Transformation GAN for Unsupervised Image Synthesis and Representation Learning

Jiayu Wang1 Wengang Zhou1 Guo-Jun Qi2,4 Zhongqian Fu3 Qi Tian1 Houqiang Li1
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
wjl031@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 transferred 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-
development 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.

### 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:

$$
\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p(z)} \left[ \log(1 - D(G(z))) \right].
$$

### 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 $p(t(x), t) = p(t)p(t|x)|t(x))$. 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 $p(t'|t(x)) \approx p(t|t(x))$, 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 $p(t(G(z))|t) \approx p(t(x)|t)$. In this way, given a random transformation $t$, the goal of generator is not only to gen-

![Figure 1. The main framework of the global discriminator $D$.](image-url)
generate 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 \( t_\theta \) can be represented by its own parameters \( \theta \). We use the loss function written as: \( \ell(t', t) = \frac{1}{2} \| \theta - \hat{\theta} \|^2 \). And the global loss functions for \( G \) and \( D \) are listed as follows:

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

\[
\mathcal{L}_G^{\text{global}} = V(G, D) + \beta \mathbb{E}_{x \sim p(x)} \mathbb{E}_{t \sim p(t)} \ell(t', t).
\]

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 a 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 pretrained 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 \( E_n \) into several blocks \( E_i \) where \( i \in \{0, 1, \ldots, k\} \), and \( k + 1 \) is the number of blocks. Each block \( E_i \) acts as a nonlinear mapping function between internal 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 \( i \in \{0, 1, \ldots, k\} \), 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 \( h_{i+1} \) from upper block \( G_{i+1} \) as input, and outputs feature \( h_i \). Note that \( h_0 = G(z) \) 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:

\[
\mathcal{L}_{D_i} = -V(G_i, D_i),
\]

\[
\mathcal{L}_{G_i}^{\text{adv}} = \sum_{i=1}^{k} \mathcal{L}_{G_i}^{\text{adv}} = \sum_{i=1}^{k} V(G_i, D_i),
\]

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 \( h_t \) 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:

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

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

\[
t' = De[E_k(E_{k-1}(\ldots E_1(\hat{h}_i))), E_k(E_{k-1}(\ldots E_1(T_i(h_i, t))))]. \tag{7}
\]

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 \( p_y(\hat{h}_i) \) represent the distribution of generated feature \( \hat{h}_i \). The FTM regularization is applied by training the generator \( G \) with the loss function:

\[
\mathcal{L}_{G}^{\text{trans}} = \sum_{i=1}^{k} \mathcal{L}_{G}^{\text{trans}} = \sum_{i=1}^{k} \mathbb{E}_{h_i \sim p(h_i)} \mathbb{E}_{t \sim p(t)} \sum_{i=1}^{k} \mathcal{L}_{T_i}^{\text{trans}}. \tag{8}
\]

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 \( \pm 0.125 \) of its height and width, then it is rotated by \( 0^\circ, 90^\circ, 180^\circ, \) or \( 270^\circ \) and randomly scaled by \( [0.8, 1.2] \). Each projective transformation \( t \) is represented by a parameterized matrix \( M(\theta) \in \mathbb{R}^{3 \times 3} \) between homogeneous coordinates of real images and their transformed counterparts. And the transformation predicting loss is defined as:

\[
\ell(t', t) = \frac{1}{3} \| M(\theta') - M(\theta) \|^2_2.
\]

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:

\[
\mathcal{L}_{G} = \mathcal{L}_{G}^{\text{global}} + \lambda_1 \mathcal{L}_{G}^{\text{adv}} + \lambda_2 \mathcal{L}_{G}^{\text{trans}}, \tag{9}
\]

where \( \lambda_1 \) and \( \lambda_2 \) 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 sample \( G(z) \) by: \( G(z) = G_0(G_1(\ldots G_k(z))) \). 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
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 $32 \times 32$ 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 $128 \times 128$ resolution. While for CELEBA-HQ, we use the official code provided by authors to achieve images at $128 \times 128$.

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 $L_D$ and $L^\text{global}_G$, respectively. We also experiment various values of the hyperparameters $\lambda_1$ and $\lambda_2$, and find that $\lambda_1 = 0.5$, $\lambda_2 = 1$ work well for all reported experiments. For 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 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>Cond-GAN</td>
<td>15.53</td>
</tr>
<tr>
<td></td>
<td>SS-GAN [32]</td>
<td>15.65</td>
</tr>
<tr>
<td></td>
<td>TrGAN</td>
<td>13.41</td>
</tr>
<tr>
<td>ImageNet</td>
<td>Cond-GAN</td>
<td>42.91</td>
</tr>
<tr>
<td></td>
<td>SS-GAN [32]</td>
<td>43.87</td>
</tr>
<tr>
<td></td>
<td>TrGAN</td>
<td>39.14</td>
</tr>
<tr>
<td></td>
<td>TrGAN</td>
<td>11.74</td>
</tr>
<tr>
<td>CELEBA-HQ</td>
<td>SS-GAN [32]</td>
<td>24.36</td>
</tr>
<tr>
<td></td>
<td>TrGAN</td>
<td>23.17</td>
</tr>
</tbody>
</table>

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 $L_G$ 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrGAN-no-(IFM&amp;FTM)</td>
<td>15.67</td>
</tr>
<tr>
<td>TrGAN-no-FTM</td>
<td>15.03</td>
</tr>
<tr>
<td>TrGAN-no-IFM</td>
<td>14.57</td>
</tr>
<tr>
<td>TrGAN</td>
<td>13.41</td>
</tr>
</tbody>
</table>

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 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.

<table>
<thead>
<tr>
<th>Block</th>
<th>Cond-GAN</th>
<th>AET-only</th>
<th>TrGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block0</td>
<td>81.72</td>
<td>78.16</td>
<td>78.74</td>
</tr>
<tr>
<td>Block1</td>
<td>84.59</td>
<td>81.92</td>
<td>82.73</td>
</tr>
<tr>
<td>Block2</td>
<td>88.87</td>
<td>87.96</td>
<td>87.12</td>
</tr>
<tr>
<td>Block3</td>
<td>91.79</td>
<td>90.08</td>
<td>90.96</td>
</tr>
</tbody>
</table>

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 (E3), 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).

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roto-Scat + SVM [26]</td>
<td>82.3</td>
</tr>
<tr>
<td>ExamplarCNN [7]</td>
<td>84.3</td>
</tr>
<tr>
<td>DCGAN [27]</td>
<td>82.8</td>
</tr>
<tr>
<td>Scattering [25]</td>
<td>84.7</td>
</tr>
<tr>
<td>RotNet + FC [9]</td>
<td>89.06</td>
</tr>
<tr>
<td>AET-project + FC [39]</td>
<td>90.59</td>
</tr>
<tr>
<td>Cond-GAN (Upper Bound)</td>
<td>91.79</td>
</tr>
<tr>
<td>AET-only</td>
<td>90.08</td>
</tr>
<tr>
<td>TrGAN</td>
<td>90.96</td>
</tr>
</tbody>
</table>

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 (E6) 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cond-GAN</td>
<td>23.5</td>
</tr>
<tr>
<td>AET-only</td>
<td>30.7</td>
</tr>
<tr>
<td>TrGAN</td>
<td>35.6</td>
</tr>
<tr>
<td>Block2</td>
<td>39.8</td>
</tr>
<tr>
<td>Block3</td>
<td>42.8</td>
</tr>
<tr>
<td>Block4</td>
<td>50.9</td>
</tr>
<tr>
<td>Block5</td>
<td>53.1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context [5]</td>
<td>45.6</td>
</tr>
<tr>
<td>Colorization [40]</td>
<td>40.7</td>
</tr>
<tr>
<td>BiGAN [6]</td>
<td>41.9</td>
</tr>
<tr>
<td>DeepCluster [4]</td>
<td>44.0</td>
</tr>
<tr>
<td>AET-project [39]</td>
<td>53.2</td>
</tr>
<tr>
<td>Cond-GAN (Upper Bound)</td>
<td>55.4</td>
</tr>
<tr>
<td>AET-only</td>
<td>44.4</td>
</tr>
<tr>
<td>TrGAN</td>
<td>49.1</td>
</tr>
</tbody>
</table>

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 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


