Hyper-GAN: Transferring Unconditional to Conditional GANs with HyperNetworks

Héctor Laria  Yaxing Wang  Joost van de Weijer  Bogdan Raducanu
Computer Vision Center
Barcelona, Spain
{hlaria, yaxing, joost, bogdan}@cvc.uab.es

Abstract

Conditional GANs have matured in recent years and are able to generate high-quality realistic images. However, the computational resources and the training 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. In this paper, we explore the transfer from high-quality pre-trained unconditional GANs to conditional GANs. To this end, we propose hypernetwork-based adaptive weight modulation. In addition, 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 knowledge transfer, we propose to use a self-supervised (contrastive) loss to improve the GAN discriminator. In extensive experiments, we validate the efficiency of the hypernetworks, self-initialization and contrastive loss for knowledge transfer on several standard benchmarks.

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 millions of parameters, in the order of 60M for unconditioned [19] and 110M for conditioned [5] models. With these numbers, the training of these models quickly become prohibitive in terms of computing resources and amount of training data. In discriminative networks, there exist well-known methods to reuse previous learned knowledge such as feature extractors [2] and by means of finetuning [36].

The study of transfer learning for generative models has started more recently. Initially, Wang et al. [45] investigate the efficiency of finetuning a pre-trained GAN to a target domain. Later research improved the quality of transferring to small domains by reducing the number of learnable parameters [30, 33, 49] or by identifying the subspace of the pre-trained GAN that best models the target data [44]. The majority of works focus on transferring from unconditional GANs to also unconditional GANs (see also Table 1). Only some recent works consider other scenarios: MineGAN [44] considers transferring from a pre-trained conditional GAN (cGAN) to a single-class target domain, and cGANTransfer [40] proposes a method to transfer between conditional GANs. To the best of our knowledge, transferring knowledge from an unconditional source model to a conditional target model has not yet been explored.

GANs have seen astonishing improvements since their inception [12]. Initially significant enhancements were obtained by architectural improvements [5, 9, 18] as well as theoretical improvements [1, 13, 27]. One of the major milestones has been the BigGAN [5] that is able to conditionally generate realistic images learned from many-class datasets. Another line of research pursued the generation of high resolution images [17, 18]. Among the state-of-the-art in this direction are the StyleGANs [18] which are unconditional GANs that explicitly disentagle style and content in the
architectural design. Training these networks requires significant amount of computational resources. For example, it needs about 41 days to train StyleGAN with one Tesla V100 GPUs on the FFHQ dataset at 1024×1024 resolution. Transferring the knowledge of these high-quality generators to new target domains is therefore highly desirable. In this paper, we focus on how these high-quality generators can be transferred to conditional GANs.

In this paper, we use Adaptive Filter Modulation [8] to transform an unconditional GAN to a conditional GAN. The advantage of this method is that the weights of the pre-trained unconditional GAN can be used directly. The conditioned output is realized by modulating these pre-trained weights that remain frozen during the transfer with class-specific modulation parameters. However, a drawback of this approach is that the class-specific modulation parameters are learned independent of each other. To exploit the similarities, that exist between the transfers from the source to the multiple classes of the target domain, we propose to use hypernetworks [14] (see also Fig. 1). Hypernetworks have been proven efficient on diverse areas, from multi-task learning [29, 37, 41] to continual learning [43], and delivering additional improvements on weight pruning [26] over traditional networks. However, to our knowledge, they have not yet been applied for transfer learning. We here aim to show that hypernetworks can result in more efficient knowledge transfer to multi-class domains, because they allow to share the parameters between the target classes. However, the hypernetwork introduces new parameters that need to be trained from scratch and when trained on smaller target domains can lead to overfitting. To accommodate this problems, we propose self-initialization method to learn well-initialized hypernetwork without getting accessing to any real data. Furthermore, we introduce a new contrastive learning method to facilitate the training of the discriminator.

In summary our paper has 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 GANs to conditional GANs.
- In addition, we propose a new approach for self-initialization of the hypernetwork parameters, and we propose to enhance the GAN discriminator with a contrastive loss. Both these novelties result in significant improvements of the knowledge transfer.
- Extensive results on several dataset show that we outperform existing methods and that FID improves significantly on several datasets (including a notable drop of 30 points on the AFHQ dataset).

| Method                        | Source | Target |
|-------------------------------|--------|--------|
| TransferGAN [45], MineGAN [44] | U      | U      |
| AdaFM [49], FreezeD [30], BSA [33] | U      | U      |
| EWCGAN [24], CDCGAN [35]     | U      | U      |
| MineGAN [44]                 | C      | U      |
| cGANTransfer [30]            | C      | C      |
| Hyper-GAN (Ours)             | U      | C      |

Table 1. Overview 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 (like StyleGAN [19]) – 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 [12] between a generator and discriminator. The discriminator aims to recognize the real distribution and the fake one, while the generator tries to synthesize a data distribution which is expected to match the real data distribution. However, optimizing GANs faces two challenges: the mode collapse and the training instability. The former means that the generated data distribution is 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. The variants of GANs [1, 13, 27] propose improved theory to address these problems. Another line of work [5, 9, 18] investigates devising efficient architectures to generate high-resolution images.

**Transfer learning.** Transfer learning aims to use the knowledge of the model (i.e., source) trained on a large domain to facilitate the performance, accelerate training and reduce the amount of training data required by a model (i.e., target). Related works explored the knowledge transfer on generative models [24, 33, 44, 45, 49] as well as discriminative models [11]. TransferGAN [45] initially explore fine-tuning of pre-trained GANs, and clearly showing the efficiency with small dataset. The direct finetuning, however, lead to the mode collapse and overfitting, since all parameters are optimized with limited data. Batch statistics adaptation (BSA) [33] instead solely update the batch normalization parameters to overcome the overfitting. However, it fails to learn the target distribution, and only model the relationship between latent vectors and sparse training samples, since it uses a mean square error loss instead of the GAN loss. To overcome the finetuning, related work [30, 49] also optimize a partial set of network parameters by fixing some layers of the pre-trained GANs. However, they require to select manually the specific layer which is frozen.
or not. The recent method [24] also explore combining pre-trained GAN training with Elastic Weight Consolidation (EWC) [22]. CDCGAN [35] perform transfer learning by discovering cross-domain correspondences. An additional difference between our work and existing work on transfer learning for GANs is that we are the first to consider transferring knowledge from unconditional GANs to conditional GANs (see Table 1).

**Hypernetworks.** Hypernetworks are implicit generators [14, 42] which aim to generate parameters for other models. Hypernetworks have been applied to various tasks: architecture search [48], few-shot learning [3] and lifelong learning [43]. In this paper, we use Hypernetworks to generate the weights to modulate the learned weight of the pre-trained GAN. To our best knowledge, we are the first one to use Hypernetworks 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 [43] of a hybernetwork, producing the entire set of weights for a target neural network. We however, substitute the task embeddings \(\{e^i\}_{i=0}^T\) for a semantically rich class space \(V\). Leveraging this space and the knowledge from the source domain of transfer learning, we are able to use more light-weight hybernetwork 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 [6]. Data augmentation [10, 46] 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 [47, 50] 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 [16], with a joint objective for real and fake samples, and SimCLR [6] is replaced with Barlow Twins [47] as the contrastive objective. To the best of our knowledge, we are the first to show that contrastive training in the GANs can also be efficiently applied to improve transfer learning of generative models.

### 3. Methodology

As discussed above, the problem of transferring knowledge from high-quality pre-trained unconditional GANs to conditional generative models has received little attention. We argue this is important because these GANs (like e.g. StyleGAN) represent the state-of-the-art in generative models of high-resolution images and thus transferring their knowledge to smaller multi-class target domains is often desirable.

To change the unconditional GAN into a conditional one, we introduce class specific parameters that result in a class-specific modulation of the forward pass through the generator. This allows it to generate similar distributions to each target class. To prevent learning of separate modulation parameters for all of the classes, we propose to use hyer-networks to efficiently compute these class-specific weights – and importantly share the knowledge required to generate them among the classes. This is motivated by the fact that hyer-networks have been shown to efficiently transfer knowledge from one task to another one in the context of continual learning [43].

In Section 3.1 we specify the modulation mechanism used to change a pre-trained unconditional GAN into a conditional GAN. Next, in Section 3.2 we propose the hyer-networks to directly estimate the modulation parameters. However, since the introduced hyer-network 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 hyer-networks without the need of any data. Finally, in Section 3.4 we show that self-supervised learning (i.e. contrastive learning) can be applied to further improve the efficiency of the knowledge transfer and improve the quality of the generation.

We consider a source domain represented by the dataset \(D_s\) and a multi-class target domain \(D_t\). Given a pre-trained model on the source domain \(f_0(\cdot)\), we aim to use transfer learning to efficiently learn a hyer-network \(f_h(\cdot)\) that can generate weights for all classes of the target domain. Transfer learning driven by hyer-networks can be generically applied to any unconditional or conditional generative network. In this work we use StyleGANv1 as a compromise between easy conceptualization (and implementation) and generation quality.

#### 3.1. Weight generators via modulation

Here we aim to reuse the knowledge of a source generative model trained on \(D_s\). Concretely, given a pre-trained (i.e., source domain) fully connected layer (or convolution, equivalently) \(h^i(x) = Wx + b\) with pre-trained weights \(W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}\) and input \(x \in \mathbb{R}^{d_{\text{in}}}\). Inspired by [8, 34, 39], we can modulate its statistics to form a different layer as

\[
\hat{W}_i = \gamma_i \odot \frac{W - \mu}{\sigma} + \beta_i, \quad (1)
\]
\[
\hat{b}_i = b + b_i, \quad (2)
\]

where \(\gamma_i, \beta_i \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}\) are learned parameters, \(i = 1, ..., N_c\) indicates the class, \(N_c\) is the number of classes, and \(\mu, \sigma\) are the mean and standard deviation of \(W_i\). The rationale behind this modulation is that it first removes the source style encoded in \(\mu, \sigma\) to then apply the learned one from \(\gamma, \beta\). This normalization was originally proposed by [8] and called *Adaptive Filter Modulation* (AdaFM) in
Hypernetworks\cite{13,14} aim to learn the parameters $\Theta_h$ of a metamodel, which then will generate the target parameters $\Theta_{tg}$ of the target model $f_{tg}$. In our work we will apply a hypernetwork $g$ to predict the modulation parameters. The input of the hypernetwork is a class embedding vector $v$.

This embedding vector is computed by a class embedding network $C(i; \Psi) = v \in \mathbb{V}$ with $i = 1, \ldots, N_c$ is the class label, $\mathbb{V}$ is the class embedding space, and with network parameters $\Psi$. The hypernetwork $g$ takes this embedding vector and maps it to the modulation parameters according to:

$$
\gamma_v, \beta_v = g(v; \Phi_a), \quad b_v = g_b(v; \Phi_b)
$$

where $g$ are simple affine projections of a point in the space $\mathbb{V}$ implemented by a single fully connected layer, with network parameters $\Phi_a$ and $\Phi_b$. We use $\Phi$ to denote the combination of all the parameters used by the hypernetwork: consisting of $\Phi_a$ and $\Phi_b$ for all the layers in the network.

Here the function $g$ is shared among the target classes. The modulation to produce activations $h_v(x) = W_v x + b_v$ are then given by:

$$
W_v = \gamma_v \odot \frac{W - \mu}{\sigma} + \beta_v, \quad (4)
$$

$$
b_v = b + b_v, \quad (5)
$$

where $W$ and $b$ are the frozen pre-trained weights.

To summarize, a hypernetwork enhanced generator $f$ will be given a class embedding $v$ and a normalized source weight $\tilde{w}$ to produce the desired target weights following Eq. (3) as $f_{\tilde{w}}(v) = \gamma_v \odot \tilde{w} + \beta_v = w$ and pictured in Fig. 2 (top). This new method can be incorporated to unconditional and conditional pre-trained architectures for transfer learning. It is also compatible with pre-existing style modulation techniques, e.g., it is shown implemented in a StyleGAN\cite{18} convolutional block in Fig. 2 (right).

Traditionally, reusability can be introduced in the hypernetwork 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\cite{43}. It is worth noting that this design does not use it since, with the use of transfer learning and the enhanced domain space, each generator can be reduced to a minimum of a learned affine transformation, 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. 5a), mainly because the parameters $\Psi, \Phi$ 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

---

**Figure 2.** Generator $f$ with hypernetwork $g$ (top), regular StyleGAN\cite{18} generator block (left) and block augmented with hypernetwork generators (right). Blocks A and g are affine transformations coming from shared latent spaces $\mathbb{W}$ (frozen) and $\mathbb{V}$ (learned), respectively. Block $B$ is a frozen noise scaling. Original figure from\cite{18}.
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 $\Psi, \Phi$. The alignment is performed between the pre-trained generator network without hypernetwork and the one with hypernetwork (see Fig. 3). 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 $F_{PT}(z) = \{G_{PT}(z^l)\}$ and the ones with hypernetwork by $F_{hyp}(z) = \{G_{hyp}(z, \phi(C(z^0); \Phi)); \}$. After self-initialization we set the class input to the class-embedding $G$ network by $F$ and improves the quality of the generated results. We verify that this significantly reduces the training time.

The alignment is performed between 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 latent vector $z$.

We employ GAN [12] to optimize this problem as follows:

$$L_{gan} = \mathbb{E}_{x \sim X, c \sim p(c)} \left[ \log D(x, c) \right] + \mathbb{E}_{z \sim p(z), c \sim p(c)} \left[ \log(1 - D(G(z, c), c)) \right],$$

where $p(z)$ follows the normal distribution, and $p(c)$ is the domain label distribution.

The final training objective is

$$L_{GAN} = L_{gan} + \lambda_{contr} L_{contr}$$

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

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 [47] for its simplicity and performance and apply it implicitly on the discriminator as in Fig. 4. 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 also left unchanged:

$$L_{contr} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} C_{ij}^2$$

4. Experiments

4.1. Settings

Training details. We devise our architecture based on the structure of a pre-trained StyleGAN. 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 $3 \times 3$ 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 generators are implemented by a single fully connected layer that maps the class branch output to a dimensionality of 512. The proposed method is implemented in PyTorch [38]. Code will be made available upon publication. We optimize the model using Adam [21] with R1 regularization [28] and keep all hyperparameters from the original model, but the model is not trained progressively but fixed at resolution 256. The batch size used overall is 10 unless otherwise specified. We use a Quadro RTX 6000 GPU (24G VRAM) to conduct all our experiments.

**Evaluation metrics.** We report results on two types of metrics: single-valued and double-valued metrics. The former contains Fréchet Inception Distance (FID) [15] and Kernel Inception Distance (KID) [4]. The latter consists of Precision and Recall (PR) [23] and Density and Coverage (DC) [31]. Both PR and DC evaluate the quality and the diversity. We use all training samples available to compute the metrics, since most datasets do not have as much as 10,000 class samples per class as suggested in [15] to have a good metric estimation. KID and DC are multiplied by 100 for easier visualization and PR is given percentage-wise. FID is calculated class-wise.

**Datasets.** Our experiments are conducted on Animal Faces dataset (AFHQ) [7], FFHQ [18], CelebA-HQ [17], Flowers102 [32] and Places 365 [51]. 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 classes is small, we comprise an unconditional dataset. In Places 365 [51] dataset, we take as our target classes 10 categories: amphitheater, aqueduct, castle, dam, field road, fire station, pagoda, underwater - ocean deep, volcano, waterfall. In this paper, all images are resized to $256 \times 256$.

**Baselines.** We are the first to explore transfer learning from unconditional to conditional GANs, there exist only few works with which we can compare. We compare against the following baselines; TransferGAN [45] directly updates both the generator and the discriminator for the target domain. GAN Memory [8] proposed a weight modulation method to address catastrophic forgetting of GAN for lifelong learning. We explore a variant of our method, named as Hyper-GAN*, for which all parameters are updated.

### 4.2. Ablation study

**Hypernetwork.** Comparing the proposed hypernetwork (config. A and B) against explicitly learning the weight modulation for each class (No hypernet.), we see that the former benefits from the joint training and the commonalities of classes to achieve better synthesis quality. It is especially notable for its diversity increase, more than doubling for both Recall and Coverage.

**Self-initialization effect.** Training with an uninitialized hypernetwork (Figs. 5a and 5b) can lead not only to worse time efficiency but worse final quality, as seen in Fig. 5d (blue line). Here we confirm that self-initialization yields huge improvements in training time as well as a significant improvement in quality, see [25], Fig. 5d (green line).

**Class network vs traditional embedding.** In addition to the properties we achieve by including the class embedding network $C$, we can also see an impact in generation quality and diversity (config. B).

**Contrastive learning.** Adding this method resulted beneficial to transfer learning. It delivers improvements with a batch size as little as 10 samples, for which we compute an FID of 37.15 and KID of 1.66, already improving configuration B. Results reported in config. C are computed for a batch size of 60 due to computational constraints, but it should continue improving by growing the size as seen in [47]. Unfortunately, contrastive learning in the generator did not result in improved quality.

**Domain-modulation method.** We wonder what are the options to include domain-specific information on a trained network architecture. AdaFM is especially well-suited.
since it can directly modulate the learned weights, contrary to AdaIN, which modulates at the activation level. However if we disregard that, we can compare to using another AdaIN for the domain modulation. Since the normalization from one block will destroy the information included by the other, we can fix this and simplify the formulation by combining style and class depicted (a graph detailing this is included in the Suppl. Material), as $\gamma_{(s,v)} \beta_{(s,v)} = g(s) + g(v; \Phi)$, corresponding to Eq. (3). 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 method). We also conduct an experiment with updating hypernetwork, self-initialization. Finally, like for the close domain transfer, the proposed techniques (i.e., hypernetwork, self-}

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 animal face. We use the pretrained StyleGAN optimized on AFHQ animal face when the target domain is CelebA human face. As reported in Table 3 (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-GAN) 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-GAN*). Hyper-GAN* further improve the performance.

We also evaluate both the proposed method and the baselines on several other metrics. As reported in Table 4 we achieve the best score on all metrics, which indicates that we not only generate high quality images (corresponding to Pre. and Den.), but also diverse images (corresponding to Rec. and Cov.).

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 3 (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 performing knowledge transfer from unconditional GAN to conditional GAN on far domain transfer.

**Qualitative results.** Figure 6 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 6 on AFHQ). Taking AFHQ as an example, given the target class label Hyper-GAN is able to provide high-quality images (e.g., the second column of Figure 6). When updating all parameters (Hyper-GAN*), we further improve the qualitative result (e.g., the third column of Figure 6).

We further demonstrate that our method has both scalability and diversity in a single model. Each row of Figure Fig. 7 shows the different results when given the target class label. Our method manages to generate diverse images when given different noise inputs. Figure Fig. 7 shows that by changing the target class label (i.e., scalability) the generator produces the corresponding target-specific output.

5. **Limitations**

While Hyper-GAN has demonstrated considerable performance in the unexplored area of transfer learning from unconditioned to conditioned GANs, there remain several lines of future work. One important issue is the method’s memory footprint, since to modulate new weights we have to keep practically the whole pre-trained network in memory. Another point of concern is the discriminator (see Suppl. Material); since it is trained via finetuning, we theorize there’s a maximum number of classes the discriminator can handle at the same time.

6. **Conclusions**

We investigated the knowledge transfer from unconditional GANs to conditionals GANs. To achieve this goal, we proposed a hypernetwork-based method to learn the class embedding for each target class, which further is used to perform weight modulation on the pretrained weights of the unconditional GANs. Training the hypernetwork from scratch complicates training, thus we proposed a self-initialization method that does not require any data to learn the well-initialized weights. To facilitate the capacity of the discriminator, we introduced a self-supervised (contrastive) loss to improve the GAN discriminator. Our qualitative and quantitative results showed the proposed method outperforms the existing state-of-the-art results.
References

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

[2] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In CVPR Workshops, 2015. 1

[3] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip H. S. Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In NeurIPS, 2016. 3

[4] M Brückowski, DJ Sutherland, M Arbel, and A Gretton. Demystifying mmd gans. In ICLR, 2018. 6

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In ICLR, 2019. 1, 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In ICLR, 2020. 3

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

[8] Yuliang Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Watenstein gans. In NeurIPS, pages 2794–2802, 2017. 1, 2

[9] 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. 1, 2

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

[11] 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

[12] 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. 1, 2, 5

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In NeurIPS, pages 5767–5777, 2017. 1, 2

[14] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In ICLR, 2017. 2, 3, 4

[15] 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

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

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In ICLR, 2018. 1, 6

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In CVPR, pages 4401–4410, 2019. 1, 2, 4, 6, 11, 12, 13

[19] 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

[20] Sergey Kasturylin, Dzhamil Zakirov, and Denis Prokopenko. PyTorch Image Quality: Metrics and measure for image quality assessment, 2019. Open-source software available at https://github.com/photosynthesis-team/piq. 11

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

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017. 3

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

[24] Yijun Li, Richard Zhang, Jingwen Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In NeurIPS, 2020. 2, 3

[25] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In CVPR, pages 10551–10560, 2019. 6

[26] 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

[27] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Yang, and Stephen Paul Smolley. Least squares generative adversarial networks. In ICCV, pages 2794–2802, 2017. 1, 2

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

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

[30] Sangwook 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

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

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

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

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

[36] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In CVPR, pages 1717–1724, 2014. 1

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

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Soumith Chintala, and Adam Paszke. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alch´e-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019. 6

[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. 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, 11, 12

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

[42] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In CVPR, pages 10743–10752, 2021. 3

[43] Johannes von Oswald, Christian Henning, Jo˜ao Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In ICLR, 2020. 2, 3, 4

[44] 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

[45] 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, 6

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

[47] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In ICML, 2021. 3, 5, 6

[48] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. In ICLR, 2019. 3

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

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

[51] 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
A. Architecture specifics

In this paper we propose a novel transfer learning strategy from unconditional GAN to conditional GAN by introducing hypernetwork-based adaptive weight modulation. Here we will detail the concrete architecture we used and the changes applied to it.

Figure 8 shows the changes made to a vanilla StyleGAN [18]. The style branch is frozen since we want to keep the learned transformations (pose rotations, color changes, etc.) from the source domain, i.e. FFHQ, unchanged. We do not see a loss in performance when transferred to other datasets (see Fig. 12). The class network C is very similar to the original mapping network and also generates an embedding space, in this case \( \mathcal{V} \), for classes, with the difference that the input comes from a learned class-embedding. The information then comes into each convolution layer to modulate the weights as explained in the main paper.

B. Domain-modulation method diagram

We depict the modulation explained in Section 4.2 (Domain-modulation method paragraph) to validate different class information introduction techniques. It can be seen in Figure 9 (left) how the naive block introduction could destroy the style information (as explained in Section 4.2). On the other hand, in Figure 9 (right) the style and class information are correctly combined. Nevertheless, we have experienced consistent underperformance when compared to the current weight modulation technique used.

C. Additional qualitative results

Qualitative results on Places365 dataset to complement quantitative ones can be seen in Figure 10. A number of 8 classes are shown from a total of 10 selected classes for training, as specified in the experiments.

D. Implementation details

As the base of our method, we use a public StyleGAN implementation \(^1\), which while it is not official, it mostly reproduces results from the original paper. As already mentioned, we keep all hyperparameters from the original paper but fix the resolution growth to the final one, then apply all the methods explained.

In the original StyleGAN paper it is mentioned training instability due to the depth of the mapping network. We experience a similar incident and therefore take the same solution of reducing the learning rate for the class network two orders of magnitude relative to the main network.

For the evaluation metrics, we use a ready-made package [20] for FID, KID and Precision & Recall, which uses the original Inception feature extractor weights, ported to PyTorch. Density & Coverage metrics have been implemented as a package extension, also included in this paper. Perceptual path similarity implementation is taken from \(^2\) applying default center crop.

E. Discriminator details

A diagram of the discriminator and its modifications detailed in the experiments section can be seen in Figure 11. As underlined in the paper, it is worth noting that this module is finetuned from the source domain. Introducing weight modulation into it leads to training instability and worse generation quality overall.

F. Additional baseline results

Additional results of mentioned baselines are shown in Table 5 for experiments on several domains. From this results we can appreciate that cGANTransfer performs badly when the number of previous learned classes is low, since its generation power comes precisely from combining these previous classes. For instance, the first column sets the problem to perform transfer learning from FFHQ (1 class) to AFHQ (3 classes), where this method seems to perform considerably worse than the proposed work, and also worse than simply learning the batch normalization statistics, as GAN Memory. When the number of source domain classes for the transfer is slightly higher, as AFHQ (3 classes) to CelebA-HQ (2 classes), the result is somewhat better since it is allowed more expressivity, but still lacking quality depending on the closeness of the target domain.

Results for several metrics on transfer learning from pretrained FFHQ model to AFHQ dataset are shown in Table 6. We can see how the proposed method performs better than simply learn the normalization statistics for each class as in GAN Memory in terms of quality and diversity, since the knowledge of other classes learned concurrently can be propagated among all of them, resulting in improved time and data efficiency during training. We have already mentioned how cGANTransfer quality degrades when not

\(^1\)https://github.com/rosinality/style-based-gan-pytorch

\(^2\)https://github.com/rosinality/stylegan2-pytorch/blob/master/ppl.py
enough pre-trained classes are given to produce an interpolation. However, we can see how the diversity is better than its quality. We assume this occurs because it can produce close-enough interpolations to resemble the target class, but it doesn’t have a meaningful basis (i.e. a sufficient number of pre-trained classes) in order to form a combination of significant quality.

To perform experiments for cGANTransfer, a BigGAN model trained ImageNet was finetuned on FFHQ dataset. We used the checkpoint\(^3\) of highest resolution publicly available. The model with best FID before collapse was used as the base for this method. The same was performed for AFHQ dataset.

\(^3\)https://github.com/ajbrock/BigGAN-PyTorch
Figure 10. Transfer learning from Animal Faces (AFHQ) to a very distant domain (Places365).

Figure 11. Original discriminator (left), modified discriminator (right). Batch dimensions at the right-most side for easier visualization. The final layer that comes from the convolutions is modified to output $N_c$ number of classes and the correct one is picked to compute the final loss.

| Method      | full   | end   |
|-------------|--------|-------|
| A           | Hyper-GAN | 61.94 | 61.78 |
| B + N space |        | 61.23 | 59.85 |

Table 7. Perceptual path length among classes, for both full paths and endpoints. All scores are in the magnitude of $10^6$.

G. Latent 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 reason, i.e., scarcity of specific training samples, complexity of the modelling, etc. To quantify the beneficial effect of the introduced module, we employ a disentanglement metric called Perceptual path length [18], 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 7 confirm us the advantage of introducing this network. Magnitudes are naturally bigger than style measurements since changes in class are non-trivial perceptual alterations compared to, i.e., color changes.

Finally, we show in Figure 12 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.
Figure 12. Sampling of class interpolation (left to right) versus noise interpolation (up to down). The hypernetwork has learned to keep every aspect of the style of an image intact, including background, while changing the class. The style mechanism was frozen since the beginning of the transfer learning training to animal faces.