Efficient Conditional GAN Transfer with Knowledge Propagation across Classes

Mohamad Shahbazi[1], Zhiwu Huang[1], Danda Pani Paudel[1], Ajad Chhatkuli[1], Luc Van Gool[1,2]

[1]Computer Vision Lab, ETH Zürich, Switzerland   [2]PSI, KU Leuven, Belgium

{mshahbazi, zhiwu.huang, paudel, ajad.chhatkuli, vangool}@vision.ee.ethz.ch

Abstract

Generative adversarial networks (GANs) have shown impressive results in both unconditional and conditional image generation. In recent literature, it is shown that pre-trained GANs, on a different dataset, can be transferred to improve the image generation from a small target data. The same, however, has not been well-studied in the case of conditional GANs (cGANs), which provides new opportunities for knowledge transfer compared to unconditional setup. In particular, the new classes may borrow knowledge from the related old classes, or share knowledge among themselves to improve the training. This motivates us to study the problem of efficient conditional GAN transfer with knowledge propagation across classes. To address this problem, we introduce a new GAN transfer method to explicitly propagate the knowledge from the old classes to the new classes. The key idea is to enforce the popularly used conditional batch normalization (BN) to learn the class-specific information of the new classes from that of the old classes, with implicit knowledge sharing among the new ones. This allows for an efficient knowledge propagation from the old classes to the new ones, with the BN parameters increasing linearly with the number of new classes. The extensive evaluation demonstrates the clear superiority of the proposed method over state-of-the-art competitors for efficient conditional GAN transfer tasks. The code is available at: https://github.com/mshahbazi72/cGANTransfer

1. Introduction

Generative adversarial networks (GANs) [10, 1] are the most common models used for image and video generation, showing very promising results both [21] in unconditional [14, 41, 16] and conditional [3, 34, 4] setups.

Learning from limited data is a well-studied problem in the discriminative setup, where the concept of knowledge transfer [31] between two different but related tasks [42] or domains [9] is ubiquitous. In contrast, the literature on transfer learning for generative adversarial models is fairly limited. One may find this unexpected, since many popular knowledge transfer methods in discriminative setup, in turn, use generative schemes [22, 37]. However, the limited literature is less surprising when the complexity of adversarial training and the mode collapse are taken into account.

A notable work by *Wang et al.* [40], first addressed the problem of training GANs on limited data using a careful fine-tuning (FT) strategy. Following works [27, 39, 44] are the variants of [40] that focus on better fine-tuning strategies. On the contrary, *Noguchi & Harada* [30] proposed the batch statistics adaptation (BSA) technique, by learning only the batch normalization parameters on a small target dataset. As most of the previous works [40, 39, 30] primarily focus on the case of unconditional GANs, we investigate in a different direction of conditional GANs (cGANs). In particular, we are interested in producing new classes given a pre-trained class-conditional cGAN. cGANs are strikingly interesting due to their capability of handling a large number of classes with a single network. For example, *Big-GAN* [3] can generate images from all 1K classes of ImageNet [8]. In fact, *BigGAN* is exploited as the pre-trained network even by the unconditional methods [39, 30]. We refrain from fine-tuning whenever possible, as we believe that new classes can be introduced within such powerful cGANs. Moreover, some powerful pre-trained cGANs, can potentially be used to add new classes in the lifelong learning [19, 30, 33, 5] fashion, which however is beyond the focus of our paper.

In this work, we study how new classes with a limited amount of samples can be added to pre-trained cGANs using knowledge transfer across classes. To do so, inspired by BSA [30], we aim at learning only the batch normalization (BN) parameters that generally encode the class-specific information. Our key idea, different from [30], relies on the assumption that the knowledge between old and new classes can also be transferred by searching for the similarity between them. Our experimental setup, however, does not allow us to access the old data used for the pre-trained model. Therefore, the similarity is searched in the conditional space of the BN parameters, during the training of cGANs. In this process, we learn the similarity scores *explicitly between old and new classes, and implicitly between new ones*. The

learned old-to-new similarity scores are then used to derive the batch statistics of new classes from that of old ones.

It is well-established in [2, 13, 38, 19, 36, 11, 6] and many other works, that learning algorithms can greatly benefit from the shared knowledge between classes. Often, such similarity is either known or discovered when all the classes are accessible. In the context of domain generalization (or in some special case of adaptation), the source data is similarly inaccessible partially or completely [32, 47, 23]. However, the latter assumes that the new classes are either the same or largely overlap with the old ones. Note that in our case, *new classes do not even overlap with the inaccessible old ones*. In addition, almost all aforementioned works seek similarity in the feature space with an exception of [6]. However, [6] is primarily designed to serve the discriminative models. Our generative case, on the other hand, hinders us to access the feature space. We, therefore, rely on the conditional space of cGANs to establish the sought class similarities. Up to our knowledge, we learn the inter-class similarities in the conditional space of the generative models, for the first time.

In summary, we utilize cross-class knowledge while introducing new classes in cGANs. While doing so, active searching of similarity scores between new and old classes with implicit knowledge sharing among new ones is suggested. In this context, we propose a novel method for finding the similarity between new and old classes without requiring access to the old data. The proposed method is particularly suitable when transferring knowledge from pre-trained cGANs.

In summary, the key contributions of our work are as follows:

- We study the new problem of efficient GAN transfer to new classes with explicit inter-class knowledge propagation in pre-trained cGANs.
- A novel method for learning similarity between old and new classes and knowledge sharing within the new classes is proposed using the batch normalization statistics of the old classes, in the conditional space of generative models.
- Our experiments on three benchmark datasets demonstrate the superiority of our method both in terms of generated image quality and the convergence speed.

2. Related Work

Class-conditional GANs. Different architectures and loss functions have been proposed for conditioning GANs on class labels [24, 26, 17]. The current state-of-the-art methods for class conditioning commonly employ cGAN with projection discriminator [26, 25]. In the generator, conditional batch normalization [7] with class-specific scale and shift parameters are applied to each layer of the generator.

The discriminator, on the other hand, is conditioned on the class labels by computing the dot product of the last feature layer and the learnable embedding of the desired class. The performance of the conditional GANs was further improved by adding self-attention layers to the generator and the discriminator [43]. BigGAN [3] was able to reach state-of-the-art performance on image generation from ImageNet, mainly by using a bigger batch size (2048) and some architectural improvements such as a hierarchical latent variable. The conditioning, however, still happens through the class-conditional batch normalization in the generator and the projection layer in the discriminator.

Transfer Learning in GANs. Iterative image generation approaches, such as DGN-AM [29] and PPGN [28], could be considered as early attempts on transfer learning in image generation by generating images via maximizing the activation of the neurons of a pre-trained classifier. TransferGAN [40] is one of the earliest studies addressing transfer learning in GANs. The authors showed that, by simply fine-tuning a pre-trained network on the target dataset, they can outperform training from the scratch in terms of image quality and convergence time. However, naive fine-tuning on small data still suffers from mode collapse and training instability. Another method [44] proposes transferring the low-level layers of the generator and the discriminator from the pre-trained network, while learning the high-level layers from scratch for the target data. In a recent study [27], it is shown that simply freezing the low-level filters of the discriminator is more effective than previous fine-tuning approaches. BSA [30], on the other hand, instead of looking for ways of fine-tuning the network, proposed freezing the weights of the pre-trained generator except the batch normalization parameters. For the target data, new BN parameters are learned without fine-tuning the generator. This allows BSA to add new classes without disturbing the old ones. MineGAN [39] learns a small fully-connected miner network at the input of a frozen pre-trained GAN. The miner learns to shift the prior input distribution to the most suitable one for the target data. After training the miner, Mine-GAN further fine-tunes both the generator and the miner as the final model. MineGAN is designed to transfer knowledge to a single-class target.

Inter-class Knowledge Transfer. A lot of works have been emerging on extending models trained on previous examples/images to perform favorably on new data, and here is where knowledge transfer becomes essential. For instance, [38] uses memory and attention modules to transfer labeled data knowledge to a new class example. A more relevant work [11] uses the class similarity between source classes and a target class to improve the classification performance. [36] is a few-shot meta-learning method that uses the so-called relation score of a new class example with previous examples in order to classify the new example. In a more
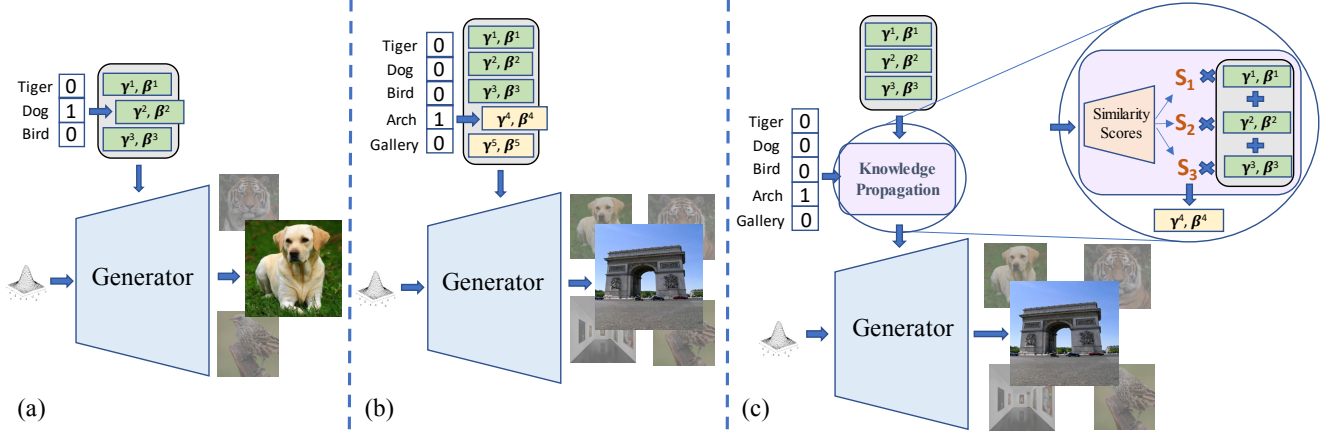
Figure 1: (a): A generator trained on a multi-class dataset. (b): The generator, extended by new classes without taking the old classes into account. (c): The conceptual diagram of our proposed method for conditional GAN transfer with knowledge propagation across classes. Here γ^i and β^i ($1 \leq i \leq 5$) represent the class-specific parameters.

related work [6], BN layers have been used for knowledge transfer from old classes, mainly targeting binary classification with a brief empirical study on style transfer. The knowledge transfer framework of [6] separately performs encoding, pre-selection, and combination of old classes. In contrast, our method jointly optimizes the prior knowledge and the transfer process, leading to an efficient learning paradigm for the generative case.

3. Problem Definition

The task of class-conditional GAN transfer aims at approximating the target data distribution by transferring knowledge from the source multi-class data to the target data using pre-trained GAN models. As shown in Fig. 1 (a), the pre-trained GAN model consists of one generator G and one discriminator D, jointly trained on the source multi-class dataset $\mathcal{X} = \{X^1, X^2, \ldots, X^N\}$, in which X^y is a set of real images in category $y \in \{1, 2, \ldots, N\}$ with the underlying distribution $P_y(x) \in \{P_1(x), P_2(x), \ldots, P_N(x)\}$. In the conditional setup, given a random noise vector z (usually sampled from $\mathcal{N}(0, I)$) and a class label y as inputs, G is trained in an adversarial game with D, to generate an image $x \sim P_y(x)$:

$$x = G(z, y);$$
$$z \sim \mathcal{N}(0, I), \ y \in \{1, \ldots, N\} \quad \text{s.t. } x \sim P_y(x). \quad (1)$$

In the state-of-the-art conditional GANs (*e.g.*, [26, 43, 3]), G is commonly conditioned on the class labels using conditional batch normalization. Specifically, the layer-wise output f_l, of layer $l \in \{1, \ldots, L\}$, is normalized and modulated by the class-specific scale $\gamma_l^y \in \{\gamma_l^1, \ldots, \gamma_l^N\}$

and shift $\beta_l^y \in \{\beta_l^1, \ldots, \beta_l^N\}$ as:

$$f_l' = \gamma_l^y \frac{f_l - \mu_l}{\sigma_l} + \beta_l^y, \quad (2)$$

where μ_l and σ_l represent the batch mean and variance for the l-th layer, and f_l' is the normalized output of the layer. Thus, the class-specific information is parameterized by the corresponding scales and shifts.

Transfer learning in conditional GANs can be defined as exploiting pre-trained GAN models to adapt the generator/discriminator to a new multi-class target data $\mathcal{X}' = \{X^{N+1}, X^{N+2}, \ldots, X^{N+M}\}$ with M new categories $\{y'\}$, $y' \in \{N+1, \ldots, N+M\}$. Mathematically, this task can be defined as the following learning problem:

$$x' = G'(z, y');$$
$$z \sim \mathcal{N}(0, I), \ y' \in \{N+1, \ldots, N+M\} \quad (3)$$
$$\text{s.t. } x' \sim P_{y'}(x), \quad \text{given } G(z, y),$$

where G' is the new generator learned for the new categories $\{y'\}$. It is often desirable that G' can also generate the previous classes, as shown in Fig. 1 (b). The problem of learning such a generator which is capable of generating images for both $\{y\}$ and $\{y'\}$ can be formulated as:

$$x = G'(z, y_f);$$
$$z \sim \mathcal{N}(0, I), \ y_f \in \{1, \ldots, N+M\} \quad (4)$$
$$\text{s.t. } x \sim P_{y_f}(x), \quad \text{given } G(z, y),$$

where we denote the final category set as $\{y_f\} = \{y\} \cup \{y'\}$ and the extended generator as G'.

Transfer learning in the context of GANs is usually approached by a careful fine-tuning of a pre-trained

model [40, 39, 27, 44], with or without modifying the architecture. These architecture modifications include adding either new layers or new batch normalization parameters, to additionally learn the new knowledge of the target data. Such approaches overlook the shared similarities among the old and new classes, resulting in an inefficient knowledge propagation. This motivates us to aim for a more efficient conditional GAN transfer. In particular, we seek for an explicit knowledge propagation from old (*i.e.*, source) classes to the new (*i.e.*, target) ones, along with the possibility of knowledge sharing among target classes. Fig. 1 (c) conceptually illustrates the problem addressed in this paper.

4. Knowledge Transfer Across Classes

We first provide an overview of the proposed method, followed by the details. Our premise is built upon a known observation in BigGAN [3], where interpolating between different classes produces *visually meaningful* intermediate images that do not exist in the training data. This implies that the learned class-specific parameters could lie on a smooth manifold, on which new classes also reside. Consequently, it begs the question: can the similarities between the source and the target classes–as well as those within the target–be exploited to learn the target representations? This question leads us to learn the parameters of the new classes, while being dependent upon the parameters (representing the knowledge) of old classes. More precisely, we propose to obtain the representation of each target class by learning a suitable linear combination over the representations of the source classes – which we call *knowledge propagation*. In addition, we propose a mechanism to enable *knowledge sharing* within target classes, by optimizing the source knowledge in favor of the multiple target classes[1]. We address the exact problem of (4), where the model aims to generate both new and old classes. Such consideration is often ignored in the literature, with an exception of BSA [30]. The details of the proposed knowledge propagation and sharing strategies are presented in Sec. 4.1 and Sec. 4.2, respectively.

4.1. Knowledge Propagation

Following the class-conditioning paradigm discussed in Sec. 3, we embed the class-specific representations in the batch normalization (BN) layers as the scale and shift parameters. Accordingly, knowledge propagation is performed in BN layers. To obtain the BN parameters (γ_l and β_l in (2)) of each new class, our model linearly combines the BN parameters of old classes $\{y\}$ using layer-wise sim-

---
[1]Although the focus of our work is on the multi-class target data, our method can be generalized well to the single-class target as well (See the supplementary material, Sec. F).
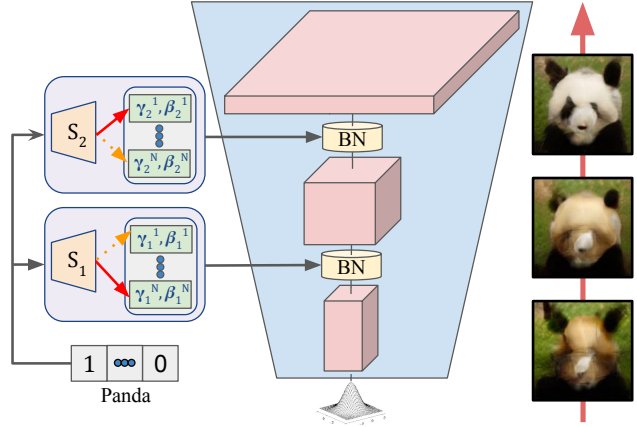

Figure 2: Intuitive visualization of how different layers might borrow information from different classes based on the hierarchy of features such as shape and color (for a better visualization, only two layers are illustrated). Pictures in the figure are obtained from our experiment on AnimalFace dataset [35] (refer to Fig. 9 for more visualizations).

ilarity scores, learned during the training step:

$$
\begin{aligned}
\gamma_l^{y'} &= [\gamma_l^1, \gamma_l^2, \ldots, \gamma_l^N] S_{\gamma_l}^{y'}, \\
\beta_l^{y'} &= [\beta_l^1, \beta_l^2, \ldots, \beta_l^N] S_{\beta_l}^{y'}, \\
y' &\in \{N+1, \ldots, N+M\}, \quad l \in \{1, \ldots, L\}.
\end{aligned}
\tag{5}
$$

Here, $S_{\gamma_l}^{y'} = [s_{\gamma_l}^{(y',1)}, \ldots, s_{\gamma_l}^{(y',N)}]^\top \in \mathbb{R}^N$ and similarly $S_{\beta_l}^{y'} \in \mathbb{R}^N$ are vectors of learned scores for the class y' in layer l. Two things are to be noted here. First, we learn the similarity scores for scale and shift parameters separately. This is because, the new classes could be similar to some of the old classes in terms of their distribution mean, while being similar to another set of classes in terms of intra-class variance. Secondly, we also learn a different set of scores per layer, since different layers of the network do not necessarily benefit from the same set of old classes. It is well-known that different layers of neural networks represent different levels of feature representation. As intuitively shown in Fig. 2, some layers could be responsible for the general shape, some for the color and the texture, and some others for finer details. Such hierarchy of features is also the main motivation of StyleGAN [16] for using layer-specific styles.

Based on the empirical observations, we propagate class-specific knowledge only in the generator and simply fine-tune the discriminator. A similar knowledge propagation in the discriminator leads to performance degradation in our experiments. One possible reason for such degradation is that the knowledge propagation speeds up the over-fitting problem of the discriminator on small datasets [45].

4.2. Knowledge Sharing

In addition to knowledge propagation, we propose a mechanism for knowledge sharing among target classes. Consider transferring a conditional GAN, pre-trained on ImageNet classes, to the AnimalFace [35] dataset containing 20 classes of different animal faces. Although faces of different animals have their own unique characteristics, they still share a level of common structure. Therefore, instead of finding each class representation independently, it is reasonable to exploit the shared knowledge between all target classes. A basic but indirect example of knowledge sharing between the target classes can be seen in methods based on fine-tuning the pre-trained convolutional filters, where all target classes contribute to optimizing the same filters. The same, however, does not happen when learning the class-specific BN parameters independent of other classes.

To empower the knowledge sharing among target classes in our method, we propose to allow the target classes to jointly optimize the prior knowledge (pre-training BN parameters) during knowledge propagation. Optimizing the prior knowledge will enable the model to obtain a shared set of intermediate representations, which are more suitable for all target classes. These shared representations–which we name pseudo-classes–then can be combined according to the similarity scores during the knowledge propagation step. Mathematically, we rewrite the propagation equation after knowledge sharing as:

$$\gamma_l^{y'} = [\hat{\gamma}_l^1, \hat{\gamma}_l^2, \ldots, \hat{\gamma}_l^N] S_{\gamma_l}^{y'},$$
$$\beta_l^{y'} = [\hat{\beta}_l^1, \hat{\beta}_l^2, \ldots, \hat{\beta}_l^N] S_{\beta_l}^{y'}, \quad (6)$$
$$y' \in \{N+1, \ldots, N+M\}, \quad l \in \{1, \ldots, l\}.$$

where $[\hat{\gamma}_l^y]$ and $[\hat{\beta}_l^y]$ are obtained by updating a copy of the source BN parameters, forming the shared representations for layer l. It is important to note that the modified BN parameters in (6) are used only for learning the new classes. We do not replace original old BN parameters, as we want to preserve the old knowledge during the transfer. This process does not further increase the number of parameters for inference, since the updated parameters (and the similarity scores) are discarded after learning the target BN parameters. Fig. 3 visualizes the proposed approach for knowledge sharing in conjunction with the knowledge propagation.

4.3. Training with Residuals and Sparsity

For the purpose of generalizability, we also consider the classes that cannot be represented well only by combining the shared intermediate representations. To address this issue, we propose to add residual vectors $r_{\gamma_l}^{y'}$ and $r_{\beta_l}^{y'}$ to the scale and shift parameters obtained from knowledge propagation, respectively. However, in order to encourage the model to use the prior knowledge as much as possible, we
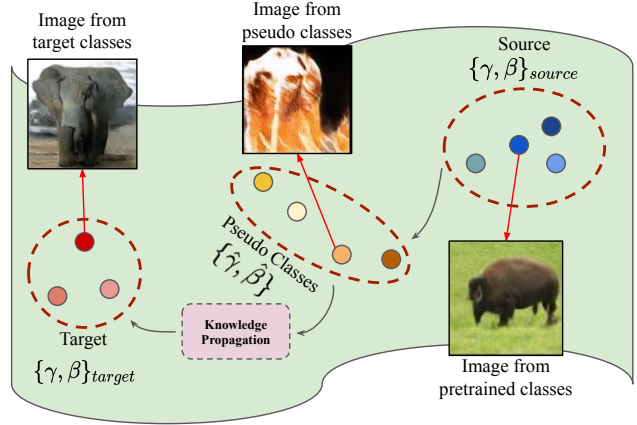

Figure 3: The visualization the proposed knowledge sharing and propagation. The sharing takes place in obtaining the pseudo-classes, by updating the source. The propagation step combines pseudo-classes to obtain the target. The sharing and the propagation are performed jointly. The images are obtained from our experiments (Please refer to Fig 7-8).
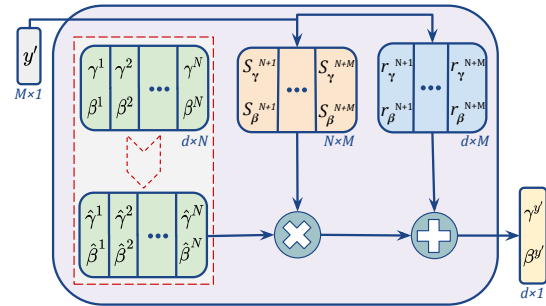

Figure 4: An overview of the proposed block for cGAN transfer with knowledge propagation across classes.

minimize the magnitude of the residual vectors using ℓ_2 regularization. Moreover, to encourage the new classes to learn from the most relevant prior knowledge, we also add an ℓ_1 sparsity regularization on the similarity scores.

In summary, we propose conditional GAN transfer from a pre-trained GAN to multi-class target data via BN parameters of pre-training classes, which are linearly combined to obtain the new classes' representations. Moreover, we enable sharing knowledge within the target classes by allowing them to update the prior knowledge according to the needs of the target data. An overview of the complete proposed method is illustrated in Fig. 4.

The final loss function of our method follows (7), where L_g is the adversarial loss [10, 1, 20] used for training the GAN, L_r is an ℓ_2 regularization over the residuals, and L_s

Method/ Experiment	20/ 600	20/ 300	20/ 100	10/ 600	10/ 300	10/ 100	mFID
Scratch	**25.35**	47.20	81.90	52.23	62.17	83.68	58.76
TransferGAN	27.67	31.19	48.67	30.58	40.74	64.75	40.60
BSA - early	37.12	32.10	50.88	42.01	40.34	66.76	44.87
BSA - full	28.03	**30.89**	40.73	30.73	35.62	56.48	37.08
Ours	29.10	30.99	**40.04**	**29.95**	**35.23**	**54.95**	**36.71**

Experiment	FID	KMMD
Scratch*	190	0.96
TransferGAN*	89.2	0.53
PPGN*	139	0.56
MineGAN*	82.3	0.47
BSA**	85.9	0.18
Ours**	**71.1**	**0.16**

Table 1: Results of the evaluation and the comparison of the proposed method using FID and KMMD score. Left: transferring 80 classes of CIFAR100 to 20 classes or 10 classes. Different columns report the best FID scores scores in difference cases. The format A/B means using A classes and B images per class for the target. BSA-early indicates the FID scores for BSA at the iteration where our method achieves its best FID score. Right: Results of transferring ImageNet to 5 classes of Places365 with 500 samples per class. * adopted from [39]. Some discrepancy between the KMMD scores of * and ** is possible[2].

is an ℓ_1 regularization over the similarity scores:

$$
\begin{aligned}
L &= L_g + \lambda_r L_r + \lambda_s L_s, \\
&= L_g + \lambda_r \sum_{l=1}^{L} \sum_{y'=N+1}^{N+M} \{||r_{\gamma_l}^{y'}||_2^2 + ||r_{\beta_l}^{y'}||_2^2\} \\
&\quad + \lambda_s \sum_{l=1}^{L} \sum_{y'=N+1}^{N+M} \{|S_{\gamma_l}^{y'}| + |S_{\beta_l}^{y'}|\},
\end{aligned}
\tag{7}
$$

where, λ_r and λ_s are the hyper-parameters associated to residual and sparsity losses, respectively.

5. Experiments

In this section, we provide the details of our experiments, the evaluation of our method (cGANTransfer), and its comparison with the baseline methods in two different setups. Then, we provide further analysis of our contributions with an ablation study and explanatory visualizations.

5.1. Experimental Setup

Datasets: To evaluate our method, we use two main experimental setups. In the first setup, we pre-train the network on 80 randomly-selected classes of CIFAR100 [18]. The remaining classes are used as the target. To have a more thorough evaluation, we evaluate our method on CIFAR100 with different numbers of target classes and images per class. For the second setup, we consider the more challenging task of extending a network pre-trained on ImageNet [8] to the Places365 dataset [46]. Following MineGAN [39], we select 5 classes (i.e., Alley, Arch, Art Gallery, Auditorium, Ballroom) and down-sample each class to 500 images. In addition, we use AnimalFace dataset [35]– containing 20 classes–for further analysis and visualizations. We down-sample each class to contain a maximum of 100 images.

---

[2]Our choice of the width of the Gaussian kernel (σ) for calculating KMMD may differ from that of [39].

Architecture: For the cGAN architecture, we use Big-GAN [3] with hinge loss, as it is one of the most widely-used state-of-the-art cGANs. We use hierarchical noise for training on ImageNet, but not for CIFAR100, following the experimental setup of BigGAN [3].

Training: To maintain our experiments in the class extension setup, we freeze the weights of the generator and only learn the parameters of our knowledge transfer block. We also find that, if the hierarchical noise is used (e.g. ImageNet setup), it is also necessary to fine-tune the linear layers that project the noise into the BN parameter's space. The reason might be that these linear layers are optimized to add the detailed style of the dataset to the generated images. Experiments on CIFAR100 are conducted using a single V100 GPU with a batch size of 50, and the experiments on the second setup are performed using 8 V100 GPUs with a batch size of 256.

5.2. Comparison with the State-of-the-Art

To quantitatively evaluate our method, we use Fréchet Inception Distance (FID) [12]. We also provide the KMMD metric (Gaussian kernel with $\sigma = 1$) for ImageNet setup, following BSA and MineGAN [30, 39]. We compare our method against training from the scratch (Scratch), TransferGAN [40], PPGN [28], MineGAN [39] and BSA [30]. Among these methods, BSA is the only one that performs the class extension. Therefore, we consider BSA as our main baseline.

Table 1 (left) shows the results for evaluating our method on CIFAR100 and its comparison with the other methods. The experiments include two different numbers of classes (20 and 10), and 3 different numbers of samples per class (600, 300, 100). As it can be seen, decreasing the number of training samples degrades the performance of learning from scratch and TransferGAN significantly. On the other hand, BSA and our method perform more robustly on small data. Compared to BSA, our method achieves slightly better FID scores (possibly because of the less challenging task on CIFAR) and a significantly better convergence
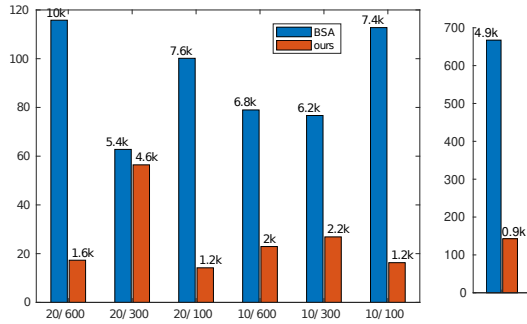
Figure 5: The convergence comparison between BSA and ours on CIFAR (left) and Places365 (right) setups. Note that the number of iterations is shown above each bar.
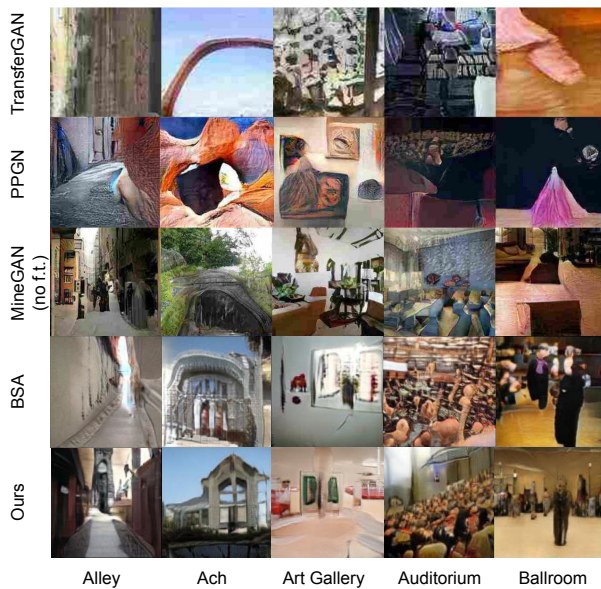

Figure 6: The visual comparison between our proposed method and baseline methods on Places365. The first three Competitors' results are adopted from [39].

speed (more than **4x** speed-up on average), as visualized in Fig. 5 (left). Our method enjoys such speed-up thanks to the knowledge propagation through the class representations, as opposed to its counterpart–BSA. The superiority of the proposed method becomes even more noticeable in the more challenging setup of transferring ImageNet to Places365. The classes of Places365 have large intra-class diversity, and there is a considerable domain gap between the source and target datasets. Therefore, the task becomes highly challenging when only use 500 images per class are used. Table 1 (right) presents the results for transferring ImageNet to 5 classes of Places365. As it can be seen, our method clearly outperforms the other methods in terms of FID and KMMD. Fig. 5 (right) again shows that the proposed method is more than **5x** faster than BSA in terms of

Experiment	Best FID	Iterations
Freezed Prior, w/o Res	96.3	3000
Freezed Prior, with Res	79.4	1100
Tunable Prior, w/o Res	81.2	4000
Shared W, Tunable Prior, w/o Res	79.8	3400
Final architecture w/o reg	82.4	2400
Final architecture w/o l1	84.7	1600
Final architecture w/o l2	81.1	2100
Final architecture, with l1 & l2	**71.1**	**900**

Table 2: Ablation study over GAN transfer from ImageNet to Places365. Iterations: iteration number for the best FID.

convergence speed. The comparison between the results of our method and BSA shows that knowledge transfer across classes results in not only faster convergence but also a more accurate distribution approximation of the target data. Fig. 6 depicts the visual results for our method and the compared ones in ImageNet-to-Places365 setup.

5.3. Ablation Study

To show the effect of each component in our method, we perform the following experiments on ImageNet-to-Places365 setup: Transferring from previous classes without updating the prior knowledge or residual learning, transferring from updatable prior knowledge without residual learning, updatable prior with residuals, sharing the combination weights between the layers, and the experiments on the regularizations. Table 2 shows the results for the ablation study of our method when transferring from ImageNet to Places. The ablation study shows that, all the proposed components contribute meaningfully to the final model.

5.4. Fine-tuning

As mentioned in Sec. 5.1, we freeze the weights of the generator in our experiments. This enables a parameter-efficient extension of the cGAN to target classes. However, when parameter efficiency is not a constraint, fine-tuning the filters could further improve the performance of the model, as shown in [39]. Therefore, we also provide the results of further fine-tuning of our model after learning the target BN parameters in Table 3. To fine-tune the model, we freeze the knowledge propagation parameters (prior knowledge and similarity scores) and only fine-tune the residuals, as well as the rest of the model. The results show the additional benefit of further fine-tuning of our model.

Dataset	FID	KMMD
Places365	65.48	0.156
AnimalFace	68.92	0.194

Table 3: The results of further fine-tuning of our model on Places365 and AnimalFace.

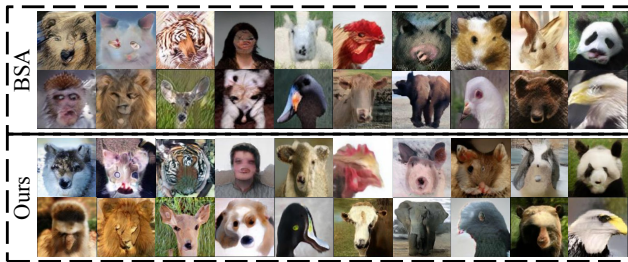| Experiment	FID	KMMD	Iters	Time (Mins)
BSA	91.9	0.25	4800	650
Ours	**85.9**	**0.23**	**1400**	**235**

Figure 7: Quantitative (top) and qualitative (bottom) results of transferring ImageNet to 20 classes of Face Animal without (BSA) and with (Ours) **cross-class knowledge propagation**. Iters: training iteration number for the best FID.

5.5. Analysis and Visualizations

To better understand the proposed method, we further analyze its main components while transferring ImageNet to AnimalFace. The quantitative (FID, KMMD, and convergence time) and qualitative results of class extension to AnimalFace, with and without class knowledge propagation, are shown in Fig. 7. The results for AnimalFace are consistent with the previous results provided in Sec. 5.2.

Knowledge sharing within target classes: An important component of our method is knowledge sharing within the target classes. Fig. 8. shows the pseudo-classes (intermediate representations) obtained by our method after training on the target data. To visualize these representations, we directly sample the BN parameters from the updated base representations, instead of combining them to obtain the new class representations (refer to Fig. 3). As it can be seen, the initial class representations of the ImageNet are transformed to some intermediate "pseudo-class" representations that contain the shared structure of the target faces, but do not belong to a particular target class. Thus, our method is able to obtain the representations of each new class by combining these new pseudo-classes, which are learned using all of the target classes.

Layer-specific knowledge transfer: Another key aspect of our method, as discussed in Sec. 4 (refer to Fig. 2) and supported by the ablation study (Table 2), is that the linear transfer weights are different for each layer. Fig. 9 visualizes the effect of using different weights per layer. To obtain these visualizations, after training the network on the target, we use the learned similarity scores of the first layer for all the layers. Then, we gradually introduce the learned scores of the next layers. From the results, we see that the first layers are mostly responsible for the general object shape, and the later layers introduce color and finer details.

Figure 8: **Knowledge sharing.** The intermediate shared representations learned for AnimalFace as the target data.
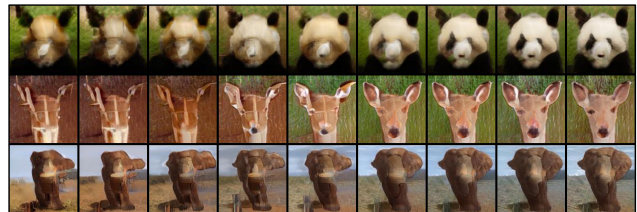

Figure 9: **Layer-wise knowledge transfer.** Different layers introduce different information to the generated images. The shape is mostly changed by the starting layers, while colors and finer details are added by the later layers.

6. Conclusion and Future Work

In this paper, we studied the problem of conditional GAN transfer by transferring the knowledge across both source and target classes. We represented the knowledge of individual classes by their respective batch normalization parameters, which are used for conditioning during the generation. To propagate the knowledge to new classes, we introduced a method that learns to update and combine the batch normalization parameters of the source classes. The evaluations on three standard benchmarks demonstrate a clear advantage of our method, both in terms of training efficiency and the image generation quality (in terms of FID and KMMD), compared to the state-of-the-art methods. Our ablation study showed the importance of jointly using the update and combination steps, which we referred to as sharing and propagation, respectively.

We believe our study can be followed by several interesting future works, such as further knowledge propagation on the discriminator, its integration with differentiable augmentation [15, 45], and the extension to lifelong learning.

Acknowledgements

References

[1] Martin Arjovsky, Chintala, Soumith, and Léon Bottou. Wasserstein generative adversarial networks. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[5] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. In *Advances in Neural Information Processing Systems*, pages 469–477, 2020.

[6] Gratianus W. P. Data, Kirjon Ngu, David W. Murray, and Victor A. Prisacariu. Interpolating convolutional neural networks using batch normalization. In *Proceedings of the European Conference on Computer Vision*, 2018.

[7] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[11] Yuchen Guo, Guiguang Ding, Yue Gao, and Jungong Han. Active learning with cross-class similarity transfer. In *Association for the Advancement of Artificial Intelligence*, pages 1338–1344, 2017.

[12] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[15] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems*, 2020.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[17] Ilya Kavalerov, Wojciech Czaja, and Rama Chellappa. cgans with multi-hinge loss. *arXiv preprint arXiv:1912.04216*, 2019.

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[19] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[20] Jae Hyun lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[21] Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. *arXiv preprint arXiv:2008.02793*, 2020.

[22] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477, 2016.

[23] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. cgans with projection discriminator. *arXiv preprint arXiv:1802.05957*, 2018.

[26] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018.

[27] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze discriminator: A simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*, 2020.

[28] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017.

[29] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[30] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2750–2758, 2019.

[31] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[32] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017.

[33] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[35] Zhangzhang Si and Song-Chun Zhu. Learning hybrid image templates (hit) by information projection. *IEEE Transactions on pattern analysis and machine intelligence*, 34(7):1354–1367, 2011.

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[38] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[39] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9332–9341, 2020.

[40] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision*, pages 218–234, 2018.

[41] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019.

[42] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[43] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[44] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for limited-data generation. *arXiv preprint arXiv:2002.11810*, 2020.

[45] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[46] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.

[47] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. Domain generalization with optimal transport and metric learning. *arXiv preprint arXiv:2007.10573*, 2020.