214–223.

[7] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–13, Nov. 2017.

[8] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[9] K. Baek, Y. Choi, Y. Uh, J. Yoo, and H. Shim, "Rethinking the truly unsupervised image-to-image translation," 2020, *arXiv:2006.06500*. [Online]. Available: http://arxiv.org/abs/2006.06500

[10] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *Proc. CVPR*, 2018, pp. 8340–8348.

[11] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *Proc. ECCV*, 2018, pp. 119–135.

[12] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[13] D. Bau *et al.*, "GAN dissection: Visualizing and understanding generative adversarial networks," in *Proc. ICLR*, 2019.

[14] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba, "Seeing what a GAN cannot generate," in *Proc. ICCV*, 2019.

[15] S. Benaim, R. Mokady, A. Bermano, D. Cohen-Or, and L. Wolf, "Structural-analogy from a single image pair," 2020, *arXiv:2004.02222*. [Online]. Available: http://arxiv.org/abs/2004.02222

[16] S. Benaim and L. Wolf, "One-sided unsupervised domain mapping," in *Proc. NeurIPS*, 2017, pp. 752–762.

[17] S. Benaim and L. Wolf, "One-shot unsupervised cross domain translation," in *Proc. NeurIPS*, 2018, pp. 2104–2114.

[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[19] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *Proc. ICLR*, 2018.

[20] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn.*, 1999, pp. 187–194.

[21] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," in *Proc. ECCV Workshop*, 2018, pp. 334–355.

[22] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. CVPR*, 2018, pp. 6228–6237.

[23] A. Borji, "Pros and cons of GAN evaluation measures," *Comput. Vis. Image Understand.*, vol. 179, pp. 41–65, Feb. 2019.

[24] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proc. ICLR*, 2019.

[25] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *Proc. ECCV*, 2020, pp. 103–120.

[26] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proc. ICCV*, 2019, pp. 5933–5942.

[27] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Proc. CVPR*, 2018, pp. 3155–3164.

[28] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu, "Lip movements generation at a glance," in *Proc. ECCV*, 2018, pp. 520–535.

[29] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *Proc. CVPR*, 2019, pp. 7832–7841.

[30] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "PixelSNAIL: An improved autoregressive generative model," in *Proc. ICML*, 2018, pp. 864–872.

[31] X. Chen, C. Xu, X. Yang, and D. Tao, "Attention-GAN for object transfiguration in wild images," in *Proc. ECCV*, 2018, pp. 164–180.

[32] 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. CVPR*, 2018, pp. 8789–8797.

[33] A. Clark, J. Donahue, and K. Simonyan, "Efficient video generation on complex datasets," 2019, *arXiv:1907.06571*. [Online]. Available: https://arxiv.org/abs/1907.06571

[34] T. Cohen and L. Wolf, "Bidirectional one-shot unsupervised domain mapping," in *Proc. ICCV*, 2019, pp. 1784–1792.

[35] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886–893.

[37] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural Comput.*, vol. 7, no. 5, pp. 889–904, Sep. 1995.

[38] E. de Bézenac, I. Ayed, and P. Gallinari, "Optimal unsupervised domain translation," 2019, *arXiv:1906.01292*. [Online]. Available: http://arxiv.org/abs/1906.01292

[39] E. L. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. NeurIPS*, 2017, pp. 4414–4423.

[40] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *Proc. ICLR*, 2015.

[41] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP" in *Proc. ICLR*, 2017.

[42] 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.

[43] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. ECCV*, 2016, pp. 391–407.

[44] H. Dong, X. Liang, K. Gong, H. Lai, J. Zhu, and J. Yin, "Soft-gated warping-GAN for pose-guided person image synthesis," in *Proc. NeurIPS*, 2018, pp. 474–484.

[45] Y. Du and I. Mordatch, "Implicit generation and generalization in energy-based models," in *Proc. NeurIPS*, 2019, pp. 3608–3618.

[46] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Proc. CVPR*, 2018, pp. 8857–8866.

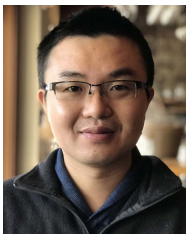
[47] C. Finn, I. Goodfellow, and S. Levine,

"Unsupervised learning for physical interaction through video prediction," in *Proc. NeurIPS*, 2016, pp. 64–72.
[48] A. Fischer and C. Igel, "An introduction to restricted Boltzmann machines," in *Proc. Iberoamerican Congr. Pattern Recognit.*, 2012, pp. 14–36.
[49] O. Fried *et al.*, "Text-based editing of talking-head video," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, Jul. 2019.
[50] O. Gafni, L. Wolf, and Y. Taigman, "Vid2Game: Controllable characters extracted from real-world videos," in *Proc. ICLR*, 2020.
[51] T. Galanti, L. Wolf, and S. Benaim, "The role of minimal complexity functions in unsupervised learning of semantic mappings," in *Proc. ICLR*, 2018.
[52] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep generative adversarial compression artifact removal," in *Proc. ICCV*, 2017, pp. 4826–4835.
[53] J. Geng, T. Shao, Y. Zheng, Y. Weng, and K. Zhou, "Warp-guided GANs for single-photo facial animation," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, Jan. 2019.
[54] A. Ghosh, V. Kulharia, V. P. Namboodiri, P. H. Torr, and P. K. Dokania, "Multi-agent diverse generative adversarial networks," in *Proc. CVPR*, 2018, pp. 8513–8521.
[55] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "GANalyze: Toward visual definitions of cognitive image properties," in *Proc. ICCV*, 2019, pp. 5744–5753.
[56] X. Gong, S. Chang, Y. Jiang, and Z. Wang, "AutoGAN: Neural architecture search for generative adversarial networks," in *Proc. ICCV*, 2019, pp. 3224–3234.
[57] A. Gonzalez-Garcia, J. Van De Weijer, and Y. Bengio, "Image-to-image translation for cross-domain disentanglement," in *Proc. NeurIPS*, 2018, pp. 1287–1298.
[58] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: http://arxiv.org/abs/1701.00160
[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[60] I. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. NeurIPS*, 2014.
[61] A. Grigorev, A. Sevastopolsky, A. Vakhitov, and V. Lempitsky, "Coordinate-based texture inpainting for pose-guided human image generation," in *Proc. CVPR*, 2019, pp. 12135–12144.
[62] K. Gu, Y. Zhou, and T. S. Huang, "FLNet: Landmark driven fetching and learning network for faithful talking facial animation synthesis," in *Proc. AAAI*, 2020, pp. 10861–10868.
[63] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. CVPR*, 2018, pp. 7297–7306.
[64] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. NeurIPS*, 2017, pp. 5767–5777.
[65] S. Ha, M. Kersner, B. Kim, S. Seo, and D. Kim, "MarioNETte: Few-shot face reenactment preserving identity of unseen targets," in *Proc. AAAI*, 2020, pp. 10893–10900.
[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 10893–10900.
[67] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NeurIPS*, 2017, pp. 6626–6637.
[68] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[69] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. ICCV*, 2017, pp. 1501–1510.
[70] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. ECCV*, 2018, pp. 172–189.
[71] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung, "ARCH: Animatable reconstruction of clothed humans," in *Proc. CVPR*, 2020, pp. 3093–3102.
[72] M. Huh, R. Zhang, J.-Y. Zhu, S. Paris, and A. Hertzmann, "Transforming and projecting images into class-conditional generative networks," in *Proc. ECCV*, 2020, pp. 17–34.
[73] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, Jul. 2017.
[74] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.
[75] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017, pp. 1125–1134.
[76] A. Jahanian, L. Chai, and P. Isola, "On the 'steerability' of generative adversarial networks," in *Proc. ICLR*, 2020.
[77] A. Jamaludin, J. S. Chung, and A. Zisserman, "You said that: Synthesising talking faces from audio," in *Proc. IJCV*, 2019, pp. 1767–1779.
[78] Y. Jo and J. Park, "SC-FEGAN: Face editing generative adversarial network with user's sketch and color," in *Proc. ICCV*, 2019, pp. 1745–1753.
[79] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016, pp. 694–711.
[80] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," in *Proc. ICLR*, 2019.
[81] A. Jolicoeur-Martineau, "On relativistic f-divergences," in *Proc. ICML*, 2020, pp. 4931–4939.
[82] D. Joo, D. Kim, and J. Kim, "Generating a fusion image: One's identity and another's shape," in *Proc. CVPR*, 2018, pp. 1635–1643.
[83] N. Kalchbrenner *et al.*, "Video pixel networks," in *Proc. ICML*, 2017, pp. 1771–1779.
[84] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. ICLR*, 2018.
[85] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. CVPR*, 2019, pp. 4401–4410.
[86] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. CVPR*, 2020, pp. 8110–8119.
[87] H. Kim *et al.*, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Aug. 2018.
[88] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. CVPR*, 2016, pp. 1646–1654.
[89] K. Kim, Y. Yun, K.-W. Kang, K. Kong, S. Lee, and S.-J. Kang, "Painting outside as inside: Edge guided image outpainting via bidirectional rearrangement with progressive step learning," 2020, *arXiv:2010.01810*. [Online]. Available: http://arxiv.org/abs/2010.01810
[90] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. ICML*, 2017, pp. 1857–1865.
[91] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.
[92] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible $1\times1$ convolutions," in *Proc. NeurIPS*, 2018, pp. 10215–10224.
[93] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2013.
[94] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," 2019, *arXiv:1906.02691*. [Online]. Available: http://arxiv.org/abs/1906.02691
[95] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. CVPR*, 2018, pp. 8183–8192.
[96] K. Kurach, M. Lučić, X. Zhai, M. Michalski, and S. Gelly, "A large-scale study on regularization and normalization in GANs," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3581–3590.
[97] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila, "Improved precision and recall metric for assessing generative models," in *Proc. NeurIPS*, 2019, pp. 3927–3936.
[98] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Proc. ECCV*, 2018, pp. 170–185.
[99] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. ICML*, 2016, pp. 1558–1566.
[100] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[101] Y. LeCun *et al.*, "A tutorial on energy-based learning," in *Predicting Structured Data*. Cambridge, MA, USA: MIT Press 2006.
[102] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. CVPR*, 2017.
[103] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," 2018, *arXiv:1804.01523*. [Online]. Available: http://arxiv.org/abs/1804.01523
[104] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. CVPR*, 2020.
[105] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. ECCV*, 2018, pp. 35–51.
[106] S. Lee, J. Ha, and G. Kim, "Harmonizing maximum likelihood with GANs for multimodal conditional generation," in *Proc. ICLR*, 2019.
[107] K. Li, T. Zhang, and J. Malik, "Diverse image synthesis from semantic layouts via conditional IMLE," in *Proc. ICCV*, 2019, pp. 4220–4229.
[108] Y. Li, C. Huang, and C. C. Loy, "Dense intrinsic appearance flow for human pose transfer," in *Proc. CVPR*, 2019, pp. 3693–3702.
[109] Z. Li, W. Xian, A. Davis, and N. Snavely, "Crowdsampling the plenoptic function," in *Proc. ECCV*, 2020, pp. 178–196.
[110] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. NeurIPS*, 2017, pp. 1744–1752.
[111] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. CVPR Workshop*, 2017, pp. 136–144.
[112] J. Hyun Lim and J. Chul Ye, "Geometric GAN," 2017, *arXiv:1705.02894*. [Online]. Available: http://arxiv.org/abs/1705.02894
[113] J. Lin, Y. Pang, Y. Xia, Z. Chen, and J. Luo, "TuiGAN: Learning versatile image-to-image translation with two unpaired images," in *Proc. ECCV*, 2020, pp. 18–35.
[114] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. ECCV*, 2018, pp. 85–100.
[115] L. Liu *et al.*, "Neural human video rendering by learning dynamic textures and rendering-to-video translation," *IEEE Trans. Vis. Comput. Graphics*, early access, May 2020, doi: 10.1109/TVCG.2020.2996594.
[116] L. Liu *et al.*, "Neural rendering and reenactment of human actor videos," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–14, Nov. 2019.
[117] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. NeurIPS*, 2017.
[118] M.-Y. Liu *et al.*, "Few-shot unsupervised image-to-image translation," in *Proc. ICCV*, 2019, pp. 10551–10560.
[119] M.-Y. Liu and O. Tuzel, "Coupled generative

adversarial networks," in *Proc. NeurIPS*, 2016, pp. 469–477.
[120] S. Liu, X. Zhang, J. Wangni, and J. Shi, "Normalized diversification," in *Proc. CVPR*, 2019, pp. 10306–10315.
[121] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, "Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis," in *Proc. ICCV*, 2019, pp. 5904–5913.
[122] X. Liu, G. Yin, J. Shao, and X. Wang, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *Proc. NeurIPS*, 2019, pp. 570–580.
[123] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, Nov. 2015.
[124] D. Lorenz, L. Bereska, T. Milbich, and B. Ommer, "Unsupervised part-based disentangling of object shape and appearance," in *Proc. CVPR*, 2019, pp. 10955–10964.
[125] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *Proc. ICLR*, 2017.
[126] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? A large-scale study," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 700–709.
[127] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Comput. Vis. Image Understand.*, vol. 158, pp. 1–16, May 2017.
[128] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool, "Exemplar guided unsupervised image-to-image translation with semantic consistency," in *Proc. ICLR*, 2019.
[129] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Proc. NeurIPS*, 2017, pp. 406–416.
[130] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Proc. CVPR*, 2018, pp. 99–108.
[131] S. Maeda, "Unpaired image super-resolution using pseudo-supervision," in *Proc. CVPR*, 2020, pp. 291–300.
[132] A. Mallya, T.-C. Wang, K. Sapra, and M.-Y. Liu, "World-consistent video-to-video synthesis," in *Proc. ECCV*, 2020, pp. 359–378.
[133] 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. CVPR*, 2019, pp. 1429–1437.
[134] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. ICCV*, 2017, pp. 2794–2802.
[135] R. Martin-Brualla *et al.*, "LookinGood: Enhancing performance capture with real-time neural re-rendering," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–14, Jan. 2019.
[136] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. ICLR*, 2016.
[137] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim, "Unsupervised attention-guided image-to-image translation," in *Proc. NeurIPS*, 2018, pp. 3693–3703.
[138] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for GANs do actually converge?" *ICML*, 2018, pp. 3693–3703.
[139] M. Meshry *et al.*, "Neural rerendering in the wild," in *Proc. CVPR*, 2019, pp. 6878–6887.
[140] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, 2018.
[141] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *Proc. ICLR*, 2018.
[142] S. Mo, M. Cho, and J. Shin, "InstaGAN: Instance-aware image-to-image translation," in *Proc. ICLR*, 2019.
[143] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *Proc. ICML*, 2020, pp. 7176–7185.
[144] K. Nagano *et al.*, "PaGAN: Real-time avatars using dynamic textures," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, Jan. 2019.
[145] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "EdgeConnect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*. [Online]. Available: http://arxiv.org/abs/1901.00212
[146] N. Neverova, R. Alp Guler, and I. Kokkinos, "Dense pose transfer," in *Proc. ECCV*, 2018, pp. 123–138.
[147] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "HoloGAN: Unsupervised learning of 3d representations from natural images," in *Proc. CVPR*, 2019, pp. 7588–7597.
[148] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra, "BlockGAN: Learning 3D object-aware scene representations from unlabelled images," 2020, *arXiv:2002.08988*. [Online]. Available: http://arxiv.org/abs/2002.08988
[149] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proc. ICCV*, 2019, pp. 7184–7193.
[150] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. NeurIPS*, 2016, pp. 271–279.
[151] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. ICML*, 2017, pp. 2642–2651.
[152] K. Olszewski *et al.*, "Realistic dynamic facial textures from a single image using GANs," in *Proc. ICCV*, 2017, pp. 5429–5438.
[153] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: http://arxiv.org/abs/1609.03499
[154] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.
[155] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. ECCV*, 2020, pp. 319–345.
[156] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. CVPR*, 2019, pp. 2337–2346.
[157] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. CVPR*, 2016, pp. 2536–2544.
[158] F. Pittaluga, S. J. Koppal, S. B. Kang, and S. N. Sinha, "Revealing scenes by inverting structure from motion reconstructions," in *Proc. CVPR*, 2019, pp. 145–154.
[159] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," in *Proc. ECCV*, 2018, pp. 818–833.
[160] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: One-shot anatomically consistent facial animation," *Int. J. Comput. Vis.*, vol. 128, pp. 698–713, Aug. 2020.
[161] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised person image synthesis in arbitrary poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8620–8628.
[162] S. Qian *et al.*, "Make a face: Towards arbitrary high fidelity face manipulation," in *Proc. ICCV*, 2019, pp. 10033–10042.
[163] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. ICLR*, 2015.
[164] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu, "SwapNet: Image based garment transfer," in *Proc. ECCV*, 2018, pp. 679–695.
[165] S. Ravuri and O. Vinyals, "Seeing is not necessarily believing: Limitations of BigGANs for data augmentation," in *Proc. ICLR Workshop*, 2019.
[166] Y. Ren, X. Yu, J. Chen, T. H. Li, and G. Li, "Deep image spatial transformation for person image generation," in *Proc. CVPR*, 2020, pp. 7690–7699.
[167] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, 2015, pp. 1530–1538.
[168] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. ICML*, 2014, pp. 1278–1286.
[169] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Proc. NeurIPS*, 2017, pp. 2018–2028.
[170] K. Saito, K. Saenko, and M.-Y. Liu, "COCO-FUNIT: Few-shot unsupervised image translation with a content conditioned style encoder," in *Proc. ECCV*, 2020, pp. 382–398.
[171] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proc. ICCV*, 2017, pp. 2830–2839.
[172] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. ICCV*, 2019, pp. 2304–2314.
[173] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," in *Proc. NeurIPS*, 2018, pp. 5228–5237.
[174] M. S. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *Proc. ICCV*, 2017, pp. 4491–4500.
[175] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Artificial Intelligence and Statistics*. 2009, pp. 448–455.
[176] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. NeurIPS*, 2016, pp. 2234–2242.
[177] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," in *Proc. ICLR*, 2017.
[178] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "GRAF: Generative radiance fields for 3D-aware image synthesis," 2020, *arXiv:2007.02442*. [Online]. Available: http://arxiv.org/abs/2007.02442
[179] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. ICCV*, 2019, pp. 4570–4580.
[180] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. CVPR*, 2020, pp. 9243–9252.
[181] Z. Shen, W.-S. Lai, T. Xu, J. Kautz, and M.-H. Yang, "Deep semantic face deblurring," in *Proc. CVPR*, 2018, pp. 8260–8269.
[182] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. CVPR*, 2016, pp. 1874–1883.
[183] A. Shysheya *et al.*, "Textured neural avatars," in *Proc. CVPR*, 2019, pp. 1874–1883.
[184] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Proc. NeurIPS*, 2019, pp. 7137–7147.
[185] A. Siarohin, S. Lathuiliére, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proc. CVPR*, 2019, pp. 2377–2386.
[186] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. CVPR*, 2018, pp. 3408–3416.
[187] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "DeepVoxels: Learning persistent 3D feature embeddings," in *Proc. CVPR*, 2019, pp. 2437–2446.
[188] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder

variational autoencoders," in *Proc. NeurIPS*, 2016, pp. 3738–3746.
[189] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised person image generation with semantic parsing transformation," in *Proc. CVPR*, 2019, pp. 2357–2366.
[190] Y. Song, J. Zhu, D. Li, A. Wang, and H. Qi, "Talking face generation by conditional recurrent adversarial network," in *Proc. IJCAI*, 2019, pp. 919–925.
[191] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proc. ICML*, 2015, pp. 843–852.
[192] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Jul. 2017.
[193] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. ICCV*, 2017, pp. 4539–4547.
[194] H. Tang, X. Qi, D. Xu, P. H. S. Torr, and N. Sebe, "Edge guided GANs with semantic preserving for semantic image synthesis," 2020, *arXiv:2003.13898*. [Online]. Available: http://arxiv.org/abs/2003.13898
[195] P. Teterwak *et al.*, "Boundless: Generative adversarial networks for image extension," in *Proc. ICCV*, 2019, pp. 10521–10530.
[196] A. Tewari *et al.*, "State of the art on neural rendering," *Comput. Graph. Forum (EG STAR)*, vol. 39, no. 2, pp. 701–727, 2020.
[197] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.
[198] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt, "Real-time expression transfer for facial reenactment," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–14, Nov. 2015.
[199] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. CVPR*, 2016, pp. 2387–2395.
[200] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Niessner, "Headon: Real-time reenactment of human portrait videos," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, Aug. 2018.
[201] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," in *Proc. Neural Netw. Mach. Learn. (COURSERA)*, 2012.
[202] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," in *Proc. ICLR*, 2018.
[203] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. ICCV*, 2017, pp. 4799–4807.
[204] M. Tschannen, E. Agustsson, and M. Lucic, "Deep generative models for distribution-preserving lossy compression," in *Proc. NeurIPS*, 2018, pp. 5929–5940.
[205] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proc. CVPR*, 2018, pp. 1526–1535.
[206] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: http://arxiv.org/abs/1607.08022
[207] A. Van den Oord *et al.*, "Conditional image generation with PixelCNN decoders," in *Proc. NeurIPS*, 2016, pp. 4790–4798.
[208] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proc. ICLR*, 2017.
[209] D. Vlasic, M. Brand, H. Pfister, and J. Popovic, "Face transfer with multilinear models," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 426–433, Jul. 2005.
[210] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. NeurIPS*, 2016, pp. 613–621.
[211] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic speech-driven facial animation with GANs," in *Proc. IJCV*, 2019, pp. 1–16.
[212] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Proc. ECCV*, 2016, pp. 835–851.
[213] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *Proc. ICCV*, 2017, pp. 3332–3341.
[214] M. Wang *et al.*, "Example-guided style-consistent image synthesis from semantic labeling," in *Proc. CVPR*, 2019, pp. 1495–1504.
[215] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot.. for now," in *Proc. CVPR*, 2020, pp. 8692–8701.
[216] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," in *Proc. NeurIPS*, 2019, pp. 8695–8704.
[217] T.-C. Wang *et al.*, "Video-to-video synthesis," in *Proc. NeurIPS*, 2018, pp. 1152–1164.
[218] 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. CVPR*, 2018, pp. 8798–8807.
[219] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," 2020, *arXiv:2011.15126*. [Online]. Available: http://arxiv.org/abs/2011.15126
[220] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. ECCV*, 2018, pp. 63–79.
[221] Y. Wang, S. Khan, A. Gonzalez-Garcia, J. van de Weijer, and F. S. Khan, "Semi-supervised learning for few-shot image-to-image translation," in *Proc. CVPR*, 2020, pp. 4453–4462.
[222] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," in *Proc. NeurIPS*, 2018, pp. 331–340.
[223] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," 2019, *arXiv:1906.01529*. [Online]. Available: http://arxiv.org/abs/1906.01529
[224] C.-Y. Weng, B. Curless, and I. Kemelmacher-Shlizerman, "Photo wake-up: 3D character animation from a single photo," in *Proc. CVPR*, 2019, pp. 5908–5917.
[225] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "SynSin: End-to-end view synthesis from a single image," in *Proc. CVPR*, 2020, pp. 7467–7477.
[226] O. Wiles, A. S. Koepke, and A. Zisserman, "X2Face: A network for controlling face generation using images, audio, and pose codes," in *Proc. ECCV*, 2018, pp. 670–686.
[227] W. Wu, Y. Zhang, C. Li, C. Qian, and C. Change Loy, "ReenactGAN: Learning to reenact faces via boundary transfer," in *Proc. ECCV*, 2018, pp. 603–619.
[228] Y. Wu and K. He, "Group normalization," in *Proc. ECCV*, 2018, pp. 3–19.
[229] W. Xiong *et al.*, "Foreground-aware image inpainting," in *Proc. CVPR*, 2019, pp. 5840–5848.
[230] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. NeurIPS*, 2016, pp. 91–99.
[231] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo, "ESTHER: Extremely simple image translation through self-regularization," in *Proc. BMVC*, 2018, pp. 91–99.
[232] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo, "Show, attend, and translate: Unsupervised image translation with self-regularization and attention," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4845–4856, Oct. 2019.
[233] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin, "Pose guided human video generation," in *Proc. ECCV*, 2018, pp. 201–216.
[234] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee, "Diversity-sensitive conditional generative adversarial networks," in *Proc. ICLR*, 2019, pp. 201–216.
[235] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. ICCV*, 2017, pp. 2849–2857.
[236] J. Yu, Y. Fan, and T. Huang, "Wide activation for efficient and accurate image super-resolution," in *Proc. BMVC*, 2019.
[237] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. CVPR*, 2018, pp. 5505–5514.
[238] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. ICCV*, 2019, pp. 4471–4480.
[239] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proc. ICCV*, 2019, pp. 9459–9468.
[240] M. Zanfir, A.-I. Popa, A. Zanfir, and C. Sminchisescu, "Human appearance transfer," in *Proc. CVPR*, 2018, pp. 5391–5399.
[241] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proc. CVPR*, 2019, pp. 1486–1494.
[242] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. ICML*, 2019, pp. 7354–7363.
[243] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[244] P. Zhang, B. Zhang, D. Chen, L. Yuan, and F. Wen, "Cross-domain correspondence learning for exemplar-based image translation," in *Proc. CVPR*, 2020, pp. 5143–5153.
[245] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in *Proc. WIFD*, 2019, pp. 1–6.
[246] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. ECCV*, 2018, pp. 286–301.
[247] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. CVPR*, 2018, pp. 2472–2481.
[248] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng, "Multi-view image generation from a single-view," in *Proc. MM*, 2018, pp. 383–391.
[249] C. Zheng, T.-J. Cham, and J. Cai, "Pluralistic image completion," in *Proc. CVPR*, 2019, pp. 1438–1447.
[250] H. Zheng, H. Liao, L. Chen, W. Xiong, T. Chen, and J. Luo, "Example-guided image synthesis across arbitrary scenes using masked spatial-channel attention and self-supervision," 2020, *arXiv:2004.10024*. [Online]. Available: http://arxiv.org/abs/2004.10024
[251] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *Proc. AAAI*, 2019, pp. 9299–9306.
[252] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, "Dance dance generation: Motion transfer for Internet videos," 2019, *arXiv:1904.00129*. [Online]. Available: http://arxiv.org/abs/1904.00129
[253] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. ECCV*, 2016, pp. 597–613.
[254] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, 2017, pp. 2223–2232.
[255] J.-Y. Zhu *et al.*, "Toward multimodal image-to-image translation," in *Proc. NeurIPS*, 2017, pp. 465–476.
[256] J.-Y. Zhu *et al.*, "Visual object networks: Image generation with disentangled 3D representations," in *Proc. NeurIPS*, 2018, pp. 118–129.
[257] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *Proc. CVPR*, 2019, pp. 2347–2356.

## ABOUT THE AUTHORS



**Ming-Yu Liu** received the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2012, advised by Prof. Rama Chellappa.

He was a Principal Research Scientist with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. He is currently a Distinguished Research Scientist and a Manager with NVIDIA Research, Santa Clara, CA, USA. His goal is to enable machines human-like imagination capability. His research interest is in generative image modeling.

Dr. Liu won the R&D 100 Award for his contribution to a commercial robotic bin picking system. His layered streetview semantic labeling paper was in the best paper finalists in the Robotics: Science and Systems (RSS) conference in 2015. He won the first place in both the Domain Adaptation for Semantic Segmentation Competition and the Robust Optical Flow Estimation Challenge in the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) in 2018. His SPADE paper was in the best paper finalists in the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) in 2019. In SIGGRAPH Real-Time Live 2019, he won both the Best in Show Award and the Audience Choice Award for his GauGAN demo. His GauGAN App further won the Best of What's New Award by the Popular Science Magazine in 2019. In 2021, his GAN for video compression work helps NVIDIA win the most disruptive innovator award by Forbes. He has served as the Area Chair for various computer vision and machine learning conferences, including Advances in Neural Information Processing Systems (NeurIPS), International Conference on Machine Learning (ICML), International Conference on Learning Representations (ICLR), Conference on Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), British Machine Vision Conference (BMVC), and IEEE Winter Conference on Applications of Computer Vision (WACV). He has served as the Program Chair of IEEE Winter Conference on Applications of Computer Vision (WACV) in 2020.

**Jiahui Yu** (Member, IEEE) received the bachelor's degree (honors) from the School of the Gifted Young in Computer Science, University of Science and Technology of China, Hefei, China, in 2016, and the Ph.D. degree from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2020.

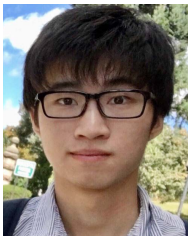
He is currently a Research Scientist with Google Brain, Mountain View, CA, USA. His research interest is in sequence modeling (language, speech, video, and financial data), machine perception (vision), generative models [generative adversarial networks (GANs)], and high-performance computing.

Dr. Yu is a member of Association for Computing Machinery (ACM) and National Conference on Artificial Intelligence (AAAI). He was a recipient of the Baidu Scholarship, the Thomas and Margaret Huang Research Award, and the Microsoft-IEEE Young Fellowship.

**Ting-Chun Wang** received the Ph.D. degree in electrical engineering and computer sciences (EECS) from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 2017, advised by Prof. Ravi Ramamoorthi and Alexei A. Efros.

He is currently a Senior Research Scientist with NVIDIA Research, Santa Clara, CA, USA. His research interests include computer vision, machine learning, and computer graphics, particularly the intersections of all three. His recent research focus is on using generative adversarial models to synthesize realistic images and videos, with applications to rendering, visual manipulations, and beyond.

Dr. Wang won the first place in the Domain Adaptation for Semantic Segmentation Competition in CVPR in 2018. His semantic image synthesis paper was in the best paper finalists in CVPR in 2019 and the corresponding GauGAN app won the Best in Show Award and Audience Choice Award in SIGGRAPH RealTimeLive in 2019. He has served as the Area Chair of WACV in 2020.



**Xun Huang** received the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 2020, under the supervision of Prof. Serge Belongie.

He is currently a Research Scientist with NVIDIA Research, Santa Clara, CA, USA. His research interests include developing new architectures and training algorithms of generative adversarial networks, as well as applications such as image editing and synthesis.

Dr. Huang was a recipient of the NVIDIA Graduate Fellowship, the Adobe Research Fellowship, and the Snap Research Fellowship.

**Arun Mallya** received the B.Tech. degree in computer science and engineering from IIT Kharagpur, Kharagpur, India, in 2012, and the M.S. degree in computer science and the Ph.D. degree, with a focus on performing multiple tasks efficiently with a single deep network, from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2014 and 2018, respectively.

He is currently a Senior Research Scientist with NVIDIA Research, Santa Clara, CA, USA. He is interested in generative modeling and enabling new applications of deep neural networks.

Dr. Mallya was selected as a Siebel Scholar in 2014.