Transformation GAN for Unsupervised Image Synthesis and Representation Learning

Jiayu Wang[1] Wengang Zhou[1] Guo-Jun Qi[2,4] Zhongqian Fu[3] Qi Tian[1] Houqiang Li[1]
[1] CAS Key Laboratory of GIPAS, Department of EEIS,
University of Science and Technology of China
[2] Laboratory for MAchine Perception and LEarning (MAPLE)
[3] Department of EST, University of Science and Technology of China
[4] Futurewei Technologies
wjy1031@mail.ustc.edu.cn,{zhwg,zqfu,lihq}@ustc.edu.cn,guojunq@gmail.com,wywqtian@gmail.com

Abstract

Generative Adversarial Networks (GAN) have shown promising performance in image generation and unsupervised learning (USL). In most cases, however, the representations extracted from unsupervised GAN are usually unsatisfactory in other computer vision tasks. By using conditional GAN (CGAN), this problem could be solved to some extent, but the main shortcoming of conditional GAN is the necessity for labeled data. To improve both image synthesis quality and representation learning performance under the unsupervised setting, in this paper, we propose a simple yet effective Transformation Generative Adversarial Networks (TrGAN). In our approach, instead of capturing the joint distribution of image-label pairs $p(x, y)$ as in conditional GAN, we try to estimate the joint distribution of transformed image $t(x)$ and transformation t. Specifically, given a randomly sampled transformation t, we train the discriminator to give an estimate of input transformation, while following the adversarial training scheme of the original GAN. In addition, intermediate feature matching as well as feature-transformation matching methods are introduced to strengthen the regularization on the generated features. To evaluate the quality of both generated samples and extracted representations, extensive experiments are conducted on four public datasets. The experimental results on the quality of both the synthesized images and the extracted representations demonstrate the effectiveness of our method.

1. Introduction

As a fundamental task in computer vision, representation learning has received lots of attention over the last decades. Thanks to the strong representational power of deep neural networks (DNN), models combined with DNN have demonstrated tremendous successes in various computer vision tasks, including image classification, semantic segmentation and image generation. Moreover, in practice, with the DNN pre-trained on large scale datasets for classification, the extracted representations could be transfered to other tasks [28], and even to other modalities [11]. However, the training methodology of deep neural networks is mainly driven by fully-supervised approaches with a large volume of labeled data. With such a limitation, when only a limited amount of labeled data is available, it becomes a highly challenging problem to train DNN models effectively. Therefore, as an effective DNN training method without the need for extensive manual annotations, unsupervised learning has received more and more attention [9, 14, 33, 40].

Generative models, for example, governed by probabilistic approaches, are trained to capture real data distribution using unlabeled datasets. In order to produce new content, generative models are required to have a good understanding of the training data, which makes them also effective in unsupervised learning tasks. One class of generative models that has been applied to representation learning is Generative Adversarial Networks (GAN) [10]. Since the discriminator is trained to extract features that is essential for distinguishing real data from generated one, the intermediate features from discriminator can be viewed as the extracted representations of its input [27]. However, it has been observed that the training dynamics in GAN are often unstable. As a result, the generated distribution varies during the training process, which poses negative influence on representation learning of the discriminator. This issue is usually addressed by conditional image generation, *i.e.,* conditional GAN [24, 36]. For example, in TripleGAN [19], the model characterizes the uncertainty of both the real data x and

the label information y, and estimates a joint distribution $p(x, y)$ of input-label pairs. Such a method introduces additional class information so as to encourage the discriminator to learn more stable representations, but doesn't conform to the perspective of GAN as a tool for unsupervised learning.

In addition to GAN, another kind of unsupervised learning methods called self-supervised learning have demonstrated its great potential for no need of manually labeled data. Leveraging the self-supervised information directly from training data themselves, such kind of methods create self-supervised objectives to train the networks. Recently, Chen *et al.* [32] proposed the self-supervised GAN (SS-GAN), in which an auxiliary image rotation degree classification objective was incorporated into the adversarial training process. And the discriminator was trained to predict the angle of rotation based only on rotated images. The advantages of such integration is that the whole model could inherit the benefits of conditional GAN without labeled data. However, in SS-GAN, the rotation-detectable regularization is only applied on the generator's output, it is still not sufficient enough. Actually, it is also important, for improved quality of generated images, to regularize the internal features of generator, and this work presents a feature-transformation matching approach to meet such a requirement.

In this work, we propose Transformation Generative Adversarial Networks (TrGAN) for improving unsupervised image synthesis and representation learning. We follow the collaborative adversarial training scheme of the self-supervised GAN [32], and re-design the self-supervision method as well as the training methodology. Inspired by Auto-Encoding Transformation (AET) [39], we adopt projective transformation to replace image rotation, and train the model to estimate the transformation based on both of original images and their transformed counterparts. In other words, we force the model to capture the relative change of geometric structures caused by the given transformation. Then, we separate both discriminator and generator into several blocks, and match the internal features between discriminator and generator. In addition, we further introduce feature-transformation regularization on internal features of generator. In other words, the generator is required to generate images and internal features that are both transformation-detectable. The main framework of our model is illustrated in Figure 1.

In short, we summarize the contributions of our work as follows:

- We propose a feature-transformation matching approach through which the proposed model can more effectively capture the real data distribution.
- The intermediate feature matching between discriminator and generator provides additional supervision on the feature space of the generator, which in turn improves the quality of features extracted from the discriminator.
- The proposed TrGAN model improves the quality of both generated images and extracted representations on multiple widely used datasets. Under the same experimental setting, our model achieves FID even better than its conditional counterpart: Projection Conditional GAN [21].

2. Related Work

Auto-Encoder. One of the most representative unsupervised learning methods is Auto-Encoder. During the training, the encoder is trained to output sufficient representations to reconstruct original images by the corresponding decoder. The common belief is that to reconstruct the input images, the extracted features should contain sufficient information. Many variants of Auto-Encoder [14, 16, 33, 34] have been proposed, in which the encoder acts as an unsupervised features extractor after being jointly trained with the decoder. For example, the variational auto-encoder [16], in which the distribution of features from the encoder is constrained to a prior distribution. In order to learn more robust representation, Denoising auto-encoder [33] is designed to reconstruct noise-corrupted data. Contrastive Auto-Encoder [29] aims to extract representation invariance to small perturbation.

GAN. In recent years, GAN has gained significant popularity in image generation tasks, in practice, it deals with the generation tasks by approximating a proper mapping relation between data distribution and low-dimensional distribution. Specifically, a random noise z is fed into the generator G to obtain a sample $G(z)$. And the discriminator D is required to distinguish between real samples and generated ones.

Benefiting from the flexibility of GAN's framework, adversarial training methodology has been successfully leveraged to many traditional tasks, including image super-resolution [18] and image-to-image translation [42]. Besides, the most representative one is unsupervised representation learning. For example, DCGAN [27] used the intermediate features from discriminator as the representations of the input images. On the other hand, the input of the generator, *i.e.,* noises, can be viewed as the representations of the output images. In [6, 8], an extra encoder was trained as an inverted version of the corresponding generator. Given an input image, the output noise of the encoder can be used as the extracted representation. However, as discussed in previous work [31], the training dynamics in GAN are quite unstable, which poses negative influence on the quality of the generated images. A large category of GAN variants have been proposed to address this problem, and recent de-

velopment has shown the promising performance of conditional GAN (CGAN) in stable training and expressive representation learning [19, 21, 23]. But the main shortcoming of CGAN is that they usually require a large volume of labeled data, which is sometimes impractical to scale.

Self-Supervised Learning. In addition to the aforementioned methods, a special paradigm called self-supervised learning has lead to a tremendous progress in unsupervised learning. Rely on the visual information present on the training data, such methods could be applied in label-free tasks. For example, Zhang *et al.* [40, 41] trained the model to predict the missing channels with only a subset of the color channels of images as input. Gidaris *et al.* [9] proposed to train the model by predicting the $2D$ rotations of four discrete angles that is applied to the image. Doersch *et al.* [5] predict the relative positions of sampled patches from the input image. Agrawal *et al.* [1] use the motion of a moving vehicle between two consecutive frames as self-supervised information. More recently, Zhang *et al.* [39] propose a novel Auto-Encoding Transformation (AET), in which the model is trained to learn representations by reconstructing input transformations. In self-supervised GAN [32], an auxiliary, self-supervised objective is integrated into the adversarial loss instead of manually labeled data.

Intermediate Feature Matching. In context of knowledge transfer, intermediate feature matching (IFM) is a training method that uses the internal features from a pre-trained model to guide another model [12, 30]. It has also been adopted in some variants of conditional GAN, such as CVAE-GAN [2] and Stacked GAN [35], in which the intermediate representations from a pre-trained classifier are used to further guide the generator. Benefiting from the transformation predicting objective, the features extracted by the discriminator could contain enough information about the visual structures, and thus make the IFM more effective in our TrGAN. Our method can be considered as a special kind of knowledge transfer. But different from CVAE-GAN and Stacked GAN, the intermediate representations used for IFM in our model are from the discriminator, and with no requirement of labeled data.

3. Proposed Approach

In the following, we first review the background of GAN in Section 3.1 for the sake of completeness. We then describe the main framework of Transformation Generative Adversarial Networks in Section 3.2. After that, we elaborate the intermediate feature matching used in our model in Section 3.3. In Section 3.4, we will focus on our proposal of feature-transformation matching. We summarize the details of the whole framework in Section 3.5.

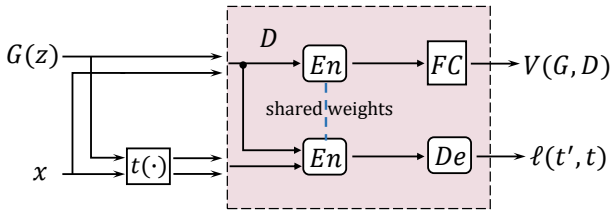


Figure 1. The main framework of the global discriminator D.

3.1. Background of GAN

The Generative Adversarial Networks consist of two main components: a generator G as well as a discriminator D. The generator G tries to generate data directly from low-dimensional noise input, while the discriminator D is required to capture distinguishing features of real data and distinguish between real and generated images. Let $p(x)$ represent real data distribution, and $p(z)$ represent the distribution of noise input. The original adversarial loss function is written as:

$$\begin{aligned}\min_G \max_D V(G,D) =& \mathbb{E}_{x\sim p(x)}[\log D(x)] \\ &+ \mathbb{E}_{z\sim p(z)}[\log(1-D(G(z)))].\end{aligned} \tag{1}$$

3.2. Transformation Generative Adversarial Networks

Recent years have witnessed many applications of conditional GAN to semi-supervised learning. The idea of conditional GAN is to capture the joint distribution of image-label pairs $p(x,y)$. While for our TrGAN, we aim to estimate the joint distribution of transformed image and transformation. Given a transformation t from a distribution $p(t)$, we apply it to a random image x, and get a transformed image $t(x)$. The joint distribution $p(t(x),t)$ can be factorized in two ways, namely, $p(t(x),t) = p(t)p(t(x)|t)$ and $p(t(x),t) = p(t(x))p(t|t(x))$. The conditional distributions $p(t(x)|t)$ and $p(t|t(x))$ are critical for image transformation and transformation predicting, respectively.

To jointly estimate these conditional distributions, TrGAN consists of two main components: (1) a global discriminator D that approximately characterizes $p\left(t'|t(x)\right) \approx p(t|t(x))$, in which a single encoder-decoder network with two heads is used to distinguish real images from generated images and predict the transformation. As shown in Figure 1, the encoder En extracts the feature from a given sample x. Then, $En(x)$ and $En(G(z))$ are fed into the final fully-connected layer to compute the adversarial loss $V(G,D)$. Meanwhile, the decoder De is trained to reconstruct the parameters (t') of the corresponding input transformation based on the features $En(x)$ and $En(t(x))$; (2) a generator G that approximately characterizes $p\left(t(G(z))|t\right) \approx p(t(x)|t)$. In this way, given a random transformation t, the goal of generator is not only to gen-

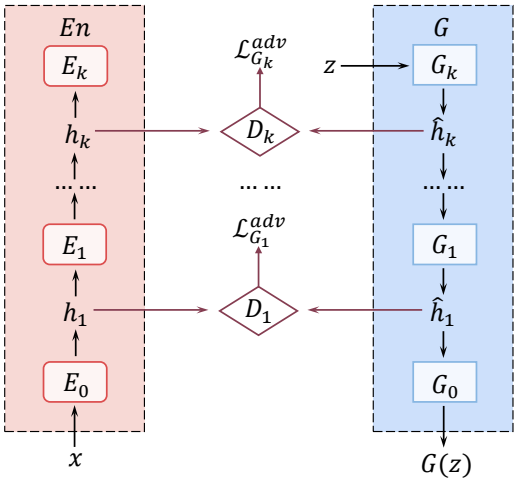


Figure 2. The workflow of intermediate feature matching.

erate images that are indistinguishable from real ones, but also to ensure the transformation-detectable property of the generated images.

For generated images $G(z)$, transformation-detectable means: with the discriminator D that is trained to predict the transformation t only on real images x and $t(x)$, when providing $G(z)$ and $t(G(z))$ as input, the corresponding t can still be correctly predicted. Such regularization could force $G(z)$ to have similar high-quality visual structures that is essential for transformation prediction as real images.

The generator G as well as the global discriminator D are trained jointly by iteratively updating each other with respect to the adversarial loss $V(G, D)$. Meanwhile, the transformation predicting loss $\ell(t', t)$ is added to the discriminator, where $t' = D[x, t(x)]$. Note that the global discriminator D is trained to predict the transformations based only on real images x and $t(x)$. For parameterized transformations, each transformation t_θ can be represented by its own parameters θ. We use the loss function written as: $\ell(t', t) = \frac{1}{2}\|\theta' - \theta\|_2^2$. And the global loss functions for G and D are listed as follows:

$$\mathcal{L}_D = -V(G, D) + \alpha \mathbb{E}_{x\sim p(x)} \mathbb{E}_{t\sim p(t)} \ell(t', t), \quad (2)$$

$$\mathcal{L}_G^{global} = V(G, D) + \beta \mathbb{E}_{x\sim p_g(x)} \mathbb{E}_{t\sim p(t)} \ell(t', t). \quad (3)$$

The transformation-detectable regularization is only applied on the generated images. However, to better utilize the advantages of self-supervised learning, we can further introduce an extra regularization on the feature space of the generator. Such a regularization could provide an label-free supervision directly on the feature space of the generator.

3.3. Intermediate Feature Matching in TrGAN

In previous works, such as Stacked GAN [35], the intermediate feature matching is usually accompanied by a pre-trained classifier. While in our model, the transformation prediction task requires the discriminator to extract more useful information about the visual structures from inputs. As a result, we can directly use features from the discriminator to further guide the generator through IFM.

Encoder Blocks. Let h_i represent the feature of original image x, h_{it} is the feature of transformed image $t(x)$. To implement IFM method, as shown in Figure 2, we first separate the encoder En into several blocks E_i, where $i \in \{0, 1, \cdots, k\}$, and $k+1$ is the number of blocks. Each block E_i acts as a nonlinear mapping function between intermediate features. Specifically, the higher-level feature h_{i+1} is achieved by feeding the lower-level feature h_i into block E_i, (*i.e.*, $h_{i+1} = E_i(h_i)$). Note that $h_0 = x$ is the input image, and the number of blocks $k+1$ is determined by the resolution of input image.

Generator Blocks. From different encoder blocks, we could learn features of different levels, and the features of higher level contain more advanced semantic information. Thus, the intuition for adopting IFM is straightforward: the generated features should also be decomposed into multiple levels, with progressively increased semantic information. Similar to encoder blocks, as shown in Figure 2, the generator G is also decomposed into several blocks: G_i, where $i \in \{0, 1, \cdots, k\}$, and $k+1$ is the number of blocks. Each block G_i is trained as an inverted version of the corresponding E_i. Specifically, each block G_i receives generated feature $\hat{h}_{i+1}$ from upper block G_{i+1} as input, and outputs feature $\hat{h}_i$. Note that $\hat{h}_0 = G(z)$ is the generated image.

To transfer knowledge from encoder blocks to generator blocks of the same level, we use adversarial loss to match intermediate representations as in Stacked GAN. Specifically, for each encoder block E_i ($i \neq 0$) and generator block G_i ($i \neq 0$), we introduce a feature discriminator D_i. During the adversarial training, D_i is trained to distinguish generated features $\hat{h}_i$ from extracted features h_i, and G_i is trained to "fool" the D_i. The loss functions for G and each D_i are listed as follows:

$$\mathcal{L}_{D_i} = -V(G_i, D_i), \quad (4)$$

$$\mathcal{L}_G^{adv} = \sum_{i=1}^{k} \mathcal{L}_{G_i}^{adv} = \sum_{i=1}^{k} V(G_i, D_i), \quad (5)$$

where $V(G_i, D_i)$ is the corresponding adversarial loss.

3.4. Feature-Transformation Matching in TrGAN

The generator G in our model has three training objectives: (1) G is trained to generate images $G(z)$ that are indistinguishable from real images x; (2) G_i is trained to generate intermediate features $\hat{h}_i$ that are indistinguishable from extracted features h_i; (3) G is trained to generate images $G(z)$ that are transformation-detectable as real images x. Intuitively, the generated features $\hat{h}_i$ should also be transformation-detectable as extracted features h_i.

Based on the above discussions, we propose a novel feature-transformation matching (FTM) regularization to encourage $\hat{h}_i$ to contain more quality visual information.

Let us denote the mapping relation between h_i and h_{it} as $f_{it}(\cdot)$. Given an arbitrary t, we have $h_{it} = f_{it}(h_i)$. For generated features $\hat{h}_i$, transformation-detectable means: with the discriminator D that is trained to predict the transformation t only on extracted features h_i and h_{it}, when providing $\hat{h}_i$ and $f_{it}(\hat{h}_i)$ as input, the corresponding t can still be correctly predicted. In essence, we aim to match the feature-transform relation between the real and generated features so that the generated ones contain the high-quality visual structures that reflected the same feature-transform relation. Specifically , for $i = 0$, we have: $f_{it}(\cdot) = t(\cdot)$, $h_i = x$, $h_{it} = t(x)$, $\hat{h}_i = G(z)$, and $f_{it}(\hat{h}_i) = t(G(z))$. But in practice, unlike the mapping relations between x and $t(x)$, $f_{it}(\cdot)$ $(i \neq 0)$ is unknown to us. As a result, we cannot infer $f_{it}(\hat{h}_i)$ directly from $\hat{h}_i$.

To implement feature-transformation matching method, as shown in Figure 3 (a), for each block E_i $(i \neq 0)$ and G_i $(i \neq 0)$, we introduce a feature-transform net T_i. Before applying FTM, we first train T_i to approximate the unknown mapping function $f_{it}(\cdot)$. Specifically, conditioned on a random t, each T_i takes in a feature h_i as input, and the output is denoted as $T_i(h_i, t)$. The feature-transform net T_i is trained with the loss function:

$$\mathcal{L}_{T_i} = \frac{1}{2} \left\| T_i\left(h_i, t\right) - h_{it} \right\|_2^2. \tag{6}$$

In each iteration, we update T_i to minimize $\mathcal{L}_{T_i}$, then, we could use $T_i\left(\hat{h}_i, t\right)$ to approximate $f_{it}(\hat{h}_i)$. As shown in Figure 3 (b), the estimated transformation can be achieved by:

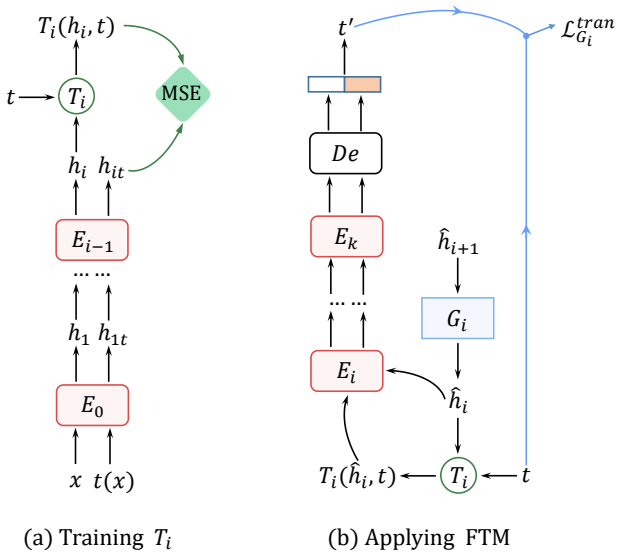
$$t' = De[E_k(E_{k-1}(\dots E_i(\hat{h}_i))), E_k(E_{k-1}(\dots E_i(T_i(\hat{h}_i, t))))]. \tag{7}$$

We set $T_i(h_i, t) = T_i(t(h_i))$, where $t(h_i)$ represents the application of the corresponding transformation on the feature map h_i, and $T_i(\cdot)$ is a learnable mapping function represented by ResNet. Let $p_g(\hat{h}_i)$ represent the distribution of generated feature $\hat{h}_i$, the FTM regularization is applied by training the generator G with the loss function:

$$\mathcal{L}_G^{tran} = \sum_{i=1}^{k} \mathcal{L}_{G_i}^{tran} = \sum_{i=1}^{k} \mathbb{E}_{\hat{h}_i \sim p_g(\hat{h}_i)} \mathbb{E}_{t \sim p(t)} \ell\left(t', t\right). \tag{8}$$

3.5. Framework Summary

For transformation t, since in AET [39], projective transformation has been shown to outperform the affine transformation in training unsupervised models, we choose to train our TrGAN by decoding projective transformations. To ensure a fair comparison, we follow the transformation setting in AET: First, we randomly translate four corners of the input image in both horizontal and vertical directions by ± 0.125 of its height and width, then it is rotated by $0°$, $90°$, $180°$, or $270°$ and randomly scaled by $[0.8, 1.2]$. Each projective transformation t is represented by a parameterized matrix $M(\theta) \in \mathbb{R}^{3\times3}$ between homogeneous coordinates of real images and their transformed counterparts. And the transformation predicting loss is defined as: $\ell\left(t', t\right) = \frac{1}{2} \left\| M(\theta') - M(\theta) \right\|_2^2$.



(a) Training T_i (b) Applying FTM

Figure 3. The training process of feature transformation matching.

In general, our model consists of a global discriminator (D), a generator (G), several feature discriminators (D_i) and feature-transform nets (T_i). During the training process, all the components are trained jointly by iteratively updating each other. The final loss function for the generator G is:

$$\mathcal{L}_G = \mathcal{L}_G^{global} + \lambda_1 \mathcal{L}_G^{adv} + \lambda_2 \mathcal{L}_G^{tran}, \tag{9}$$

where λ_1 and λ_2 are hyper-parameters that control different loss terms.

After the training, our TrGAN can be applied to both unsupervised image generation and representation extracting tasks. To sample generated images from G, all G_i are stacked in a top-down manner, as shown in Figure 2. With noise vector z as input, we can get generated sample $G(z)$ by: $G(z) = G_0\left(G_1\left(\cdots G_k(z)\right)\right)$. We use the global discriminator D for representation extracting. With an arbitrary input image x, the output of each encoder block E_i can be view as the extracted representations.

4. Experiments

In the following, we utilize a variety of datasets including CIFAR-10 [17], ImageNet [22], CELEBA-HQ [15] and LSUN-BEDROOM [37] to comprehensively verify the effectiveness of our proposed method. We test the quality

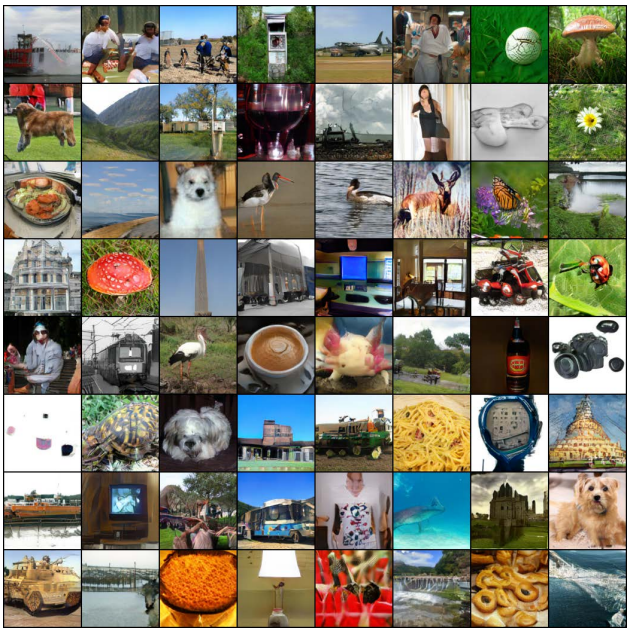


Figure 4. Random samples of unconditionally generated images from the TrGAN on ImageNet.

of extracted representations, and compare the results with other self-supervised models. Under equal training conditions, our TrGAN achieves a better performance over baseline conditional GAN with respect to FID [13].

4.1. Implementation Details

We thoroughly evaluate TrGAN on four datasets: CIFAR-10, ImageNet, CELEBA-HQ, and LSUN-BEDROOM. CIFAR-10 dataset consists of $60k$ 32×32 labeled images. There are $50k$ training images and $10k$ test images in 10 classes. For ImageNet, there are $1.3M$ training images and $50k$ test images in 1000 classes. Since labels are available only for CIFAR-10 and ImageNet, We make a direct comparison between TrGAN and the baseline conditional GAN only on these two datasets. CELEBA-HQ contains $30k$ high-quality human-face images. And the LSUN-BEDROOM contains $3M$ images. To ensure a fair comparison with self-supervised GAN, we pre-process ImageNet and LSUN-BEDROOM images by resizing to the 128×128 resolution. While for CELEBA-HQ, we use the official code provided by authors to achieve images at 128×128.

We choose the Projection conditional GAN [21] (denoted as Cond-GAN) as the baseline model. We choose this model to follow the experimental settings in self-supervised GAN, but also because it is adopted by other best performing GAN [3, 38]. The ResNet architectures of generator and discriminator from Miyato *et al.* [20] are adopted in Cond-GAN and TrGAN. We also use label-conditional batch normalization for the conditional generator in Cond-GAN. And the self-modulated batch normalization is applied in the generator of TrGAN to ensure a similar effect.

TrGAN and the baseline models are trained with a batch size of 64 images before and after transformations. For the adversarial loss $V(G, D)$, we adopt the hinge loss used in Miyato *et al.* [20]. In our model, optimizing T_i needs to train the encoder. We find that the convergence rate of the encoder is faster than that of T_i. During the training, we try both 1 and 2 steps to update the encoder per T_i step, and observe that 1 encoder step per T_i step works well in all experiments. The weights in the model are initialized with Orthogonal Initialization. For all datasets, the Adam optimizer is adopted to train our model, with an initial learning rate of 0.0002. We set $\alpha = 0.2$, $\beta = 1$ for loss $\mathcal{L}_D$ and $\mathcal{L}_G^{global}$, respectively. We also experiment various values of the hyperparameter λ_1 and λ_2, and find that $\lambda_1 = 0.5, \lambda_2 = 1$ work well for all reported experiments. For all other hyperparameters, we use the value in Projection conditional GAN [21] and Miyato *et al.* [20].

4.2. Image Synthesis

In order to quantitatively measure the quality of the generated images from different methods, we adopt the Fréchet Inception Distance (FID) introduced by [13]. FID calculates the Wasserstein-2 distance between the real images and the generated images, lower FID values reveal closer distances between synthetic and real data distributions. Although another approximate measure of sample quality: the Inception score [31] (IS) is also widely used. FID is considered as a more principled and comprehensive metric, and has been shown to be more consistent with human evaluation in assessing the realism and quality of the generated samples [13]. Thus, we use FID as the metrics of sample quality. We train ImageNet for $1M$ iterations. For CIFAR-10, CELEBA-HQ, and LSUN-BEDROOM, we train for $100k$ iterations. We use $3k$ samples for CELEBA-HQ and $10k$ for the other datasets to calculate the FID.

Visual results of generated images on ImageNet are shown in Figure 4. In Table 1, we report the FID of our TrGAN and other baseline models on four datasets. On LSUN-BEDROOM and CELEBA-HQ, our TrGAN outperforms the Self-supervised GAN (SS-GAN) under the same unsupervised settings. On CIFAR-10 and ImageNet, SS-GAN indeed closes the gap between unconditional GAN and its conditional counterpart (Cond-GAN), but Cond-GAN still outperforms SS-GAN. Our TrGAN, on the contrary, achieves better FID rather than Cond-GAN with no requirement of labeled data.

Although there are some other better performing conditional GAN, such as BigGANs [3], that our TrGAN is still far behind from. The additional techniques used in BigGANs (*e.g.*, extremely huge batch-size) require massive computing resources. Our TrGAN, on the other hand, out-

performs the conditional counterpart under the same usual experimental settings. To this end, our result is still significant.

Dataset	Model	FID
CIFAR-10	Cond-GAN	15.53
	SS-GAN [32]	15.65
	TrGAN	**13.41**
ImageNet	Cond-GAN	42.91
	SS-GAN [32]	43.87
	TrGAN	**39.14**
LSUN-BEDROOM	SS-GAN [32]	13.30
	TrGAN	**11.74**
CELEBA-HQ	SS-GAN [32]	24.36
	TrGAN	**23.17**

Table 1. Comparison between TrGAN and other baseline models on four datasets. The best Fréchet Inception Distance (FID) are reported.

Ablation Studies. We comprehensively verify the effectiveness of different components in $\mathcal{L}_G$ by conducting extensive ablation studies. Specifically, we evaluate several variants of our proposed TrGAN on CIFAR-10 dataset. For all models listed below, the same training hyper-parameters are used as the full TrGAN model.

(1) TrGAN: The proposed TrGAN, as described in Section 3.

(2) TrGAN-no-IFM: Same generator and global discriminator as (1), but trained without intermediate feature matching.

(3) TrGAN-no-FTM: Same generator and global discriminator as (1), but trained without feature-transformation matching.

(4) TrGAN-no-(IFM&FTM): Same generator and global discriminator as (1), but trained without intermediate feature matching and feature-transformation matching.

Model	FID
TrGAN-no-(IFM&FTM)	15.67
TrGAN-no-FTM	15.03
TrGAN-no-IFM	14.57
TrGAN	13.41

Table 2. Ablation studies on CIFAR-10 dataset. The best Fréchet Inception Distance (FID) are reported.

We compare the FID of several variants of TrGAN, and the results are reported in Table 2 and Figure 5. As we can see: (1) Simply integrating the transformation predictions into the adversarial training did not bring significant improvement in FID; (2) The performance improvement of TrGAN over SS-GAN and Cond-GAN mainly comes from

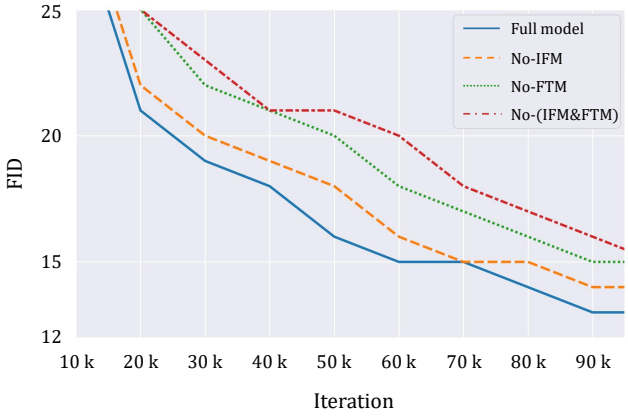


Figure 5. FID curves of TrGAN and its variants on CIFAR-10.

the proposed FTM method; (3) The FID curves on CIFAR-10 have shown the stable training property of our model. In general, IFM as well as our proposed FTM can make the model better utilize the advantages of self-supervised learning, and offering an label-free supervision directly on the feature space of the generator.

4.3. Representation Quality

To evaluate the quality of the representations from TrGAN, we test the representations extracted from each encoder block E_i. For evaluation method, we follow [39] by training a non-linear classifier. Specifically, we train the non-linear classifier with the representations from each encoder block to perform classification task on CIFAR-10 and ImageNet. The number of encoder blocks is determined by the resolution of input images. In detail, the encoder contains four blocks on CIFAR-10 and six blocks on ImageNet.

We choose AET as baseline model for it has demonstrated much more competitive performances in unsupervised representation learning. AET adopts the Network-In-Network and AlexNet as feature extractor on CIFAR-10 and ImageNet, respectively. While for our TrGAN, we choose ResNet architecture for the encoder in consideration of the requirement for adversarial training. For a fair comparison with AET, we adopt AET loss to train a ResNet encoder without the adversarial training process, and denote it AET-only as an alternative baseline model.

	Cond-GAN	AET-only	TrGAN
Block0	81.72	78.16	78.74
Block1	84.59	81.92	82.73
Block2	88.87	87.96	87.12
Block3	91.79	90.08	90.96

Table 3. Comparison between TrGAN, AET-only and Cond-GAN on CIFAR-10. Top-1 classification accuracy are reported.

We test the representation quality from each block on

CIFAR-10, the results are shown in Table 3. The results of Cond-GAN could be viewed as the upper bound of our TrGAN. Except the block2 (E_3), our TrGAN provides better representations than AET-only model across all 3 other blocks. Table 4 further compare the representation quality of TrGAN to state-of-the art unsupervised learning methods on CIFAR-10. AET-only model achieves a slight lower accuracy than original AET model. On the other hand, with the collaborative adversarial training, our TrGAN outperforms both the AET-only model and original AET. These baseline methods are usually based on different architectures and hyperparameters, which makes it difficult for our TrGAN to make a direct comparison with them. But the result is still significant, as the accuracy of TrGAN is close to the upper bound set by the fully-supervised counterpart (Cond-GAN).

Method	Accuracy
Roto-Scat + SVM [26]	82.3
ExamplarCNN [7]	84.3
DCGAN [27]	82.8
Scattering [25]	84.7
RotNet + FC [9]	89.06
AET-project + FC [39]	90.59
Cond-GAN (Upper Bound)	91.79
AET-only	90.08
TrGAN	**90.96**

Table 4. Comparison with other unsupervised representation learning methods by top-1 accuracy on CIFAR-10.

We also compare the representation quality of TrGAN to other state-of-the-art self-supervised learning algorithms on ImageNet. After unsupervised features are extracted, the non-linear classifier is trained on the output from final encoder block (E_6) with labeled samples. As the experimental settings on CIFAR-10, the results of the fully supervised counterpart: Cond-GAN gives upper bounded performance. The results of each block are shown in Table 5.

	Cond-GAN	AET-only	TrGAN
Block0	23.5	24.0	24.7
Block1	30.7	27.8	31.1
Block2	35.6	30.9	33.6
Block3	42.8	37.4	40.9
Block4	50.9	41.6	45.0
Block5	53.1	44.4	49.1

Table 5. Comparison between TrGAN, AET-only and Cond-GAN on ImageNet. Top-1 classification accuracy are reported.

As shown in Table 6, among all the baseline methods, our TrGAN outperforms Context [5], Colorization [40], BiGAN [6] and DeepCluster [4]. There is still a gap between TrGAN and the best performing method: AET-project [39]. We posit that this is mainly due to the limited upper bound set by the fully-supervised counterpart. In summary, our TrGAN achieves the best results on CIFAR-10 as well as competitive results on ImageNet and drastically reduces the gap between unsupervised and supervised learning in terms of both FID and representation quality.

Method	Accuracy
Context [5]	45.6
Colorization [40]	40.7
BiGAN [6]	41.9
DeepCluster [4]	44.0
AET-project [39]	**53.2**
Cond-GAN (Upper Bound)	55.4
AET-only	44.4
TrGAN	49.1

Table 6. Comparison with other unsupervised representation learning methods by top-1 accuracy on ImageNet.

5. Conclusions

In this work, we propose a novel generative model, namely, Transformation Generative Adversarial Network (TrGAN). As a combination of self-supervised learning and GAN, TrGAN could cover the benefits of conditional GAN, such as stable training and visually sharper samples. To better utilize the meaningful features extracted by self-supervised learning, we introduce intermediate feature matching (IFM) methods to further guide the training of internal generator blocks. Also, IFM could provide an additional supervision on the feature space of generator with no need of label information. Besides the requirement for generating transformation-detectable images, we take a further step to to match the feature-transform relation between the real and generated features as well, namely, feature-transformation matching regularization. We then show that this unsupervised generative model can be trained to attain better FID even than its conditional counterpart. The experiments results in terms of both FID and representation quality demonstrate the effectiveness of our method.

Acknowledgement

This work was supported in part to Prof. Houqiang Li by National Natural Science Foundation of China (NSFC) under contract 61836011, and in part to Prof. Wengang Zhou by National Natural Science Foundation of China (NSFC) under contract 61822208 and 61632019, and Youth Innovation Promotion Association CAS 2018497.

References

[1] Pulkit Agrawal, J. Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 3

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: Fine-grained image generation through asymmetric training. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 6

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, 2018. 8

[5] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 3, 8

[6] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017. 2, 8

[7] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 8

[8] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2017. 2

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 8

[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 (NeurIPS)*, 2014. 1

[11] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836, 2016. 1

[12] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[13] 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 (NeurIPS)*, 2017. 6

[14] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, 2011. 1, 2

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 5

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014. 2

[17] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *technical report*, 2009. 5

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[19] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 3

[20] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 6

[21] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. 2, 3, 6

[22] H. Su J. Krause S. Satheesh S. Ma Z. Huang A. Karpathy A. Khosla M. Bernstein A. C. Berg O. Russakovsky, J. Deng and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 5

[23] Augustus Odena. Semi-supervised learning with generative adversarial networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2016. 3

[24] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017. 1

[25] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 8

[26] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*, 2015. 1, 2, 8

[28] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[29] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of International Conference on Machine Learning (ICML)*, 2011. 2

[30] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015. 3

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2, 6

[32] Chen Ting, Zhai Xiaohua, Ritter Marvin, Lucic Mario, and Houlsby Neil. Self-supervised generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 7

[33] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of International Conference on Machine Learning (ICML)*, 2008. 1, 2

[34] Jiayu Wang, Wengang Zhou, Jinhui Tang, Zhongqian Fu, Qi Tian, and Houqiang Li. Unregularized auto-encoder with generative adversarial networks for image generation. In *ACM International Conference on Multimedia (ACM MM)*, 2018. 2

[35] Huang Xun, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4

[36] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision (ECCV)*, 2016. 1

[37] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[38] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 6

[39] Liheng Zhang, Guo Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 5, 7, 8

[40] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 3, 8

[41] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 2