On Leveraging Pretrained GANs for Generation with Limited Data

Miaoyun Zhao [* 1] **Yulai Cong** [* 1] **Lawrence Carin** [1]

Abstract

Recent work has shown generative adversarial networks (GANs) can generate highly realistic images, that are often indistinguishable (by humans) from real images. Most images so generated are not contained in the training dataset, suggesting potential for augmenting training sets with GAN-generated data. While this scenario is of particular relevance when there are limited data available, there is still the issue of training the GAN itself based on that limited data. To facilitate this, we leverage existing GAN models pretrained on large-scale datasets (like ImageNet) to introduce additional knowledge (which may not exist within the limited data), following the concept of transfer learning. Demonstrated by natural-image generation, we reveal that low-level filters (those close to observations) of both the generator and discriminator of pretrained GANs can be transferred to facilitate generation in a perceptually-distinct target domain with limited training data. To further adapt the transferred filters to the target domain, we propose adaptive filter modulation (AdaFM). An extensive set of experiments is presented to demonstrate the effectiveness of the proposed techniques on generation with limited data.

1. Introduction

Recent research has demonstrated the increasing power of generative adversarial networks (GANs) to generate high-quality samples, that are often indistinguishable from real data (Karras et al., 2017; Lucic et al., 2018; Miyato et al., 2018; Brock et al., 2019; Karras et al., 2019a); this demonstrates the capability of GANs to exploit the valuable information within the underlying data distribution.

Although many powerful GAN models pretrained on large-scale datasets have been released, few efforts have been made (Giacomello et al., 2019) to take advantage of the valuable information within those models to facilitate downstream tasks; this shows a clear contrast with the popularity of transfer learning for discriminative tasks (*e.g.,* to reuse the feature extractor of a pretrained classifier) (Bengio, 2012; Donahue et al., 2014; Luo et al., 2017; Zamir et al., 2018), and transfer learning in natural language processing (*e.g.,* to reuse the expensively-pretrained BERT model) (Devlin et al., 2018; Bao & Qiao, 2019; Peng et al., 2019; Mozafari et al., 2019).

Motivated by the significant value of released pretrained GAN models, we propose to leverage the information therein to facilitate downstream tasks in a target domain with limited training data. This situation arises frequently due to expensive data collection or privacy issues that may arise in medical or biological applications (Yi et al., 2019). We concentrate on the challenging scenario of GAN model development when limited training data are available. One key observation motivating our method is that a well-trained GAN can generate realistic images not observed in the training dataset (Brock et al., 2019; Karras et al., 2019a; Han et al., 2019), demonstrating the generalization ability of GANs to capture the training data manifold. Likely arising from novel combinations of information/attributes/styles (see stunning illustrations in StyleGAN (Karras et al., 2019a)), this generalization of GANs is extremely appealing for scenarios in which there are limited data (Yi et al., 2019; Han et al., 2019). For example, GANs can be used to augment the training set via realistic data generation, to alleviate overfitting or provide regularizations for classification (Wang & Perez, 2017; Frid-Adar et al., 2018), segmentation (Bowles et al., 2018), or detection (Han et al., 2019; 2020).

However, the limited data in the target domain manifests a problem in learning the underlying GAN model, as GANs typically require substantial training data. When a limited quantity of data are available, to naively train a GAN is prone to overfitting, as powerful GAN models have numerous parameters that are essential for realistic generation (Bermudez et al., 2018; Bowles et al., 2018; Frid-Adar et al., 2018; Finlayson et al., 2018). To alleviate overfitting, one may consider transferring additional information from other domains via transfer learning, which may deliver

---

[*]Equal contribution [1]Department of Electrical and Computer Engineering, Duke University, Durham NC, USA. Correspondence to: Miaoyun Zhao <miaoyun9zhao@gmail.com>, Yulai Cong <yulaicong@gmail.com>.

simultaneously better training efficiency and performance (Caruana, 1995; Bengio, 2012; Sermanet et al., 2013; Donahue et al., 2014; Zeiler & Fergus, 2014; Girshick et al., 2014). However, most transfer learning work has focused on discriminative tasks, based on the foundation that low-level filters (those close to *input* observations) of a classifier pretrained on a large-scale source dataset are fairly general (like Gabor filters) and thus transferable to different target domains (Yosinski et al., 2014); as the well-trained low-level filters (often data-demanding (Frégier & Gouray, 2019; Noguchi & Harada, 2019)) provide additional information, transfer learning often leads to better performance (Yosinski et al., 2014; Long et al., 2015; Noguchi & Harada, 2019). Compared to transfer learning on discriminative tasks, fewer efforts have been made for generation tasks (Shin et al., 2016b; Wang et al., 2018b; Noguchi & Harada, 2019), as summarized in Section 2. The work presented here addresses this challenge, considering transfer learning for GANs when there are limited data in the target domain.

Leveraging insights from the aforementioned transfer learning on discriminative tasks, we posit that the low-level filters of a GAN discriminator pretrained on a large-scale source dataset are likely to be generalizable and hence transferable to various target domains. For a pretrained GAN generator, it's shown (Bau et al., 2017; 2018; Karras et al., 2019a) that the low-level layers (those close to *output* observations) capture properties of generally-applicable local patterns like materials, edges, and colors, while the high-level layers (those distant from observations) are associated with more domain-specific semantic aspects of data. We therefore consider transferring/freezing the low-level filters from both the generator and discriminator of a pretrained GAN model to facilitate generation in perceptually-distinct target domains with limited training data. As an illustrative example, we consider the widely studied GAN scenario of natural-image generation, although the proposed techniques are general and may be applicable to other domains, such as in medicine or biology. The principal contributions of this paper are as follows.

- We demonstrate empirically that the low-level filters (within both the generator and the discriminator) of a GAN model, pretrained on large-scale datasets, can be transferred to perceptually-distinct target domains, yielding improved GAN performance in scenarios for which limited training data are available.
- We tailor a compact domain-specific network to harmoniously cooperate with the transferred low-level filters, which enables style mixing for diverse synthesis.
- To better adapt the transferred filters to the target domain, we introduce an approach we term adaptive filter modulation (AdaFM), that delivers boosted performance.
- Extensive experiments are conducted to verify the effectiveness of the proposed techniques.

2. Background and Related Work

2.1. Generative Adversarial Networks (GANs)

GANs have demonstrated increasing power for synthesizing highly realistic data (Brock et al., 2019; Karras et al., 2019a;b); accordingly, they are widely applied in various research fields, such as image (Hoffman et al., 2017; Ledig et al., 2017; Ak et al., 2019), text (Lin et al., 2017; Fedus et al., 2018; Wang & Wan, 2018; Wang et al., 2019), video (Mathieu et al., 2015; Wang et al., 2017; 2018a; Chan et al., 2019), and audio (Engel et al., 2019; Yamamoto et al., 2019; Kumar et al., 2019).

A GAN often consists of two adversarial components, *i.e.,* a generator G and a discriminator D. As the adversarial game proceeds, the generator learns to synthesize increasingly realistic fake data, to confuse the discriminator; the discriminator seeks to discriminate real and fake data synthesized by the generator. The standard GAN objective (Goodfellow et al., 2014) is

$$\min_{G} \max_{D} \mathbb{E}_{\boldsymbol{x} \sim q_{\text{data}}(\boldsymbol{x})} \big[\log D(\boldsymbol{x}) \big] \\ + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})} \big[\log(1 - D(G(\boldsymbol{z}))) \big], \quad (1)$$

where $p(\boldsymbol{z})$ is an easy-to-sample distribution, like a normal distribution, and $q_{\text{data}}(\boldsymbol{x})$ is the empirical data distribution.

2.2. GANs on Limited Data

Existing work addressing the design of GANs based on limited data can be roughly summarized into two groups.

Exploit GANs for better usage of the information within the limited data. In addition to traditional data augmentations like shift, zooming, rotation, or flipping, GANs trained on limited data can be leveraged for synthetic augmentations, like synthesized observations with transformed styles (Wang & Perez, 2017) or fake observation-label/segmentation pairs (Bowles et al., 2018; Frid-Adar et al., 2018; Han et al., 2019; 2020). However, because of the limited available data, a relatively small GAN model is often employed, leading to reduced generative power. Furthermore, only the information within the limited data are utilized.

Use GANs to transfer additional information to facilitate generation with limited data. As the available data are limited, it's often preferred to transfer additional information from other domains via transfer learning (Yosinski et al., 2014; Long et al., 2015; Noguchi & Harada, 2019). TransferGAN (Wang et al., 2018b) initializes the target GAN model with parameters pretrained on source large-scale datasets, followed by fine-tuning the whole model with the limited target data. As source model architecture (often large) is directly transferred to the target domain, fine-tuning with too limited target data may suffer from overfitting, as verified empirically in our experiments; since
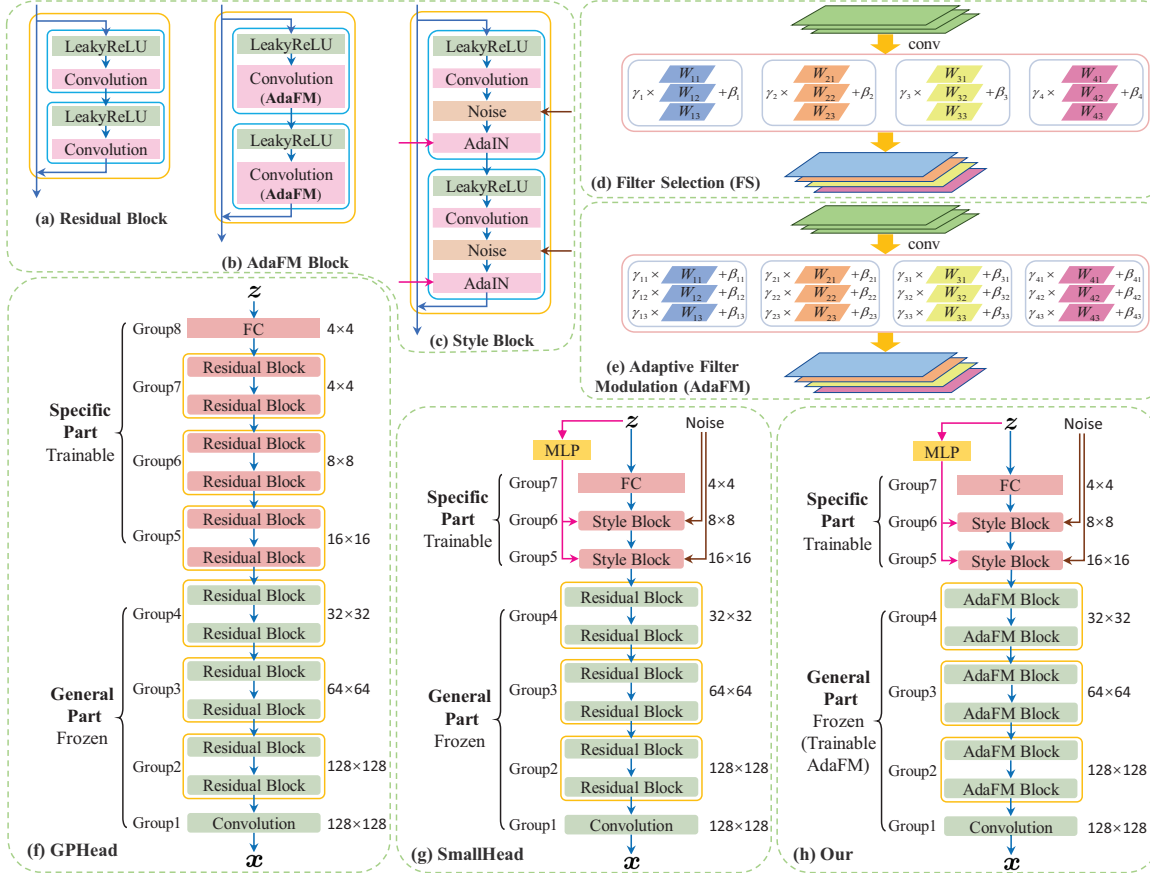
(a) Residual Block — LeakyReLU, Convolution, LeakyReLU, Convolution

(b) AdaFM Block — LeakyReLU, Convolution (AdaFM), LeakyReLU, Convolution (AdaFM)

(c) Style Block — LeakyReLU, Convolution, Noise, AdaIN, LeakyReLU, Convolution, Noise, AdaIN

(d) Filter Selection (FS) — conv; $\gamma_1 \times \begin{smallmatrix}W_{11}\\W_{12}\\W_{13}\end{smallmatrix} + \beta_1$, $\gamma_2 \times \begin{smallmatrix}W_{21}\\W_{22}\\W_{23}\end{smallmatrix} + \beta_2$, $\gamma_3 \times \begin{smallmatrix}W_{31}\\W_{32}\\W_{33}\end{smallmatrix} + \beta_3$, $\gamma_4 \times \begin{smallmatrix}W_{41}\\W_{42}\\W_{43}\end{smallmatrix} + \beta_4$

(e) Adaptive Filter Modulation (AdaFM) — conv; $\gamma_{11}\times W_{11}+\beta_{11}$, $\gamma_{12}\times W_{12}+\beta_{12}$, $\gamma_{13}\times W_{13}+\beta_{13}$, $\gamma_{21}\times W_{21}+\beta_{21}$, $\gamma_{22}\times W_{22}+\beta_{22}$, $\gamma_{23}\times W_{23}+\beta_{23}$, $\gamma_{31}\times W_{31}+\beta_{31}$, $\gamma_{32}\times W_{32}+\beta_{32}$, $\gamma_{33}\times W_{33}+\beta_{33}$, $\gamma_{41}\times W_{41}+\beta_{41}$, $\gamma_{42}\times W_{42}+\beta_{42}$, $\gamma_{43}\times W_{43}+\beta_{43}$

(f) GPHead — z, FC 4×4 (Group8); Specific Part Trainable: Residual Block 4×4 (Group7), Residual Block 8×8 (Group6), Residual Block 16×16 (Group5); General Part Frozen: Residual Block 32×32 (Group4), Residual Block 64×64 (Group3), Residual Block 128×128 (Group2), Convolution 128×128 (Group1); x

(g) SmallHead — z, Noise, MLP; Specific Part Trainable: FC 4×4 (Group7), Style Block 8×8 (Group6), Style Block 16×16 (Group5); General Part Frozen: Residual Block 32×32 (Group4), Residual Block 64×64 (Group3), Residual Block 128×128 (Group2), Convolution 128×128 (Group1); x

(h) Our — z, Noise, MLP; Specific Part Trainable: FC 4×4 (Group7), Style Block 8×8 (Group6), Style Block 16×16 (Group5); General Part Frozen (Trainable AdaFM): AdaFM Block 32×32 (Group4), AdaFM Block 64×64 (Group3), AdaFM Block 128×128 (Group2), Convolution 128×128 (Group1); x

Figure 1. Network architectures. A "group" in general contains blocks with the same feature-map size. The multilayered perceptron (MLP) consists of 8 fully-connected (FC) layers. With a 64-dimensional z, the number of trainable parameters is 79.7M for GPHead, 23.0M for SmallHead, and 24.4M for Our model.

the high-level semantically specific filters are also transferred, the similarity between the source and target domains is often critical for a beneficial transfer (Wang et al., 2018b). Similarly, based on the assumption that the source and target domains share the same support, Wang et al. (2020) introduces an additional miner network to mine knowledge from pretrained GANs to form target generation, likewise fine-tuning the whole model with the limited data.

Different from the above fine-tuning methods, Noguchi & Harada (2019) propose batch-statistics adaptation (BSA) to transfer/freeze the whole source generator but introduce new trainable parameters to adapt its hidden batch statistics for generation with extremely-limited target data; however, the generator is not adversarially trained (L1/Perceptual loss is used instead), leading to blurry generation in the target domain (Noguchi & Harada, 2019). By comparison, our method transfers/freezes the generally-applicable low-level filters – in the generator and discriminator – from source to (perceptually-distinct) target domains, followed by employing a small tailored high-level network and the newly-introduced adaptive filter modulation (AdaFM) to better adapt to the target limited data. Accordingly, our

proposed method, when compared to the fine-tuning methods (Wang et al., 2018b; 2020), is expected to suffer less from overfitting and behave more robustly in the presence of differences between the source and target domains; when compared to (Noguchi & Harada, 2019), our method is more flexible and provides clearly better generation, thanks to its adversarial training.

Recently, a concurrent work (Mo et al., 2020) reveals that freezing low-level layers of the GAN discriminator delivers better fine-tuning of GANs; it may be viewed as a special case of our method, which transfers/freezes low-level layers of *both* the generator and discriminator, leveraging a tailored high-level network and employing the proposed AdaFM. Consequently, our method is expected to perform better on (extremely) limited data.

3. Proposed Method

For GAN training with limited data in the target domain, we propose to transfer additional information by leveraging the valuable low-level filters (those close to observations) from existing GANs pretrained on large-scale source

datasets. Combining that prior knowledge, from the transferred low-level filters that are often generally-applicable but data-demanding to train (Yosinski et al., 2014; Frégier & Gouray, 2019), one may expect less overfitting and hence better GAN performance. Specifically, given a pretrained GAN model, we reuse its low-level filters (termed the *generalizable or general part* of the model) in a target domain, and replace the high-level layers (termed the domain *specific part*) with another smaller network, and then train that specific part using the limited target data, while keeping the transferred general part frozen (see Figures 1(f) and (g)). Hence, via this approach, we leverage the transferred general part, trained on a much larger source dataset, and by employing the simplified domain-specific part, the total number of parameters that need be learned is reduced substantially, aligning with the limited target-domain data.

In what follows, we take natural image generation as an example, and present our method by answering three questions in Sections 3.1, 3.2, and 3.3, respectively:

- How to specify the general part appropriate for transfer?
- How to tailor the specific part so that it is simplified?
- How to better adapt the transferred general part?

Before introducing the proposed techniques in detail, we first discuss source datasets, available pretrained GAN models, and evaluation metrics. Intuitively, to realize generally-applicable low-level filters, one desires a large-scale source dataset with rich diversity. In the context of image analysis, a common choice is the ImageNet dataset (Krizhevsky et al., 2012; Shin et al., 2016a), which contains 1.2 million high-resolution images from 1000 classes; we consider this as the source dataset. Concerning publicly available GAN models pretrained on ImageNet, available choices include SNGAN (Miyato et al., 2018), GP-GAN (Mescheder et al., 2018), and BigGAN (Brock et al., 2019); we select the pretrained GP-GAN model (with resolution 128×128) because of its well-written codebase. To evaluate the generative performance, we adopt the widely used Fréchet inception distance (FID, lower is better) (Heusel et al., 2017), a metric assessing the realism and variation of generated samples (Zhang et al., 2018).

3.1. On Specifying the General Part for Transfer

As mentioned in the Introduction, both generative and discriminative image models share a similar pattern: higher-level convolutional filters portray more domain-specific semantic information, while lower-level filters portray more generally applicable information (Yosinski et al., 2014; Zeiler & Fergus, 2014; Bau et al., 2017; 2018). Given the GP-GAN model pretrained on ImageNet, the question is how to specify the low-level filters (*i.e.,* the general part of the model) to be transferred to a target domain. Generally speaking, the optimal solution is likely to be a compromise depending on the available target-domain data; if plenty

Figure 2. Sample images from the ImageNet and CelebA datasets. Although quite different, they are likely to share the same set of low-level filters describing basic shapes, like lines, curves and textures.

of data are provided, less low-level filters should be transferred (less prior knowledge need be transferred), but when the target-domain data are limited, it's better to transfer more filters (leveraging more prior information). We empirically address that question by transferring the pretrained GP-GAN model to the CelebA dataset (Liu et al., 2015), which is fairly different from the source ImageNet (see Figure 2). It's worth emphasizing that the general part discovered here[1] also delivers excellent results on three other datasets (see the experiments and Appendix E).

3.1.1. GENERAL PART OF THE GENERATOR

To determine the appropriate general part of the GP-GAN generator, to be transferred to the target CelebA dataset,[2] we employ the GP-GAN architecture and design experiments with increasing number of lower layers included in the transferred/frozen general part of the generator; the remaining specific part (not transferred) of the generator (see Figure 1(f)) and the discriminator are reinitialized and trained with CelebA.

Four settings for the generator general part are tested, *i.e.,* 2, 4, 5, and 6 lower groups to be transferred (termed G2, G4, G5, and G6, respectively; G4 is illustrated in Figure 1(f)). After 60,000 training iterations (generative quality stabilizes by then), and we show in Figure 3 the generated samples and FIDs of the four settings. It's clear that transferring the G2/G4 generator general part delivers decent generative quality (see eye details, hair texture, and cheek smoothness), despite the fact that the source ImageNet is perceptually distant from the target CelebA, confirming the generalizable

---

[1]This general part may not be the optimum with the best generalization, which is deemed intractable. The key is that it's applicable to various target domains (see the experiments); in addition, the AdaFM technique introduced in Section 3.3 delivers significantly improved adaptation/transfer and thus greatly relaxes the requirement of an optimal selection of the general part.

[2]To verify the generalization of the pretrained filters, we bypass the limited-data assumption in this section and employ the whole CelebA data for training.

G2D0
(22.33)

G4D0
(13.12)

G5D0
(15.20)

G6D0
(22.98)

Figure 3. Generated samples and FIDs from different settings for the general part of the generator. GmDn indicates freezing the lower m/n groups as the general part of generator/discriminator.

nature of the low-level filters within up to 4 lower groups of the pretrained GP-GAN generator (which is also verified on three other datasets in the experiments). The lower FID of G4 than that of G2 indicates that transferring more low-level filters pretrained on large-scale source datasets potentially benefits better performance in target domains.[3] But when we transfer and hence freeze more groups as the general part of the generator (*i.e.,* G5 and G6), the generative quality drops quickly; this is expected as higher-level filters are more specific to the source ImageNet and may not fit the target CelebA. By reviewing Figure 3, we choose G4 as the setting for the generator general part for transfer.

3.1.2. GENERAL PART OF THE DISCRIMINATOR

Based on the G4 general part of the generator, we next conduct experiments to specify the general part of the discriminator. We consider transferring/freezing 0, 2, 3, and 4 lower groups of the pretrained GP-GAN discriminator (termed D0, D2, D3, and D4, respectively; D2 is illustrated in Figure 15 of the Appendix). Figure 4 shows the generated samples and FIDs for each setting. Similar to what's observed for the generator, transferring low-level filters from the pretrained GP-GAN discriminator also benefits a better generative performance (compare the FID of D0 with that of D2), thanks to the additional information therein; however, as the higher-level filters are more specific to the source ImageNet, transferring them may lead to a decreased generative quality (see the results from D3 and D4).

Considering both the generator and discriminator, we transfer/freeze the G4D2 general part[4] from the pretrained GP-

---

[3]Another reason might be that to train well-behaved low-level filters is time-consuming and data-demanding (Frégier & Gouray, 2019; Noguchi & Harada, 2019). The worse FID of G2 is believed caused by the insufficiently trained low-level filters, as we find the images from G2 show a relatively lower diversity and contain strange textures in the details (see Figure 18 in Appendix). FID is biased toward texture rather than shape (Karras et al., 2019b).

[4]Appendix C.2 discusses other settings for the general part.

G4D0
(13.12)

G4D2
(11.14)

G4D3
(13.99)

G4D4
(25.08)

Figure 4. Generated samples and FIDs from different settings for the general part of the discriminator.

GPHead   SmallHead

Figure 5. Generated images from the GPHead and SmallHead trained on the Flowers dataset (Nilsback & Zisserman, 2008).

GAN model, which will be shown in the experiments to work quite well on three other target datasets.

3.2. On Tailoring the High-Level Specific Part

Even with the transferred/frozen G4D2 general part, the remaining specific (target-dependent) part may contain too many trainable parameters for the limited target-domain data (*e.g.,* the GPHead model in Figure 1(f) shows mode collapse (see Figure 5) when trained on the small Flowers dataset (Nilsback & Zisserman, 2008)); another consideration is that, when using GANs for synthetic augmentation for applications with limited data, style mixing is a highly appealing capability (Wang & Perez, 2017). Motivated by those considerations, we propose to replace the high-level specific part of GP-GAN with a tailored smaller network (see Figure 1(g)), to alleviate overfitting, enable style mixing, and also lower the computational/memory cost.

Specifically, that tailored specific part is constructed as a fully connected (FC) layer followed by two successive style blocks (borrowed from StyleGAN (Karras et al., 2019a) with an additional short cut, see Figure 1(c)). Similar to Style-GAN, the style blocks enable unsupervised disentanglement of high-level attributes, which may benefit an efficient exploration of the underlying data manifold and thus lead to better generation; they also enable generating samples with new attribute combinations (style mixing), which dramatically enlarges the generation diversity (see Figures 14 and 28 of the Appendix). We term the model consisting of the tailored specific part and the G4D2 general part as SmallHead. Note
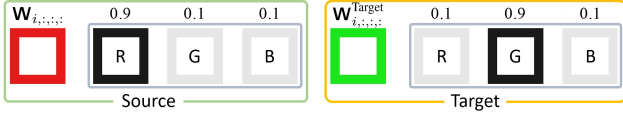
Figure 6. An illustrative example motivating AdaFM. Both the source and target domains share the same basic shape within each channel but use a different among-channel correlation. AdaFM learns $\boldsymbol{\gamma}_{i,:} = [1/9, 9, 1]$ to adapt source $\mathbf{W}_{i,:,:,:}$ to target $\mathbf{W}^{\text{Target}}_{i,:,:,:}$.

that proposed specific part is also used in our method (see Figure 1(h)). Different from the GPHead, the SmallHead has proven to train in a stable manner, without mode collapse on Flowers (see Figure 5). In the experiments, the SmallHead is found to work well on other small datasets.

3.3. Better Adaption of the Transferred General Part

Based on the above transferred general part and tailored specific part, we next present a new technique, termed adaptive filter modulation (AdaFM), to better adapt the transferred low-level filters to target domains for boosted performance, as shown in the experiments. In this proposed approach, we no longer just "freeze" the transferred filters upon transfer, but rather augment them in a target-dependent manner.

Motivated by the style-transfer literature (Huang & Belongie, 2017; Noguchi & Harada, 2019), where one manipulates the style of an image by modifying the statistics (*e.g.*, mean or variance) of its latent feature maps, we alternatively consider the variant of manipulating the style of a function (*i.e.,* the transferred general part) by modifying the statistics of its convolutional filters via AdaFM.

Specifically, given a transferred convolutional filter $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times K_1 \times K_2}$ in the general part, where $C_{\text{in}}/C_{\text{out}}$ denotes the number of input/output channels and $K_1 \times K_2$ is the kernel size, AdaFM introduces a small amount of learnable parameters, *i.e.,* scale $\boldsymbol{\gamma} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ and shift $\boldsymbol{\beta} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$, to modulate its statistics via

$$\mathbf{W}^{\text{AdaFM}}_{i,j,:,:} = \boldsymbol{\gamma}_{i,j} \mathbf{W}_{i,j,:,:} + \boldsymbol{\beta}_{i,j}, \qquad (2)$$

for $i \in \{1, 2, \cdots, C_{\text{out}}\}$ and $j \in \{1, 2, \cdots, C_{\text{in}}\}$. $\mathbf{W}^{\text{AdaFM}}$ is then used to convolve with input feature maps for output ones. Applying AdaFM to convolutional kernels of a residual block (see Figure 1(a)) (He et al., 2016) gives the AdaFM block (see Figure 1(b)). With the residual blocks of the SmallHead replaced with AdaFM blocks, we yield our generator, as shown in Figure 1(h), which delivers boosted performance than the SmallHead in the experiments.

For better understanding the power of our AdaFM, below we draw parallel connections to two related techniques, which may be viewed as the special cases of AdaFM to some extent. The first one is the weight demodulation revealed in the recent StyleGAN2 (Karras et al., 2019b), a model with state-of-the-art generative performance. Compared with
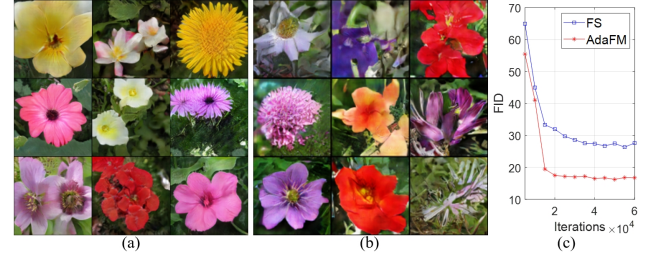

Figure 7. Generated samples from our model (a) with AdaFM, and (b) with AdaFM replaced by FS. (c) FID scores along training.

AdaFM, the weight demodulation employs zero shift $\boldsymbol{\beta} = 0$ and a rank-one scale $\boldsymbol{\gamma} = \boldsymbol{\eta} \boldsymbol{s}^T$, where style $\boldsymbol{s} \in \mathbb{R}^{C_{\text{in}}}$ is produced by a trainable mapping network (often a MLP) and $\boldsymbol{\eta} \in \mathbb{R}^{C_{\text{out}}}$ is calculated as

$$\boldsymbol{\eta}_i = \frac{1}{\sqrt{\epsilon + \sum_{j,k_1,k_2} \boldsymbol{s}_j \mathbf{W}_{i,j,k_1,k_2}}}, \qquad (3)$$

where ϵ is a small constant to avoid numerical issues. Despite being closely related, AdaFM and the weight demodulation are motivated differently. We propose AdaFM to better adapt the transferred filters to target domains, while the weight demodulation is used to relax instance normalization while keeping the capacity for controllable style mixing (Karras et al., 2019b). See Appendix C.7 for more discussions.

Another special case of AdaFM is the filter selection (FS) presented in (Noguchi & Harada, 2019), which employs rank-one simplification to both scale $\boldsymbol{\gamma}$ and shift $\boldsymbol{\beta}$. Specifically, $\boldsymbol{\gamma} = \hat{\boldsymbol{\gamma}} \mathbf{1}^T$ and shift $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}} \mathbf{1}^T$ with $\hat{\boldsymbol{\gamma}} \in \mathbb{R}^{C_{\text{out}}}$ and $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{C_{\text{out}}}$ (see Figure 1(d)). The goal of FS is to "select" the transferred filters $\mathbf{W}$; for example if $\hat{\boldsymbol{\gamma}}$ is a binary vector, it literally selects among $\{\mathbf{W}_{i,:,:,:}\}$. As FS doesn't modulate among C_{in} input channels, its basic assumption is that the source and target domains share the same correlation among those channels, which might not be true. See the illustrative example in Figure 6, where the source/target domain has the basic pattern/filter of an almost-red/almost-green square (the shape is the same); it's clear simply selecting (FS) the source filter won't deliver a match to the target, appealing for a modulation (AdaFM) among (input) channels. Figure 7 shows results from models with FS and AdaFM (using the same architecture in Figure 1(h)). It's clear that AdaFM brings boosted performance, empirically supporting the above intuition that the basic shape/pattern within each $\mathbf{W}_{i,j,:,:}$ are generally applicable while the correlation among i-,j-channels may be target-specific (this is further verified by AdaFM delivering boosted performance on other datasets in the experiments).

4. Experiments

Taking natural image generation as an illustrative example, we demonstrate the effectiveness of the proposed techniques
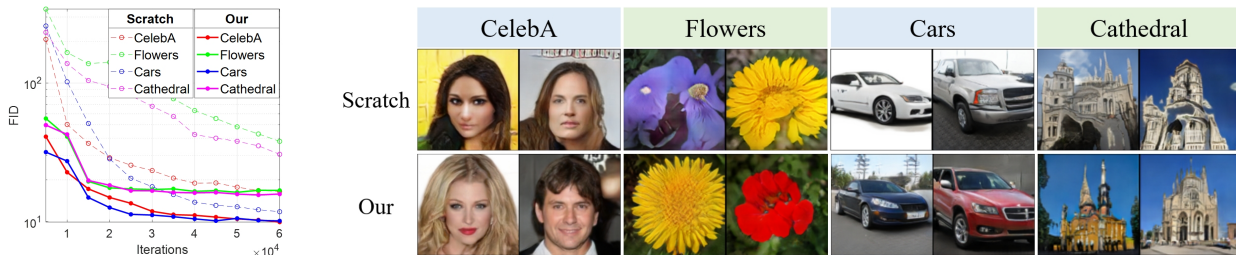
Figure 8. FID scores (left) and generated images (right) of Scratch and our method on 4 target datasets. The transferred general part from the pretrained GP-GAN model dramatically accelerates the training, leading to better performance.

by transferring the source GP-GAN model pretrained on the large-scale ImageNet (containing 1.2 million images from 1,000 classes) to facilitate generation in perceptually-distinct target domains with (*i*) four smaller datasets, *i.e.,* CelebA (Liu et al., 2015) (202,599), Flowers (Nilsback & Zisserman, 2008) (8,189), Cars (Krause et al., 2013) (8,144), and Cathedral (Zhou et al., 2014) (7,350); (*ii*) their modified variants containing only 1,000 images; and (*iii*) two extremely limited datasets consisting of 25 images (following (Noguchi & Harada, 2019)).

The experiments proceed by (*i*) demonstrating the advantage of our method over existing approaches; (*ii*) conducting ablation studies to analyze the contribution of each component of our method; (*iii*) verifying the proposed techniques in challenging settings with only 1,000 or 25 target images; (*iv*) analyzing why/how AdaFM leads to boosted performance; and (*v*) illustrating the potential for exploiting the tailored specific part of our model for data augmentation for applications with limited data. Generated images and FID scores (Heusel et al., 2017) are used to evaluate the generative performance. Detailed experimental settings and more results are provided in the Appendix. Code is available at github.com/MiaoyunZhao/GANTransferLimitedData.

4.1. Comparisons with Existing Methods

To demonstrate our contributions over existing approaches, we compare our method with (*i*) TransferGAN (Wang et al., 2018b), which initializes with the pretrained GP-GAN model (accordingly the same network architecture is adopted; refer to Figure 1(f)), followed by fine-tuning all parameters on the target data. We also consider (*ii*) Scratch,

Table 1. FID scores of the compared methods after 60,000 training iterations. Lower is better. "Failed" means training/mode collapse.

Method\Target	CelebA	Flowers	Cars	Cathedral
TransferGAN	18.69	failed	failed	failed
Scratch	16.51	29.65	11.77	30.59
Our	**9.90**	**16.76**	**10.10**	**15.78**

which trains a model with the same architecture as ours (see Figure 1(h)) from scratch with the target data.

The experimental results are shown in Figure 8, with the final FID scores summarized in Table 1. Since TransferGAN employs the source (large) GP-GAN architecture, it may suffer from overfitting if the target data are too limited, which manifests as training/mode collapse; accordingly, TransferGAN fails on the 3 small datasets: Flowers, Cars, and Cathedral. By comparison, thanks to the tailored specific part, both Scratch and our method train stably on all target datasets, as shown in Figure 8. Compared to Scratch, our method shows dramatically increased training efficiency, thanks to the transferred low-level filters, and significantly improved generative quality (much better FIDs in Table 1), which are attributed to both the transferred general part and a better adaption to target domains with AdaFM.

4.2. Ablation Study of Our Method

To reveal how each component contributes to the excellent performance of our method, we consider four experimental settings in a sequential manner. *(a)* **GP-GAN**: adopt the GP-GAN architecture (similar to Figure 1(f) but all parameters are trainable and randomly initialized), used as a baseline where no low-level filters are transferred. *(b)* **GPHead**: use the model in Figure 1(f), to demonstrate the contribution
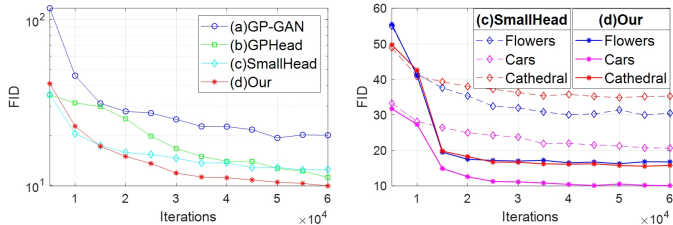

Figure 9. FID scores from the ablation studies of our method on CelebA (left) and the 3 small datasets: Flower, Cars, and Cathedral (right).

Table 2. FID scores from ablation studies of our method after 60,000 training iterations. Lower is better.

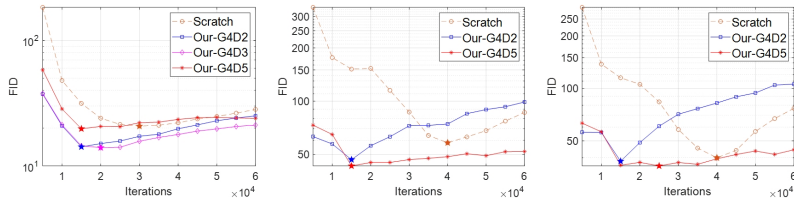| Method\Target	CelebA	Flowers	Cars	Cathedral
(a): GP-GAN	19.48	failed	failed	failed
(b): GPHead	11.15	failed	failed	failed
(c): SmallHead	12.42	29.94	20.64	34.83
(d): Our	**9.90**	**16.76**	**10.10**	**15.78**

Figure 10. FID scores on CelebA-1K (left), Flowers-1K (center), and Cathedral-1K (right). The best FID achieved is marked with a star.

Table 3. The best FID achieved within 60,000 training iterations on the limited-1K datasets. Lower is better.

Method\Target	CelebA-1K	Flowers-1K	Cathedral-1K
Scratch	20.75	58.18	39.97
Our-G4D2	14.19	46.68	38.17
Our-G4D3	**13.99**	-	-
Our-G4D5	19.77	**43.05**	**35.88**

of the transferred general part. *(c)* **SmallHead**: employ the model in Figure 1(g), to reveal the contribution of the tailored specific part. *(d)* **Our**: leverage the model in Figure 1(h), to show the contribution of the presented AdaFM.

The FID curves during training and the final FID scores of the compared methods are shown in Figure 9 and Table 2, respectively. By comparing GP-GAN with GPHead on CelebA, it's clear that the transferred general part contributes by dramatically increasing the training efficiency and by delivering better generative performance; this is consistent with what's revealed in the previous section (compare Scratch with Our in Figure 8 and Table 1). Comparing SmallHead to both GPHead and GP-GAN in Table 2 indicates that the tailored specific part helps alleviate overfitting and accordingly delivers stable training. By better adapting the transferred general part to the target domains, the proposed AdaFM contributes most to the boosted performance (compare SmallHead with Our in Figure 9 and Table 2), empirically confirming our intuition in Section 3.3.

4.3. Generation with Extremely Limited Data

To verify the effectiveness of the proposed techniques in more challenging settings, we consider generation with only 1,000 or 25 target samples. Specifically, we randomly select 1,000 images from CelebA, Flowers, and Cathedral to form their limited-1K variants, termed CelebA-1K, Flowers-1K, and Cathedral-1K, respectively. Since TransferGAN fails

when given about 8,000 target images (see Section 4.1), we omit it and only compare our method with Scratch on these 1K variants. Regarding the extremely limited setup with 25 samples, we follow Noguchi & Harada (2019) to select 25 images from Flowers and FFHQ (Karras et al., 2019a) to form the Flowers-25 and FFHQ-25 datasets, on which their BSA and our method are compared.

The FID curves versus training iterations on the 1K datasets are shown in Figure 10, with the lowest FIDs summarized in Table 3. In this challenging setting, both Scratch and our method with the G4D2 general part (labeled Our-G4D2) suffer from overfitting. Scratch suffers more due to more trainable parameters; as our method has a much higher training efficiency, a false impression may potentially arise (Cong et al., 2019; Mo et al., 2020); for clarity, see the significantly improved best performance of our method and the more gentle ending slope of its FID curves. To alleviate overfitting, we transfer more discriminator filters from the pretrained GP-GAN model, with the results also given in Figure 10 and Table 3. It's clear that intuitive patterns emerge, *i.e.,* less data appeal for more transferred information. On the other hand, the comparable FIDs of Our-G4D2 (see Table 3) indicates that the G4D2 general part discovered in Section 3.1 works fairly well on these 1K datasets. Concerning early stopping for generation with limited data, we empirically find that the discriminator loss may be leveraged for that goal in our setup (see Appendix C.5 for detailed
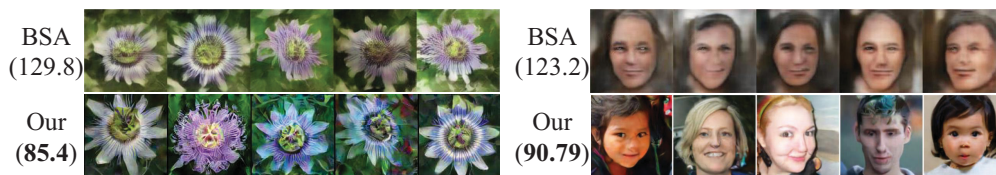

Figure 11. Generated images and FID scores of the compared methods on Flowers-25 (left) and FFHQ-25 (right). The BSA results are copied from the original paper (Noguchi & Harada, 2019).

Figure 12. Interpolations between two random samples from our method on Flowers-25 (first row) and FFHQ-25 (second row).
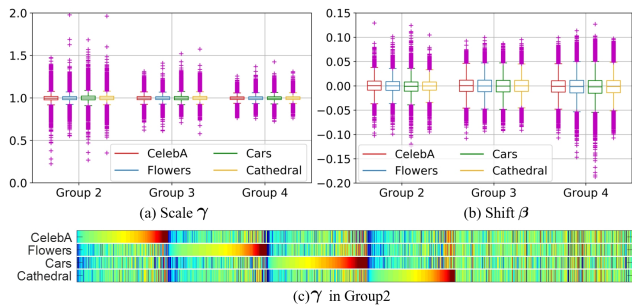
Figure 13. Boxplots of the learned scale γ (a) and shift β (b) within each group of the generator general part. (c) Sorted comparison of the γ learned on different datasets (see Appendix C.6 for details).

discussions).

On Flowers-25 and FFHQ-25, since the target data are extremely limited in quantity, we transfer more filters (*i.e.,* G4D6) from the pretrained GP-GAN model and apply GP (gradient penalty) on both real and fake samples to alleviate overfitting (see Appendix B for detailed settings). Figure 11 shows the generated samples and FID scores from the BSA (Noguchi & Harada, 2019) and our method. It's clear that our method with the G4D6 general part works reasonably well even in such settings with extremely limited data, with a much better performance than the L1/Perceptual-loss-based BSA. To illustrate the learned data manifold, Figure 12 shows the smooth interpolations between two random samples from our method, demonstrating the effectiveness of the proposed techniques on generation with extremely limited data.

4.4. Analysis of AdaFM and Style Augmentation with the Tailored Specific Part

To better understand why adopting AdaFM in the transferred general part of our model leads to boosted performance, we summarize in Figures 13(a) and 13(b) the learned scale γ and shift β from different groups of the generator general part. Apparently, all transferred filters are used in the target domains (no zero-valued γ) but with modulations (γ/β has values around $1/0$). As AdaFM delivers boosted performance, it's clear those modulations are crucial for a successful transfer from source to target domain, confirming our intuition in Section 3.3. To illustrate how γ/β behaves on different target datasets, we show in Figure 13(c) the sorted comparisons of the learned γ in Group 2; as expected, different datasets prefer different modulations, justifying the necessity of AdaFM and its performance gain. Concerning further demonstration of AdaFM and medical/biological applications with gray-scale images, we conduct another experiment on a gray-scale variant of Cathedral (results are given in Appendix D due to space constraints); we find that without AdaFM to adapt the transferred filters, worse (blurry and messy) details are observed in the generated images (refer also to Figure 7), likely because of the mis-
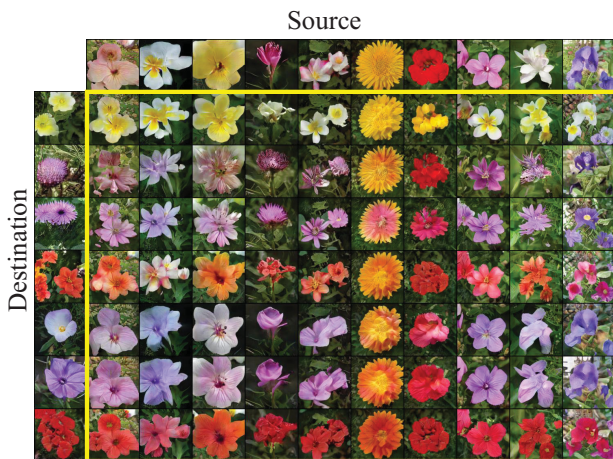

Figure 14. Style mixing on Flowers via the tailored specific part of our model. The "Source" sample controls flower shape, location, and background, while the "Destination" sample controls color and petal details.

matched correlation among channels between source and target domains.

To reveal the potential in exploiting the tailored specific part of our model for data augmentation for applications with limited data, we conduct style mixing with the specific part, following (Karras et al., 2019a). Figure 14 shows the results on Flowers (see Appendix F for details and more results). It's clear that style mixing enables synthesizing a vast set of new images via style/attribute combinations. Therefore, the tailored specific part of our model can be exploited for a diverse synthetic augmentation, which is believed extremely appealing for downstream applications with limited data.

5. Conclusions

We reveal that the valuable information (*i.e.,* the low-level filters) within GAN models pretrained on large-scale source datasets (*e.g.,* ImageNet) can be transferred to facilitate generation in perceptually-distinct target domains with limited data; this transfer is performed on both the generator and discriminator. To alleviate overfitting due to the limited target-domain data, we employ a small specific network atop the transferred low-level filters, which enables style mixing for a diverse synthetic augmentation. To better adapt the transferred filters to target domains, we present adaptive filter modulation (AdaFM), that delivers boosted performance on generation with limited data. The proposed techniques are shown to work well in challenging settings with extremely limited data (*e.g.,* 1,000 or 25 samples).

Acknowledgements

References

Ak, K., Lim, J., Tham, J., and Kassim, A. Attribute manipulation generative adversarial networks for fashion images. In *ICCV*, pp. 10541–10550, 2019.

Bao, X. and Qiao, Q. Transfer learning from pre-trained bert for pronoun resolution. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pp. 82–88, 2019.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, pp. 6541–6549, 2017.

Bau, D., Zhu, J., Strobelt, H., Zhou, B., Tenenbaum, J., Freeman, W., and Torralba, A. GAN dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, 2012.

Bermudez, C., Plassard, A., Davis, L., Newton, A., Resnick, S., and Landman, B. Learning implicit brain MRI manifolds with deep learning. In *Medical Imaging 2018: Image Processing*, volume 10574, pp. 105741L. International Society for Optics and Photonics, 2018.

Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D., Hernández, M., Wardlaw, J., and Rueckert, D. GAN augmentation: augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Caruana, R. Learning many related tasks at the same time with backpropagation. In *NIPS*, pp. 657–664, 1995.

Chan, C., Ginosar, S., Zhou, T., and Efros, A. Everybody dance now. In *CVPR*, pp. 5933–5942, 2019.

Cong, Y., Zhao, M., Bai, K., and Carin, L. GO gradient for expectation-based objectives. In *ICLR*, 2019. URL https://openreview.net/forum?id=ryf6Fs09YX.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pp. 647–655, 2014.

Engel, J., Agrawal, K., Chen, S., Gulrajani, I., Donahue, C., and Roberts, A. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

Fedus, W., Goodfellow, I., and Dai, A. MaskGAN: better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*, 2018.

Finlayson, S., Lee, H., Kohane, I., and Oakden-Rayner, L. Towards generative adversarial networks as a new paradigm for radiology education. *arXiv preprint arXiv:1812.01547*, 2018.

Frégier, Y. and Gouray, J. Mind2mind: transfer learning for GANs. *arXiv preprint arXiv:1906.11613*, 2019.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018.

Giacomello, E., Loiacono, D., and Mainardi, L. Transfer brain MRI tumor segmentation models across modalities with adversarial networks. *arXiv preprint arXiv:1910.02717*, 2019.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pp. 580–587, 2014.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *NIPS*, pp. 2672–2680, 2014.

Han, C., Murao, K., Noguchi, T., Kawata, Y., Uchiyama, F., Rundo, L., Nakayama, H., and Satoh, S. Learning more with less: conditional PGGAN-based data augmentation for brain metastases detection using highly-rough annotation on MR images. *arXiv preprint arXiv:1902.09856*, 2019.

Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., and Hayashi, H. Infinite brain MR images: PGGAN-based data augmentation for tumor detection. In *Neural Approaches to Dynamics of Signal Exchanges*, pp. 291–303. Springer, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*, pp. 6626–6637, 2017.

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pp. 1501–1510, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *CVPR*, June 2019a.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019b.

Krause, J., Stark, M., J., D., and Fei-Fei, L. 3D object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W., Sotelo, J., de Brebisson, A., Bengio, Y., and Courville, A. MelGAN: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*, 2019.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pp. 4681–4690, 2017.

Lin, K., Li, D., He, X., Zhang, Z., and Sun, M. Adversarial ranking for language generation. In *NIPS*, pp. 3155–3165, 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *ICCV*, pp. 3730–3738, 2015.

Long, M., Cao, Y., Wang, J., and Jordan, M. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are GANs created equal? a large-scale study. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *NeurIPS*, pp. 700–709. Curran Associates, Inc., 2018.

Luo, Z., Zou, Y., Hoffman, J., and Fei-Fei, L. Label efficient learning of transferable representations acrosss domains and tasks. In *NIPS*, pp. 165–177, 2017.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for GANs do actually converge? In *ICML*, pp. 3478–3487, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Mo, S., Cho, M., and Shin, J. Freeze discriminator: A simple baseline for fine-tuning GANs. *arXiv preprint arXiv:2002.10964*, 2020.

Mozafari, M., Farahbakhsh, R., and Crespi, N. A BERT-based transfer learning approach for hate speech detection in online social media. *arXiv preprint arXiv:1910.12574*, 2019.

Nilsback, M. and Zisserman, A. Automated flower classification over a large number of classes. In *ICCV, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Noguchi, A. and Harada, T. Image generation from small datasets via batch statistics adaptation. In *ICCV*, pp. 2750–2758, 2019.

Peng, Y., Yan, S., and Lu, Z. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*, 2019.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

Shin, H., Roth, H., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016a.

Shin, S., Hwang, K., and Sung, W. Generative knowledge transfer for neural language models. *arXiv preprint arXiv:1608.04077*, 2016b.

Wang, J. and Perez, L. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 2017.

Wang, K. and Wan, X. SentiGAN: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pp. 4446–4452, 2018.

Wang, R., Zhou, D., and He, Y. Open event extraction from online text using a generative adversarial network. *arXiv preprint arXiv:1908.09246*, 2019.

Wang, T., Liu, M., Zhu, J., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018a.

Wang, X., Shrivastava, A., and Gupta, A. A-fast-RCNN: Hard positive generation via adversary for object detection. In *CVPR*, pp. 2606–2615, 2017.

Wang, Y., Wu, C., Herranz, L., van de Weijer, J., Gonzalez-Garcia, A., and Raducanu, B. Transferring GANs: generating images from limited data. In *ECCV*, pp. 218–234, 2018b.

Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F., and van de Weijer, J. MineGAN: effective knowledge transfer from GANs to target domains with few images. In *CVPR*, 2020.

Yamamoto, R., Song, E., and Kim, J. Probability density distillation with generative adversarial networks for high-quality parallel waveform generation. *arXiv preprint arXiv:1904.04472*, 2019.

Yi, X., Walia, E., and Babyn, P. Generative adversarial network in medical imaging: A review. *Medical image analysis*, pp. 101552, 2019.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *NIPS*, pp. 3320–3328, 2014.

Zamir, A., Sax, A., Shen, W., Guibas, L., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *CVPR*, pp. 3712–3722, 2018.

Zeiler, M. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, pp. 818–833. Springer, 2014.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *NIPS*, pp. 487–495, 2014.