Transferring Unconditional to Conditional GANs with Hyper-Modulation

Héctor Laria[1] Yaxing Wang[2] Joost van de Weijer[1] Bogdan Raducanu[1]
[1] Computer Vision Center, Barcelona, Spain
[2] Nankai University, China
{hlaria, yaxing, joost, bogdan}@cvc.uab.es

Abstract

GANs have matured in recent years and are able to generate high-resolution, realistic images. However, the computational resources and the data required for the training of high-quality GANs are enormous, and the study of transfer learning of these models is therefore an urgent topic. Many of the available high-quality pretrained GANs are unconditional (like StyleGAN). For many applications, however, conditional GANs are preferable, because they provide more control over the generation process, despite often suffering more training difficulties. Therefore, in this paper, we focus on transferring from high-quality pretrained unconditional GANs to conditional GANs. This requires architectural adaptation of the pretrained GAN to perform the conditioning. To this end, we propose hyper-modulated generative networks that allow for shared and complementary supervision. To prevent the additional weights of the hypernetwork to overfit, with subsequent mode collapse on small target domains, we introduce a self-initialization procedure that does not require any real data to initialize the hypernetwork parameters. To further improve the sample efficiency of the transfer, we apply contrastive learning in the discriminator, which effectively works on very limited batch sizes. In extensive experiments, we validate the efficiency of the hypernetworks, self-initialization and contrastive loss for knowledge transfer on standard benchmarks. Our code is available at https://github.com/hecoding/Hyper-Modulation.

1. Introduction

Generative Adversarial Networks (GANs) have become ubiquitous in a vast array of applications due to their modelling and synthesis power. Current high-quality GANs consist of several millions of parameters [23]. In this magnitude range, the training of these models quickly become prohibitive in terms of computing resources and amount of training data required. Transfer learning for generative models explores how the knowledge of pretrained GANs can be transferred to new domains potentially with much fewer training samples.

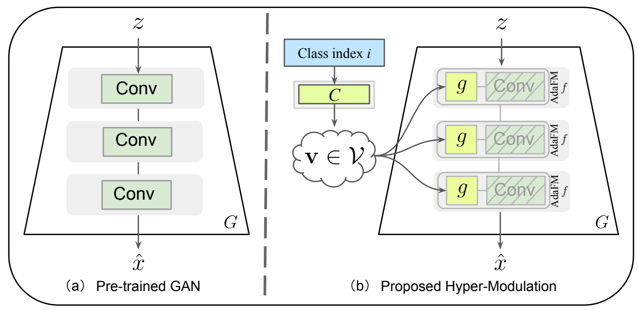

Figure 1. Transfer learning from unconditional (a) to conditional GAN (b), via on-the-fly modulation of the pre-trained weights. The class network C outputs a point $\mathbf{v}$ in the class space given a label i, which is then fed into the modulator g for domain-specific generation.

In the transfer learning area of generative models, Wang *et al*. [49] initially investigated unconditional transferring by finetuning a pre-trained GAN to a target domain. Further research improved the quality of transfer learning to small domains by reducing the number of learnable parameters [31, 34, 55] or by identifying the subspace of a pre-trained GAN that best models the target data [46]. The majority of efforts (see Table 1) have been driven towards transferring knowledge from unconditional GANs to also unconditional GANs (single source and target), from conditional to unconditional [46] (multiple sources, single target), which considers transferring a pre-trained cGAN to a single-class target domain, and from conditional to conditional [40] (multiple sources and targets), which proposes a method to transfer between conditional GANs through linear combination of conditionings.

In this work, we investigate the knowledge transfer from an unconditional to a conditional GAN. This setup is especially relevant, because of the availability of many high-quality unconditional pretrained GANs. There exist pretrained conditional models (cGANs), however, they have not seen adoption as widely as unconditional ones since they suffer from unstable performance among training runs [5], data and computational resources needed are higher, and do not employ an intermediate latent space, which is essential for GAN-based image editing [38, 42, 43, 58]. On the other hand, for many applications, it is required that the generation process should be conditional. Therefore, we investigate the transfer from unconditional pretrained GAN models to conditional GANs. An additional benefit of transferring to conditional GANs (when compared to transferring to multiple unconditional GANs) is the fact that they enable the sharing of weights between the multiple classes, thereby exploiting the similarities between the various classes.

In this paper, we leverage weight modulation from the context of continual learning [9, 39] to transform an unconditional source GAN to a cGAN, as depicted in Fig. 1. Our method allows for efficient transfer learning, where frozen pre-trained weights are conditionally modulated to yield target-specific outputs. However, a drawback of this approach is that the class-specific modulation parameters are learned independent of each other. To exploit the existing similarities among the multiple classes of the target domain, we propose the use of hypernetworks [16]. Hypernetworks have been proven efficient on diverse areas, from multi-task learning [30, 37, 41] to continual learning [45], delivering additional improvements on weight pruning [27] over traditional networks. Yet to our knowledge, they have not been applied to transfer learning. In this work, we aim to show that hypernetworks can result in more efficient knowledge transfer to multi-class domains, due to their intrinsic knowledge sharing among layers [16, 45]. However, the hypernetwork introduces new parameters that need to be trained from scratch to even regain the source generation power. To initialize these parameters, we propose a *self-alignment* method that learns well-initialized hypernetworks without getting access to any real data. Furthermore, we introduce contrastive learning in the discriminator for quality improvement like other generative methods propose [18, 19, 52], except that this effectively works with very limited batch sizes, *i.e*. 10 samples, contrary to current literature [7, 15, 19].

In summary, we propose the following contributions.

- We are the first to investigate knowledge transfer from unconditional to conditional GANs.
- We propose a new method based on hypernetworks and adaptive weight modulation that efficiently transfers unconditional to conditional GANs.
- In addition, we propose an approach for self-initialization of the hypernetwork parameters, that further allows applying a contrastive loss to the GAN discriminator with tiny batch sizes. Both these novelties result in significant improvements of the knowledge transfer.
- Results on several datasets show that we outperform existing methods and that FID improves on several datasets (including a notable drop of 30 points on the AFHQ dataset).

Method	Source	Target
TransferGAN [49], MineGAN [46] AdaFM [55],FreezeD [31], BSA [34] EWCGAN [26], CDCGAN [36]	U	U
MineGAN [46]	C	U
cGANTransfer [40]	C	C
Hyper-Modulation (Ours)	U	C

Table 1. Overview of existing transfer learning methods for GANs according to whether involved GANs for source and target domain are unconditional (U) or conditional (C). Even though transfer learning for GANs has seen an increased research activity, transferring unconditional to conditional has not been addressed before. The existence of high-quality unsupervised models [23] – that are the state of the art in high-resolution image generation – makes their transfer to conditional target domains especially pertinent.

2. Related work

Generative adversarial networks. GANs play a minimax game [13] between a generator and discriminator. The discriminator aims to tell the real distribution and the fake one apart, while the generator tries to synthesize a data distribution good enough to be mistaken by the real data distribution. However, optimizing GANs faces two challenges: mode collapsing and training instability. The former means that the generated data distribution concentrates on a small subset of outputs. The latter is due to the case that preserving a Nash equilibrium for both discriminator and generator is non-trivial. GAN variants [1, 14, 28] propose improved theory to address these problems. Another line of work [5, 10, 22] investigates devising efficient architectures to generate high-resolution images.

Transfer learning. This area aims to use the knowledge of the model (*i.e*., *source*) trained on a large domain to accelerate the training and reduce the amount of training data required by a model (i.e., *target*). Related works study knowledge transfer on generative models [26, 34, 46–50, 55] as well as discriminative models [12]. Regarding generative models, TransferGAN [49] is one of the first works that explores transfer learning, using finetuning on pre-trained GANs and denoting good performance on small dataset.

Hypernetworks. Hypernetworks are implicit generators [16, 44] that aim to generate parameters for other models. Hypernetworks have been applied to various tasks: architecture search [54], few-shot learning [2] and lifelong learning [45]. In this paper, we use Hypernetworks to generate the weights to modulate the learned weight of the pre-trained GAN. To our best knowledge, hypernetworks have not been used before to perform knowledge transfer. Furthermore, we use Hypernetworks to achieve the knowledge transfer from an unconditional GAN to a conditional GAN. Our method can be seen as a straightforward implementation [45] of a hypernetwork, producing the entire set of weights for a target neural network. However, we substitute the task embeddings $\{e^i\}_{i=0}^T$ for a semantically rich class space $\mathcal{V}$. Leveraging this space and the knowledge from the source domain of transfer learning, we are able to use more light-weight hypernetwork submodules, *i.e.*, mainly consisting of simple affine transformations.

Contrastive learning. In recent years, contrastive learning has been bridging the gap between supervised and unsupervised learning [7]. Data augmentation [11, 51] has very often been used in representation learning to keep the mutual information of different augmentations while disregarding nuisances not useful for generalization. We can see it explicitly mixed into the GAN training dynamics [53, 56] when applied to the discriminator or the GAN objective as a form of data efficiency regularization. Our application can be seen as a more simplified version of contraD [18], with a joint objective for real and fake samples, and SimCLR [7] is replaced with Barlow Twins [53] as the contrastive objective. To our knowledge, while some work has been carried out on augmentation [6] and semi-supervision [4], no other work has applied contrastive training to improve hypernetworks.

3. Methodology

We consider a source domain represented by the dataset $\mathcal{D}_s$ and a multi-class target domain $\mathcal{D}_t$. Given a pre-trained model on the source domain $f_0(\cdot)$, we aim to use transfer learning to efficiently learn a hypernetwork $f_h(\cdot)$ that can generate weights for all classes of the target domain.

To shape an unconditional GAN into a conditional one, we introduce class specific parameters in Section 3.1 that result in a certain modulation of the forward pass through the generator. This allows to drive the model toward the distributions of each target class. Next, to prevent learning of separate modulation parameters for all the classes, in Section 3.2 we propose the hypernetworks to directly estimate the modulation parameters – and importantly share the knowledge required to generate them among the classes. This is motivated by the fact that hypernetworks have been shown to efficiently transfer knowledge from one task to another one in the context of continual learning [45]. However, since the introduced hypernetwork needs to be trained from scratch, the system suffers from hard optimization and long training. Thus, in Section 3.3, we present a new self-distillation method to learn well-initialized weight for hypernetworks without the need of any data. Finally, in Section 3.4 we show that contrastive learning can be applied to further improve the efficiency of the knowledge transfer and improve the quality of the generation.

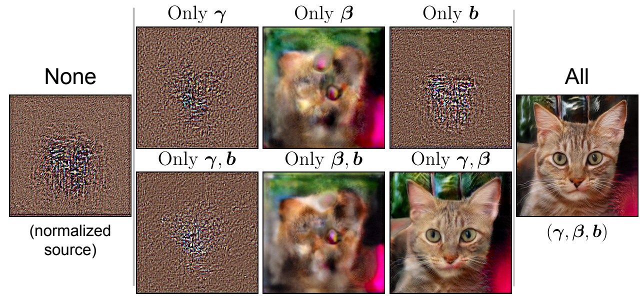

Figure 2. Effect of the modulation parameters on the domain transfer. Parameters γ generate high-frequency details, *i.e.*, texture and structure. β takes care of low-frequency details, *i.e.*, color. b is for localized details unattainable otherwise. A detailed figure is shown in the Appendix.

3.1. Domain transfer

Given a source generative model trained on $\mathcal{D}_s$, we aim to apply its knowledge to aid the learning of arbitrarily far domains. Concretely, given a pre-trained (*i.e.*, source domain) fully connected layer (or convolution, equivalently) $h^{\mathrm{s}}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ with pre-trained weights $\boldsymbol{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and input $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$. Inspired by [9, 35, 39], we can modulate its statistics to form a different layer as

$$\hat{\boldsymbol{W}}_i = \boldsymbol{\gamma}_i \odot \frac{\boldsymbol{W} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \boldsymbol{\beta}_i, \tag{1}$$

$$\hat{\boldsymbol{b}}_i = \boldsymbol{b} + \boldsymbol{b}_i, \tag{2}$$

where $\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ are learned parameters, $i = 1, ..., N_c$ indicates the class, N_c is the number of classes, and $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are the mean and standard deviation of $\boldsymbol{W}_i$. The rationale behind this modulation is that it first removes the source style encoded in $\boldsymbol{\mu}, \boldsymbol{\sigma}$ and then apply the learned one from $\boldsymbol{\gamma}, \boldsymbol{\beta}$ to model the statistics of a generative process of the target distribution. This normalization was originally proposed by [9] and called *Adaptive Filter Modulation* (AdaFM) in the context of continual learning of GANs.

In another vein, we apply this modulation concurrently to tackle the problem of transfer learning to multiple domains. The network weights $\boldsymbol{W}$ and $\boldsymbol{b}$ are shared among all the transferred classes, while the modulation parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{b}$ are the only ones changing. In Figure 2 we can see the effect of each parameter in the knowledge transference. In [9] they show that this modulation allows to model large domain shifts. Conditioning $\boldsymbol{\gamma}, \boldsymbol{\beta}$ and $\boldsymbol{b}$ we will be able to harness the modulated generation to produce conditional networks from an unconditional base.

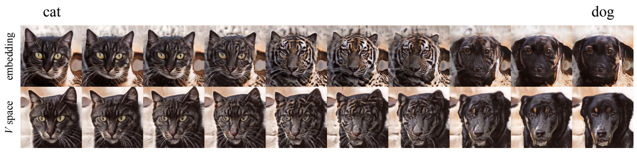

Figure 3. Conditioning interpolation on the modulation. Introducing a class projector on $\mathcal{V}$ results in smoother interpolation, although some features of other classes keep leaking while traversing. Additional interpolations can be found in the Supplementary Material.

3.2. Hyper-modulation

The method proposed in the previous section (Eq. 1) is optimized for each class in the target domain separately, and no parameters of the modulation are shared among the classes. As a result, we do not exploit similarities among classes in the target domains. To solve this, we propose the usage of hypernetworks [16], allowing us to share information and reduce memory usage by accumulating knowledge in the newly introduced modules.

Neural networks $f(\mathbf{x}, \Theta)$ are a family of functions that, given an input $\mathbf{x}$ and an output $\mathbf{y}$ coming from a dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$, typically learn a set of parameters Θ to find a function that maximizes the log likelihood of the data. Hypernetworks [16, 45] aim to learn the parameters Θ_h of a metamodel, which then will generate the target parameters Θ_{trg} of the target model f_{trg}.

In this work, we apply a hypernetwork g to predict the modulation parameters conditionally, which eventually enables us to produce a generative model for each target. The input of the hypernetwork is a vector coming from a class embedding network $C(i; \Psi) = \mathbf{v} \in \mathcal{V}$ where $(i = 1, ..., N_c)$ is the class label, $\mathcal{V}$ is the class embedding space, and Ψ are network parameters. Figure 3 shows qualitative improvement over learnable embeddings and Supplementary Material includes metrics and more extensive visualizations. By varying the number of parameters Ψ, we are able to vary the class knowledge capacity of the system. The hypernetwork g takes the embedding vector v and maps it to the modulation parameters according to:

$$\boldsymbol{\gamma}_{\mathbf{v}}, \boldsymbol{\beta}_{\mathbf{v}} = g(\mathbf{v}; \Phi_a), \qquad \boldsymbol{b}_{\mathbf{v}} = g_b(\mathbf{v}; \Phi_b) \tag{3}$$

where g are affine projections of a point in the space $\mathcal{V}$, with network parameters Φ_a and Φ_b. We use $\boldsymbol{\Phi}$ to denote the combination of all the parameters used by the hypernetwork, consisting of Φ_a and Φ_b for all the layers in the network. Each modulated layer has a g projector, but layer-wise, these are shared among target classes.

The modulation that produces target-specific activations

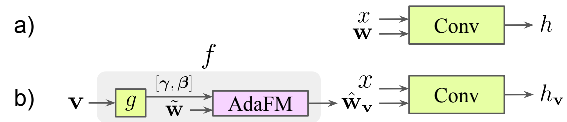

Figure 4. a) Activations h from a generator convolution in the source domain. b) Domain-specific activations from a hypermodulator f, conditioned on a point $\mathbf{v}$ in the class space $\mathcal{V}$.

$h_{\mathbf{v}}(\boldsymbol{x}) = \hat{\boldsymbol{W}}_{\mathbf{v}}\boldsymbol{x} + \hat{\boldsymbol{b}}_{\mathbf{v}}$ is of the form

$$\hat{\boldsymbol{W}}_{\mathbf{v}} = \boldsymbol{\gamma}_{\mathbf{v}} \odot \frac{\boldsymbol{W} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \boldsymbol{\beta}_{\mathbf{v}}, \tag{4}$$

$$\hat{\boldsymbol{b}}_{\mathbf{v}} = \boldsymbol{b} + \boldsymbol{b}_{\mathbf{v}}, \tag{5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the frozen source weights. Ultimately, a hypermodulator f will be given a class embedding $\mathbf{v}$ and a normalized source weight $\tilde{\mathbf{w}}$ to produce the desired target weights as $f_{\tilde{\mathbf{w}}}(\mathbf{v}) = \boldsymbol{\gamma}_{\mathbf{v}} \odot \tilde{\mathbf{w}} + \boldsymbol{\beta}_{\mathbf{v}} = \hat{\mathbf{w}}_{\mathbf{v}}$, following Eqs. (3) and (4) and pictured in Fig. 4.

Traditionally, reusability can be introduced in hypernetworks to reduce the number of trainable parameters. This is achieved by reapplying the metamodel for different partitions of the target model parameters, also called chunking [45]. We do not use chunking since each generator can be reduced to a minimum of a learned affine transformation thanks to transfer learning and the enhanced domain space $\mathcal{V}$, constituting a rather shallow but performing hypernetwork.

3.3. Self-alignment

The introduction of the new modules causes the augmented source model to initially lose its learned synthesis performance (see also Fig. 7a), mainly because the parameters Ψ, Φ have not been learned yet, as well as due to the removal of domain-specific statistics prior to the introduction of new ones, as seen in Eq. (4). This procedure is not necessarily bad, since new classes will only learn to produce their respective target statistics and not to compensate for the source ones. However, general training times will be affected since the network has to re-learn multi-scale feature statistics that produce real-world pixel distributions.

Therefore, we propose to self-align the parameters Ψ, Φ. The alignment is performed between the pre-trained generator network without hypernetwork and the one with hypernetwork (see Fig. 5). The aim is to not simply recover the original weight statistics, but also to initialize a sensible latent space for the embedding vectors v that could be further augmented by new classes.

We will perform this initialization as a first step before the final finetuning on the target data takes place. The hierarchical features extracted from the pre-trained model are given by $F_{\text{PT}}(\mathbf{z}) = \{G_{\text{PT}}(\mathbf{z})_l\}$ and the ones with hypernetwork by $F_{\text{hyp}}(\mathbf{z}) = \{G'_{\text{PT}}(\mathbf{z}, g(C(c^0; \Psi); \Phi))_l\}$ where

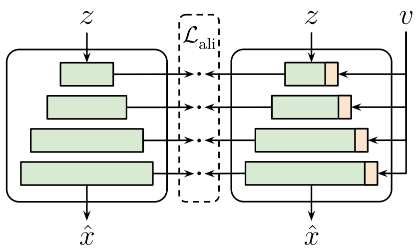

Figure 5. Self-alignment of the pre-trained generator (left) and the one with hypernetwork (right). Both networks are initialized with the same pre-trained weights (green) that are frozen. The new hypernetwork weights (yellow) are learned during the self-alignment. This operation does not require any data, since it can be performed by simply sampling a latent vector z.

$G(\cdot)_l$ is the l-th convolution block output. During the self-initialization, we set the class input to the class-embedding network C as $c^0 = 1$. The loss for this stage is:

$$\mathcal{L}_{\text{ali}} = \sum_l \|F_{\text{PT}}(\mathbf{z}) - F_{\text{hyp}}(\mathbf{z})\|_1. \tag{6}$$

Note that this operation does not require any real data, since we can align the two networks by simply sampling random vectors $\mathbf{z}$. After self-initialization, the network with the hypernetwork generates high-quality images (compare Fig. 7b and Fig. 7c).

In conclusion, the self-alignment initializes the hypernetwork parameters Ψ, Φ. When we now finetune the network on the multi-class target domain, we do not have to learn these parameters from scratch. In the experimental section, we verify that this significantly reduces the training time and improves the quality of the generated results.

3.4. Contrastive learning

We further extend this work to achieve better sample efficiency by applying contrastive learning on the discriminator used during adversarial training. Recent works on self-supervised learning have shown that by mapping different views (generated by taking different data augmentations of the same image) to the same point in latent space, strong semantically-rich feature representations can be learned that rival their supervised counterparts. Here, the idea is to exploit this fact to improve the quality of the discriminator used in adversarial training. The underlying insight is that if the discriminator can extract higher quality features, it can also better distinguish fake from real images, and as a consequence better challenge the generator, leading to higher quality images.

Concretely, we make use of Barlow Twins [53] for its simplicity and performance and apply it implicitly on the discriminator as in Fig. 6. We reuse all transformations for real and fake samples, but we employ no projector network because it resulted in worse quality. The loss function is

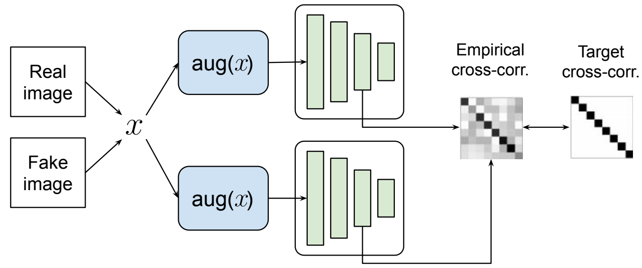

Figure 6. General scheme for contrastive learning on the discriminator. Cross-correlation (*cross-corr.*) is computed between the extracted features of two views of a real or fake image. Gradients don't flow back to the generator.

also left unchanged:

$$\mathcal{L}_{\text{contr}} = \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2 \tag{7}$$

with the scaling factor λ and the cross-correlation matrix $\mathcal{C}$ computed between the intermediate representations before the final layer.

We employ GAN [13] to optimize this problem:

$$\begin{aligned} \mathcal{L}_{gan} &= \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \mathbf{c} \sim p(\mathbf{c})} \left[\log D\left(\mathbf{x}, \mathbf{c}\right)\right] \\ &+ \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \mathbf{c} \sim p(\mathbf{c})} \left[\log(1 - D\left(G\left(\mathbf{z}, \mathbf{c}\right), \mathbf{c}\right)\right], \end{aligned} \tag{8}$$

where $p\left(\mathbf{z}\right)$ follows the normal distribution, and $\mathbf{p}\left(\mathbf{c}\right)$ is the domain label distribution.

The final training objective is

$$\mathcal{L}_{\text{GAN}} = \mathcal{L}_{gan} + \lambda_{\text{contr}} \mathcal{L}_{\text{contr}} \tag{9}$$

where λ_{contr} is a balancing hyperparameter set to $\lambda_{\text{contr}} = 1e-3$ in all our experiments. In the experimental section, we verify that contrastive learning can significantly improve the quality of the generated images.

4. Experiments

4.1. Settings

Training details. Our method is applied to a pre-trained StyleGAN [22]. Concretely, both the generator and discriminator are direct copies of the architecture, except for the top layer of the discriminator, for which the last fully connected layer has been replaced by a convolutional layer with 3×3 filter size, stride of 1 and output channel dimensionality of N_c number (classes in target domain). The hypernetwork class network $C(\cdot)$ consists of an embedding layer for all domains, followed by four fully connected layers. The dimensionality of the whole branch is 64. The hypernetwork modulators are implemented by a single fully connected layer that maps the class branch output to a dimensionality of 512. Hyperparameters from the original model are kept,

Configuration	mFID ↓	mKID ↓	P ↑	R ↑	D ↑	C ↑
No hypernet.	61.84	3.75	8.89	22.77	2.37	1.33
A Hyper-Mod	50.67	3.00	12.46	31.60	3.58	3.03
B + $\mathcal{V}$ space	45.28	2.28	12.12	40.18	3.67	3.31
C + Contrastive D	26.74	0.92	28.02	55.19	10.13	11.95

Table 2. Ablation on hypernetwork and the contrastive learning on AFHQ.

including Adam [24] and R1 regularization [29], while the model is trained at 256×256 resolution.

Evaluation metrics. We report results on two types of metrics: single-valued and double-valued metrics. The former contains Fréchet Inception Distance (FID) [17] and Kernel Inception Distance (KID) [3]. The latter consists of Precision and Recall (PR) [25] and Density and Coverage (DC) [32]. Both PR and DC evaluate the quality and the diversity. We use all training samples available to compute the metrics as suggested in [17, 21], since most datasets do not have as much as 10,000 class samples per class to have a good metric estimation. KID and DC are multiplied by 100 for easier visualization, and PR is given as percentage. FID is calculated per class and the average is taken (mFID).

Datasets. Our experiments are conducted on Animal Faces dataset (AFHQ) [8], FFHQ [22], CelebA-HQ [20], Flowers102 [33] and Places 365 [57]. AFHQ contains 3 classes, each one has about 5000 images. In CelebA-HQ, we use gender as a class, with 10k male and 18k female images in the training set. Flowers102 consists of 102 categories, but since the number of samples per class is small, we ignore the labels to form an unconditional dataset. In Places 365 [57] dataset, we select only 10 categories as target: *amphitheater*, *aqueduct*, *castle*, *dam*, *field road*, *fire station*, *pagoda*, *underwater - ocean deep*, *volcano* and *waterfall*. In this paper, all images are resized to 256×256.

Baselines. Since no previous work has explored transfer learning from unconditional to conditional GANs, there exist only few works to properly compare against. We use the following baselines: *GAN Memory* [9] (unconditional to unconditional) proposed a weight modulation method to address catastrophic forgetting of GAN for lifelong learning, *cGANTransfer* [40] (conditional to conditional) introduced a conditional batch normalization method to perform knowledge transfer, which aims to learn the class-specific information of the new classes from that of the old classes. We explore a variant of our method, named as *Hyper-Mod-FT*, for which all parameters are updated.

4.2. Ablation study

Hypernetwork. Comparing a modulation like *GAN Memory* (No hypernet.) to the proposed hypernetwork (config. A) in Table 2, we can appreciate better synthesis quality

Method	full	end
A Hyper-Mod	61.94	61.78
B + $\mathcal{V}$ space	**61.23**	**59.85**

Table 3. Perceptual path length among classes, for both full paths and endpoints. All scores are in the magnitude of 10^6.

and especially a diversity increase for the latter, more than doubling for both Recall and Coverage. We attribute that to the knowledge sharing and complementary supervision in the joint training, since each input is affecting and shaping the whole hypernetwork as opposed to learning separate embedding points for modulation.

Self-initialization effect. Training with an uninitialized hypernetwork (Fig. 7a) is compared to a self-aligned one (Fig. 7b) towards a source model (Fig. 7c). Figure 7d shows huge improvements in training time as well as a significant improvement in quality. We argue that learning proper hierarchical modulation correlation plays a crucial role for consecutive direct application of training information to each target, compared to learning both concurrently from scratch.

Target space. A commonly desired characteristic of latent spaces is the linearity of its factors of variation (*e.g.* pose, color, etc.). Our goal with the class network C introduced in Section 3.2 is to unwarp subspaces that the learned class embedding could have had difficulties dealing with for several reasons, *i.e.* scarcity of specific training samples or complexity of the modelling. To quantify the beneficial effect of the introduced module, we employ a disentanglement metric called Perceptual path length [22], consisting on measuring how drastic perceptual changes in the image occur while performing interpolation. Intuitively, a linear latent space presents smoother transitions than a warped one. Results shown in Table 3 confirm us the advantage of introducing this network against class embeddings. Magnitudes are naturally bigger than style measurements since changes in class are non-trivial perceptual alterations compared to, *e.g.*, color changes. Generation metrics also denote improvement in quality and diversity in Table 2 (config. B). Finally, we show in Supplementary Material Figure 10 that class regarding style has appropriate independence, *i.e.*, changes in class only affect shape, but fur color, background, etc. are left unchanged. Style regarding class cannot be dependent since the style mechanism is frozen at the beginning of the transfer.

Contrastive learning. We found self-supervision beneficial to transfer learning. We tried some contrastive losses (see Table 5) and choose the best one. This provides improvements even with a small batch size of 10 samples, for which we compute an FID of 37.15 and KID of 1.66, already improving config. B in Table 2. Results reported in config. C are computed for a batch size of just 60 due to

(a) Uninitialized. (b) Self-aligned. (c) Source. (d) Generation quality over time.

Figure 7. Self-initialization details. Generator outputs of a hypermodulator (a, b). Source pre-trained generator (c) for comparison. Training efficiency of self-alignment (d).

Method \ Dataset	Close domain		Far domain	
	FFHQ→AFHQ	AFHQ→CelebA	FFHQ→Flower102	FFHQ→Places365
Hyper-Mod-S	498.41	498.41	498.41	498.41
GAN Memory	61.84	49.30	144.93	229.49
cGANTransfer	112.64	105.95	-	-
Hyper-Mod-FT	30.11	24.54	40.07	98.24
Hyper-Mod	45.28	45.54	127.78	132.42

Table 4. Comparison with baselines on mean FID. A→B: From source A to target B. S: From scratch. FT: finetune source weights.

Configuration	mFID ↓	mKID ↓	P ↑	R ↑	D ↑	C ↑
GAN Memory [9]	61.84	3.75	8.89	22.77	2.37	1.33
cGANTransfer [40]	112.64	9.90	2.93	18.95	0.73	2.10
Hyper-Mod	45.28	2.28	12.12	40.18	3.67	3.31
Hyper-Mod-FT	30.11	1.09	16.99	62.68	5.76	6.49
Hyper-Mod + DCL [52] (bs 60)	42.28	2.00	19.82	42.05	7.31	5.67
Hyper-Mod + BT [53] (bs 60)	26.74	0.92	28.02	55.19	10.13	11.95

Table 5. Comparison with baselines on several metrics on AFHQ. P: Precision, R: Recall, D: Density and C: Coverage.

computational constraints. These are expected to further improve with larger batch sizes, as in [53]. Unfortunately, contrastive learning in the generator did not result in improved quality.

Domain information injection. Is weight modulation the best method to incorporate target information during the transfer learning? We could think about style transfer techniques such as Adaptive Instance Normalization (AdaIN) [22], which modulates at the activation level. In Appendix E we provide specifications on the implementation of this method in place of modulation. This modification can be compared to config. B and yields an FID and KID of 110.55 and 9.26 respectively (compared to our proposed architecture with 45.28 and 2.28). Thus, we conclude that weight modulation is favorable over other style transfer methods.

4.3. Result

Quantitative results. To evaluate the performance of the proposed method, we test our method on both *close domain transfer* and *far domain transfer*. The former means both source and target domain have small domain shift, and the latter is they have a large domain gap. These two settings are used to validate the effectiveness of the proposed method on different target domains.

Close domain transfer. Here, we use both the AFHQ animal dataset and CelebA human face as our target domains. For the former, the pretrained StyleGAN is optimized on FFHQ human face. We use the pretrained StyleGAN optimized on AFHQ animal face when the target domain is CelebA human face. As reported in Table 4 (*close domain* column), training the network from scratch obtains catastrophic results (e.g., 498.41 FID). Using the transfer learning method (like GAN Memory) largely improves the performance (e.g., 61.84 FID for GAN Memory). The proposed method achieves better performance (denoted as *Hyper-Mod* in the table), we generate more realistic and correct class-specific images among the compared methods. In addition, we also conduct an experiment with updating all parameters (denoted as *Hyper-Mod-FT*). *Hyper-Mod-FT* further improves the performance.

We also evaluate our method and the baselines on several other metrics. As reported in Table 5, we achieve the best score on all metrics, which indicates that we not only generate high-quality images (corresponding to P. and D.), but also diverse images (corresponding to R. and C.).

Far domain transfer. We also consider the challenging setting by using a target dataset which has a large domain gap with the source domain. Here we consider two target domains: Flower102 and Places365. For the two target datasets, we use the same source pre-trained StyleGAN, which is optimized on FFHQ. As reported in Table 4 (*far domain* column), in the far domain setting our method still obtains a large advantage when compared to the baselines (e.g., 127.78 FID (ours) vs 144.93 FID (GAN Memory) on Flower102). What is more interesting is that we are able to greatly improve the performance when further updating all parameters. Finally, like for the close domain transfer, the proposed techniques (*i.e.*, hypernetwork, self-alignment and contrastive learning) are effective when per-

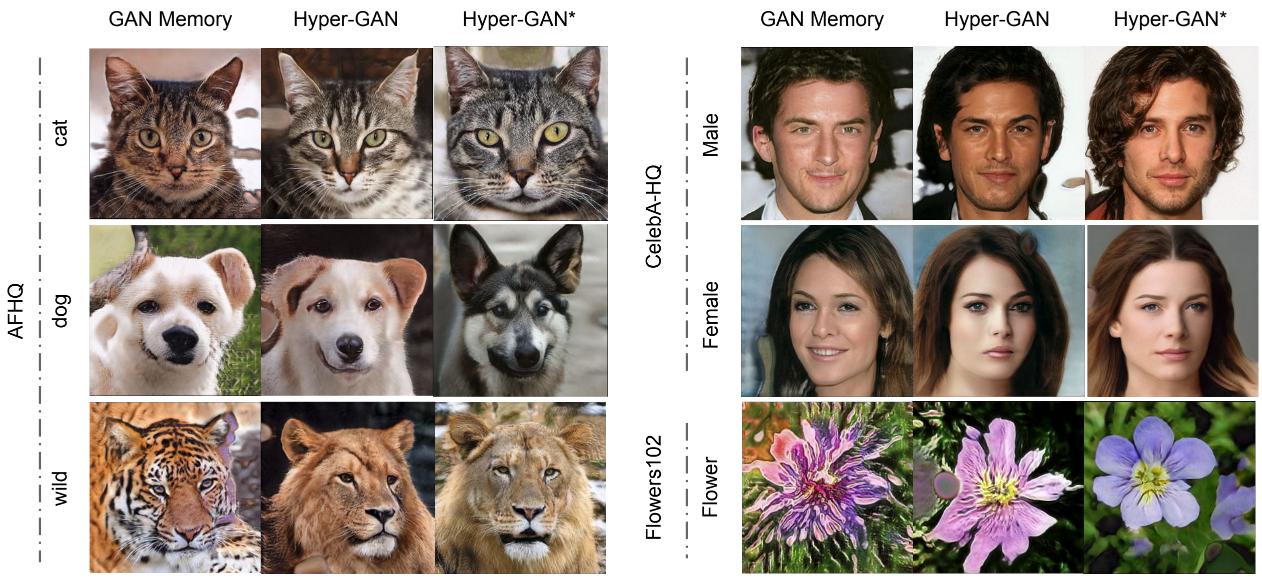

Figure 8. Qualitative comparison on AFHQ, CelebA-HQ and Flowers102 datasets. More examples are shown in the Suppl. Mat. Section.

Figure 9. Qualitative results of the proposed method on AFHQ. Each row is corresponding to one target class. For each target, we show five different breeds appearing in the generation.

forming knowledge transfer from unconditional GAN to conditional GAN on far domain transfer.

Qualitative results. Regarding *close domain transfer*, Figure 8 shows the comparison to baselines on AFHQ, CelebA and Flowers102 datasets. Although GAN Memory is able to conduct multi-class generation, it fails to generate highly realistic images (first column of Figure 8 on AFHQ). Taking AFHQ as an example, given the target class label, the proposed method is able to provide high-quality images (e.g., the second column of Figure 8). when updating all parameters (Hyper-Mod-FT), we further improve the qualitative result (e.g., the third column of Figure 8). Moreover, we demonstrate that our method has both scalability and diversity in a single model. Each row of Figure 9 shows the different results when changing the target class label. Our method manages to cover different breeds in the same class, while keeping source style controls (*i.e.*, colors, pose, background, etc.) unaltered (Fig. 3).

For *far domain transfer*, qualitative results on *Places365* dataset to complement quantitative ones can be seen in Supplementary Material Figure 11, together with additional unfiltered generations and interpolations.

5. Conclusions

We investigated the knowledge transfer from GAN to cGAN. To tackle it, we proposed hyper-modulation to produce weight modulation parameters on-the-fly for a source model. Training the hypernetwork from scratch complicates training, thus we proposed a self-initialization method that does not require any data to learn well-initialized weights. To enhance the capacity of the discriminator, we introduced self-supervision for it. Our qualitative and quantitative results showed the proposed method outperforms existing state-of-the-art results on transfer learning.

Limitations One further line of work is memory, to not keep the whole pre-trained network in memory. Second, the state of the art in unconditional generation uses a similar modulation [23] to incorporate the style. We are positive that combining and leveraging both methods is possible.

Acknowledgements

We acknowledge the support from Huawei Kirin Solution, Spain Government funded project PID2019-104174GB-I00/AEI/10.13039/501100011033 and the EU Project CybSpeed MSCA-RISE-2017-777720.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *ICLR*, 2017. 2

[2] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *NeurIPS*, 2016. 2

[3] M Bińkowski, DJ Sutherland, M Arbel, and A Gretton. Demystifying mmd gans. In *ICLR*, 2018. 6

[4] Dhanajit Brahma, Vinay Kumar Verma, and Piyush Rai. Hypernetworks for continual semi-supervised learning. *arXiv preprint arXiv:2110.01856*, 2021. 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[6] Chih-Yang Chen and Che-Han Chang. Hypernetwork-based augmentation. *arXiv preprint arXiv:2006.06320*, 2020. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 6

[9] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. GAN memory with no forgetting. In *NeurIPS*, 2020. 2, 3, 6, 7

[10] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, pages 1486–1494, 2015. 2

[11] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 3

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2, 5

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5767–5777, 2017. 2

[15] Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(11):307–361, 2012. 2

[16] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *ICLR*, 2017. 2, 4

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, pages 6626–6637, 2017. 6

[18] Jongheon Jeong and Jinwoo Shin. Training gans with stronger augmentations via contrastive discriminator. In *ICLR*, 2021. 2, 3

[19] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21357–21369. Curran Associates, Inc., 2020. 2

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 6

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 6

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 2, 5, 6, 7

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 1, 2, 8

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 6

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019. 6

[26] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In *NeurIPS*, 2020. 2

[27] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting (Tim) Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *ICCV*, pages 3296–3305, 2019. 2

[28] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017. 2

[29] Lars M. Mescheder. On the convergence properties of GAN training. *CoRR*, abs/1801.04406, 2018. 6

[30] Elliot Meyerson and Risto Miikkulainen. Modular universal reparameterization: Deep multi-task learning across diverse domains. In *NeurIPS*, 2019. 2

[31] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. In *CVPR AI for Content Creation Workshop*, 2020. 1, 2

[32] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *ICML*, pages 7176–7185, 2020. 6

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008. 6

[34] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *ICCV*, pages 2750–2758, 2019. 1, 2

[35] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, pages 2642–2651, 2017. 3

[36] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *CVPR*, pages 10743–10752, 2021. 2

[37] Zheyi Pan, Yuxuan Liang, Junbo Zhang, Xiuwen Yi, Yong Yu, and Yu Zheng. Hyperst-net: Hypernetworks for spatio-temporal forecasting. *CoRR*, abs/1809.10889, 2018. 2

[38] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NeurIPS*, 2016. 2

[39] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, pages 3942–3951, 2018. 2, 3

[40] Mohamad Shahbazi, Zhiwu Huang, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Efficient conditional gan transfer with knowledge propagation across classes. In *CVPR*, pages 12167–12176, 2021. 1, 2, 6, 7

[41] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *CVPR*, pages 8061–8069, 2018. 2

[42] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 2

[43] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, pages 5541–5550, 2017. 2

[44] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *CVPR*, pages 10753–10764, 2021. 2

[45] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In *ICLR*, 2020. 2, 3, 4

[46] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *CVPR*, pages 9332–9341, 2020. 1, 2

[47] Yaxing Wang, Abel Gonzalez-Garcia, Chenshen Wu, Luis Herranz, Fahad Shahbaz Khan, Shangling Jui, and Joost van de Weijer. Minegan++: Mining generative models for efficient knowledge transfer to limited data domains. *arXiv preprint arXiv:2104.13742*, 2021. 2

[48] Yaxing Wang, Héctor Laria, Joost van de Weijer, Laura Lopez-Fuentes, and Bogdan Raducanu. Transferi2i: Transfer learning for image-to-image translation from small datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14010–14019, 2021. 2

[49] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *ECCV*, pages 218–234, 2018. 1, 2

[50] Yaxing Wang, Lu Yu, and Joost van de Weijer. Deepi2i: Enabling deep hierarchical image-to-image translation by transferring from gans. *NeurIPS*, 2020. 2

[51] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *CVPR*, 2018. 3

[52] Ning Yu, Guilin Liu, Aysegul Dundar, Andrew Tao, Bryan Catanzaro, Larry S Davis, and Mario Fritz. Dual contrastive loss and attention for gans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6731–6742, 2021. 2, 7

[53] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 3, 5, 7

[54] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. In *ICLR*, 2019. 2

[55] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for limited-data generation. *ICML*, 2020. 1, 2

[56] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. In *NeurIPS*, 2020. 3

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2017. 6

[58] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *ECCV*, 2020. 2