LT-GAN: Self-Supervised GAN with Latent Transformation Detection

Parth Patel[2*†] Nupur Kumari[*1] Mayank Singh[*1] Balaji Krishnamurthy[1]
{parpatel,nupkumar,msingh,kbalaji}@adobe.com
1. Media and Data Science Research Lab, Adobe
2. Birla Institute of Technology & Science, Pilani India

Abstract

Generative Adversarial Networks (GANs) coupled with self-supervised tasks have shown promising results in unconditional and semi-supervised image generation. We propose a self-supervised approach (LT-GAN) to improve the generation quality and diversity of images by estimating the GAN-induced transformation (i.e. transformation induced in the generated images by perturbing the latent space of generator). Specifically, given two pairs of images where each pair comprises of a generated image and its transformed version, the self-supervision task aims to identify whether the latent transformation applied in the given pair is same as that of the other pair. Hence, this auxiliary loss encourages the generator to produce images that are distinguishable by the auxiliary network, which in turn promotes the synthesis of semantically consistent images with respect to latent transformations. We show the efficacy of this pretext task by improving the image generation quality in terms of FID on state-of-the-art models in conditional and unconditional settings on CIFAR-10, CelebA-HQ and ImageNet datasets. Moreover, we empirically show that LT-GAN helps in improving controlled image editing for CelebA-HQ, and ImageNet over baseline models. We experimentally demonstrate that our proposed LT self-supervision task can be effectively combined with other state-of-the-art training techniques for added benefits. Consequently, we show that our approach achieves the new state-of-the-art FID score of 9.8 on conditional CIFAR-10 image generation.

1. Introduction

Generative Adversarial Networks (GANs) have become a popular class of generative models as they have shown impressive capacity in modelling complex data distributions, such as images [2, 23] and audio [9, 10]. GANs consist of a generator and a discriminator network with competing goals: the generator's objective is to generate realistic samples to fool the discriminator and the discriminator's objective is to distinguish between the real samples and the fake ones synthesized by the generator. The generator learns a mapping from a latent distribution to the data distribution via adversarial training with the discriminator. Despite the significant progress of GANs, there lacks enough understanding of how different semantics are encoded in the latent space of generator. It has been observed that in a well trained GAN, semantics encoded in the latent space are disentangled to some extent which makes it possible to perform controlled image editing [20, 37, 42].

Another class of unsupervised learning called self-supervision has demonstrated promising results on a diverse set of computer vision tasks [11, 14, 28, 49, 52]. This training paradigm usually involves designing an auxiliary task with pseudo-labels derived from the structural information of the data for better feature representation learning. Self-supervised learning has also been used in collaboration with adversarial training in GANs [4, 19] to improve training stability and unconditional/semi-supervised image generation quality [27]. Generally, the role of self-supervised methods in GANs is to regularize the discriminator which in turn guides the generator to produce images with more informed geometric or global structure. For example, SS-GAN [4] introduced an auxiliary task of predicting the degree of rotation in the input image to the discriminator during GAN training.

The authors of [51] propose to learn an unsupervised feature representation by encoding the input data transformation rather than data itself [48, 18]. Specifically, in AET [51], image transformation operators are sampled, and the objective is to estimate the transformation given the feature representation of the original and the transformed images. This framework of unsupervised feature learning encourages encoding of the essential visual structure of the transformation so that the transformation can be predicted. AET [51] has shown promising results on various standard visual downstream tasks. Inspired by this approach [51], in the domain of GAN, we propose a binary classification

[*] Authors contributed equally
[†] Work done during Adobe MDSR internship

self-supervised task that aims to detect if the GAN-induced transformation (as described in [51]) applied on two generated images is same. The key idea in this transformation prediction task is to promote useful representation learning as the features would have to encode sufficient information about visual structures of both the original and transformed images for successful training of the auxiliary task.

In this work, we propose a self-supervised task called Latent Transformation (LT) Detection for improving the quality of image synthesis and latent space semantics. Previous methods ([4], [50]) make use of limited predetermined augmentation or transformations (such as rotation, translation) to define the self-supervised loss. However, we utilize GAN-induced transformations (as described in [51]) to define our pretext task. In contrast to earlier works [44, 19] that add a self-supervised loss to the discriminator, our auxiliary task promotes the generator to synthesize images such that the GAN-induced transformations are distinguishable at the feature representation level.

Our main contributions in this paper are the following :

- We propose a novel self-supervised task of distinguishing between GAN-induced transformations to optimize the generator in collaboration with the adversarial training of GANs.
- We demonstrate the efficacy of our proposed LT-GAN approach on several standard datasets and architectures by improving the conditional and unconditional state-of-the-art image generation performance measured using Fréchet Inception Distance (FID) [17] metric.
- We empirically show that our LT-GAN model improves controlled image editing (using existing semantic editing frameworks [42, 37]) over baseline models.

2. Background and Related Work

Self-supervised learning is an unsupervised learning framework that seeks to leverage supervisory signals from the structural information present in the data by defining pretext tasks. Self-supervision techniques have shown a huge potential in diverse research areas, ranging from robotics to computer vision [21, 41, 35, 28, 11]. In visual domain, a pretext task is designed with labels derived from the images themselves that help in learning rich feature representation useful for downstream tasks [14]. Some of the earliest efforts [8] in this area utilize relative position prediction of image patches. Inspired by this task's relation to prediction of context in images, the authors of [30, 31, 33] use a pretext task of predicting the permutation in a image with shuffled patches. [11] used the surrogate objective of predicting the angle of rotation for unlabelled image. The task of in-painting [36] and image colorization [52, 53] have also been used as auxiliary tasks in the self-supervised learning framework. In contrast with utilizing the geometric and structural in-variances, [3] uses the task of predicting the cluster assignment in feature space as pseudo labels for unlabeled data. Also, [32] obtains supervisory signal by counting the visual primitives present in the patches of images. Along the lines of transformation prediction task like [11], AET [51] introduces a surrogate task of reconstruction of input data transformations to learn unsupervised feature representations. Inspired by this work, our approach LT-GAN proposes the auxiliary task of estimating GAN-induced transformations. We hypothesize that it would encourage the generator to synthesize semantically consistent image transformations with respect to similar latent space perturbations.

GANs with self-supervised auxiliary tasks Recently, self-supervised learning has been coupled with adversarial training to improve the training stability and image quality of GANs [4, 19, 45]. The motivation behind adding self-supervised loss to GAN training is to equip the feature representations to recognize the global structures present in real data through the pretext tasks. SS-GAN [4] uses the auxiliary task of image rotation degree classification based on the discriminator features. The authors of [19] propose to use the pretext task of distinguishing between real images and corrupted real images with GAN training. These corrupted images are created by randomly exchanging pairs of patches in an image's convolutional feature map.

Latent Space Manipulation for semantic editing in GANs Conventional approaches of finding interpretable manipulations in GAN latent space compute linear directions corresponding to attribute change by using annotated attributes tags of the images [38, 23]. [47] showed this to be true even for pre-trained classifiers where interpolation in latent feature space of target and source images leads to interpretable transfer of visual properties from a source image to a target image. To assume control over the image generation process in GANs, works [34, 5] propose modifications in architecture and training approach. [34] allows the generation of images belonging to a certain class and therefore requires access to labels for training the model. [5] learns disentangled representations by maximizing the mutual information between a subset of the latent variables and the observation, which enables the process of finding a posteriori semantic direction. However, work by [20, 37, 12] shows that the latent space directions corresponding to transformations (such as zoom, scale, shift, brightness) can be computed using the respective augmentation of images on pre-trained GAN models. These approaches [20, 37] lighten the requirement of attribute tagged images for some general image editing tasks and can also serve as a measure of generalization capacity of generative models. The performance of these latent self-supervised trajectories are limited by biases in the training dataset and the models' generalization performance [7, 20]. Recent advances in GANs [2, 23] in generating photo-realistic images have unlocked the potential for content creation and fine-tuning modifications [46, 1]. [42] performs semantic face editing (for changing attributes such as age, expression, etc.) on a fixed pre-trained GAN model by using linear subspace projection techniques and thus demonstrating disentanglement of the latent space of pre-trained GANs. We show that our approach LT-GAN improves controlled image editing over baseline models by using existing semantic editing frameworks [42, 37].

3. Methodology

In this section, we first present the standard GAN formulation and the terminologies used in the paper. We then introduce our training methodology for LT-GAN that leverages a self-supervised task defined on the latent space of generator to better organize the semantics encoded in the latent space and promote diverse image generation.

3.1. Generative Adversarial Networks

Generative adversarial network (GAN) consists of a generator $G : z \rightarrow x$ and a discriminator $D : x \rightarrow \mathbb{R}$. G learns a mapping from the latent code $z \in \mathbb{R}^d$, sampled from a prior distribution $p(z)$, to an observation $x \in \mathbb{R}^n$ (e.g. a natural image manifold). The role of discriminator D is to differentiate between samples from real data distribution $p(x)$ and the ones generated from G. The standard training of GAN involves minimizing the following loss function in an alternating fashion:

$$\begin{aligned} &L_D : -\mathbb{E}_{x\sim p(x)}[\log(D(x))] - \mathbb{E}_{x\sim p(z)}[1 - \log(D(G(z)))] \\ &L_G : -\mathbb{E}_{z\sim p(z)}[\log(D(G(z)))] \end{aligned} \tag{1}$$

The above loss is commonly known as non-saturating loss and was originally proposed in [13]. A notable modification of the loss for improved training is the hinge loss [43]:

$$\begin{aligned} &L_D : \mathbb{E}_{x\sim p(x)}[1 - D(x)]_+ + \mathbb{E}_{x\sim p(z)}[1 + D(G(z))]_+ \\ &L_G : -\mathbb{E}_{z\sim p(z)}[D(G(z))] \\ &\text{where } [y]_+ = max(0, y) \end{aligned} \tag{2}$$

The latent code z is usually sampled from a normal distribution. For each step of generator update, discriminator is updated for d_{step} times. A common issue with GANs is its instability during training that generally requires stabilization techniques [40, 15, 29]. In this work, we use the widely accepted practice of spectral normalization [29, 2] for stable training.

3.2. Latent Transformation GAN (LT-GAN)

One of the attractive use-cases of GANs which has recently received significant attention is controlled image synthesis by latent space manipulation. Inspired by this, we introduce a self-supervision task on the generator for improved steerability in latent space by leveraging GAN-induced transformations.

GAN-induced Transformation Give a latent code z sampled from a prior distribution $p(z)$ and the corresponding generated image $I = G(z)$, we define GAN-induced transformation as:

$$\mathcal{T}_\epsilon(G(z)) = G(z + \epsilon) \ : \ \epsilon \sim p(\epsilon) \tag{3}$$

For a fixed generator, the transformation $\mathcal{T}$ is parameterized by $\epsilon \in \mathbb{R}^d$, a perturbation of small magnitude, sampled from a distribution $p(\epsilon)$. Applying $\mathcal{T}_\epsilon$ to the image I generated from latent code z generates a transformed version of the image $\mathcal{T}_\epsilon(I)$.

LT Self Supervision In our self-supervision task, we aim to enforce that when a transformation $\mathcal{T}$ parameterized by a particular ϵ is applied to different latent codes, the change in (i.e. difference between) original and transformed images is semantically consistent across all generated images (examples of change can be translation, scaling, etc.). Let $E : x \rightarrow E(x)$ be an encoder network to extract the features of an image. Given a transformation $\mathcal{T}_\epsilon$, the feature representation corresponding to the change between original and transformed images can be written as:

$$f(z, z + \epsilon) = E(G(z)) - E((\mathcal{T}_\epsilon(G(z))) \tag{4}$$

where f captures the change in the original image features and features of its GAN-induced transformation. We choose to implement f simply as the subtraction of encoder features, though other operations like concatenation are valid choices to explore. In LT self supervision, given $f_1 = f(z_1, z_1 + \epsilon_1)$ and $f_2 = f(z_2, z_2 + \epsilon_2)$ where $z_1, z_2 \sim p(z)$, we introduce an auxiliary network A to classify whether the above pair of features $\{f_1, f_2\}$ are corresponding to transformations parameterized by same ϵ or different. Specifically, the self-supervision loss is defined as:

$$\begin{aligned} &L_A = \mathop{\mathbb{E}}_{\substack{z_1,z_2\sim p(z)\\ \epsilon_1,\epsilon_2\sim p(\epsilon)}} L\Big(A\big([f(z_1, z_1 + \epsilon_1), f(z_2, z_2 + \epsilon_2)]\big), y_{ss}\Big) \\ &y_{ss} = (\epsilon_1 == \epsilon_2) \end{aligned} \tag{5}$$

where L is standard binary cross entropy loss and label y_{ss} is 1 if ϵ_1 is equal to ϵ_2 else 0. During training with out self-supervision loss, generator G and the auxiliary network A

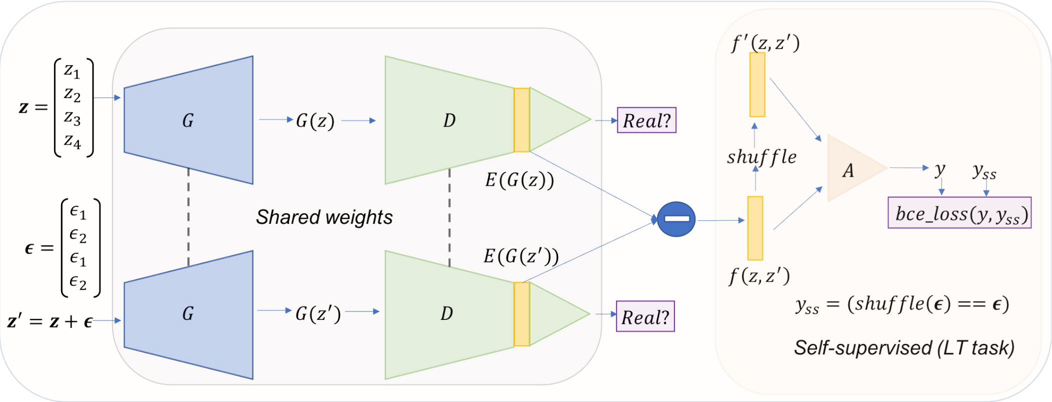

$z = [z_1; z_2; z_3; z_4]$
$\epsilon = [\epsilon_1; \epsilon_2; \epsilon_1; \epsilon_2]$
$z' = z + \epsilon$
G
$G(z)$
D
Real?
$E(G(z))$
Shared weights
$E(G(z'))$
G
$G(z')$
D
Real?
$f'(z, z')$
shuffle
A
y
y_{ss}
$bce\_loss(y, y_{ss})$
$f(z, z')$
$y_{ss} = (shuffle(\epsilon) == \epsilon)$
Self-supervised (LT task)

Figure 1: Overview of our proposed LT-GAN self-supervision task for generator training with an example batch size of $2b = 4$. Generated images $G(z)$ and it's GAN-induced transformations $G(z + \epsilon)$ are used for defining the self-supervision loss (*bce_loss*). Given intermediate discriminator features of above generated images i.e. $E(G(z))$ and $E(G(z + \epsilon))$, the feature representation of the GAN-induced transformation is $f(z, z + \epsilon)$. Auxiliary network A and generator G are trained simultaneously on the pretext task of predicting if $f(z_i, z_i + \epsilon_l)$ and $f(z_j, z_j + \epsilon_k)$ features are generated from same ϵ perturbation in the latent space (where $i, j \in \{1, .., 2b\}$ and $l, k \in \{1, 2\}$).

are updated simultaneously alternating with discriminator updates. Thus, the training objective of LT-GAN is:

$$\begin{aligned} L_G &: -\mathbb{E}_{\substack{z \sim p(z) \\ \epsilon \sim p(\epsilon)}}[D(G(z)) + D(\mathcal{T}_\epsilon(G(z)))] + \lambda.L_A \\ L_D &: \mathbb{E}_{x \sim p(x)}[1 - D(x)]_+ \ + \\ &\quad \mathbb{E}_{\substack{z \sim p(z) \\ \epsilon \sim p(\epsilon)}}([1 + D(G(z))]_+ + [1 + D(\mathcal{T}_\epsilon(G(z)))]_+) \end{aligned} \tag{6}$$

Here, λ denotes the weightage of self-supervision loss in generator. We choose $p(z)$ and $p(\epsilon)$ both to be a normal distribution with standard deviation σ_z and σ_ϵ respectively, where $\sigma_\epsilon < \sigma_z$. The function f in Eq. 4 is implemented as difference of encoded features. $E(G(z))$ features are chosen as the intermediate layer activations of the discriminator. Furthermore, in order to balance the min-max training between the generator and the discriminator, we also train the discriminator to predict fake on GAN-induced transformations. An overview of the generator training in LT-GAN is shown in Fig. 1 and pseudo-code of LT-GAN training is explained in Algorithm 1.

4. Experiments and Results

Datasets We validate our proposed self-supervised task on CIFAR-10 [24], STL-10 [6], CelebA-HQ-128 [22] and ImageNet-2012 [25] datasets. CIFAR-10 consists of 60K 32×32 images, belonging to 10 different classes: 50K images for training and 10K for testing. In STL-10, we use 100K unlabelled images for training (resized to 48 × 48) and 8K images for testing. CelebA-HQ consists of 30K 128×128 face images. We randomly sample 3K images for testing and the rest for training. ImageNet-2012 consists of approximately 1.2 million images which we downsample to 128-128 resolution for our experiments. We use the 50K validation set images of ImageNet for testing.

Algorithm 1 Latent Transformation GAN (LT-GAN)

```
begin
    Input: G, D and A network parameters $\theta_G, \theta_D$ and $\theta_A$. Batch size
    $2b$, weight of self-supervision loss $\lambda$, standard deviation $\sigma_\epsilon$ of normal
    distribution $p(\epsilon)$, discriminator update steps $d_{step}$ for each generator
    update, Adam hyperparemters $\alpha, \beta_1, \beta_2$.
    for number of training iterations do
        for $t : 1...d_{step}$ do
            Sample batch $x \sim p_{data}(x)$
            Sample $\{z^{(i)}, \epsilon^{(i)}\}_{i=1}^{b} \sim p(z), p(\epsilon)$
            $z = \{z^{(i)}\}_{i=1}^{b} \cup \{z^{(i)} + \epsilon^{(i)}\}_{i=1}^{b}$
            $L_D = [1 - D(x)]_+ + [1 + D(G(z))]_+$
            Update $\theta_D \leftarrow Adam(L_D, \alpha, \beta_1, \beta_2)$
        end
        Sample $z = \{z^{(i)}\}_{i=1}^{2b} \sim p(z)$ , $\epsilon_1, \epsilon_2 \sim p(\epsilon)$
        $\epsilon = [\epsilon_1, \epsilon_2].repeat(b)$        ▷ repeat along batch dimension
        Generate images $G(z)$
        Generate GAN-induced transformation $G(z + \epsilon)$
        $f(z, z + \epsilon) = E(G(z)) - E(G(z + \epsilon))$
        $shuffle() =$ permutation($2b$)
        $L_A = L\big(A([f(z, z + \epsilon), f(z, z + \epsilon).shuffle()]), y_{ss}\big)$
        $y_{ss} = (\epsilon == \epsilon.shuffle())$
        $L_G = -D(G(z)) - D(G(z + \epsilon))$
        Update $\theta_A \leftarrow Adam(L_A, \alpha, \beta_1, \beta_2)$
        Update $\theta_G \leftarrow Adam((L_G + \lambda.L_A), \alpha, \beta_1, \beta_2)$
    end
end
```

GAN Architectures and Evaluation We use the GAN architecture of BigGAN [2], StyleGAN [23], SNDC-GAN [29] with their proposed training techniques as the baseline. We also compare against state-of-the-art training technique CR-GAN [50]. In the conditional setting, we perform experiments on CIFAR-10 and ImageNet-2012 with

BigGAN architecture. In the unconditional setting, we perform experiments on CelebA-HQ-128 with StyleGAN and SNDCGAN, on CIFAR-10 with SNDCGAN and STL-10 with ResNet architecture [29].

We use Fréchet Inception Distance (FID) [17] as the primary metric for evaluating image quality and diversity. FID has been shown to be more consistent with human evaluation of image quality and also helps in detecting intra-class mode collapse [17]. We calculate FID between test set images and equal number of generated images for all datasets and report the best FID obtained across 3 runs. We found our methodology to be stable and we show the variance analysis of FID in the supplementary section.

4.1. Training and Implementation Details

The architecture of the auxiliary network A used for the self supervision task consists of a two-layer fully connected network with ReLU activation at the hidden layer and sigmoid activation at the output. Let the features $E(G(z))$ extracted from the discriminator network be of shape $C \times H \times W$. The input layer of A is of $2 \times C \times H \times W$ dimension (since we flatten and concatenate the features $E(G(z))$ and $E(\mathcal{T}_\epsilon(G(z)))$) and the hidden layer is of C dimension. The self-supervised task is introduced after n warmup iterations of training using the standard GAN loss to ensure that the generated images and its transformations are closer to natural image manifold. Furthermore, we only experimented with sampling two distinct ϵ and repeating along the batch dimension for calculating GAN-induced transformation. We leave exploring the effect of varying the above number and relaxing the strict equality between ϵ while calculating self-supervision loss in Eq. 5 for the future work.

Across all model architectures and datasets, we observe the optimal value of σ_ϵ to lie in the range of $[0.4, 0.6]$. For hyper-parameter λ, we found the value of 1.0 to work well except for BigGAN on ImageNet and StyleGAN on CelebA-HQ where we use the value of 0.5. More details about training hyper-parameters for each dataset and architecture are mentioned in the supplementary.

We use Adam optimizer in all our experiments and spectral normalization (SN) [29] in the discriminator (except in the case of StyleGAN). Hinge loss is used by default for training (except in case of StyleGAN, which uses non-saturating loss with R_1 regularization [26]). We follow the default configuration for all architectures and hence we train till 200K generator steps for CIFAR-10 and STL-10, 100K generator steps for CelebA-HQ on SNDCGAN and 525K generator steps on StyleGAN. For ImageNet, we train the model for 250K steps unless the training collapses.

In the following sections, we show that our proposed self-supervision task helps in improving FID scores over the baseline models and can be effectively combined with other regularization techniques in GANs, e.g. CR-GAN [50], across datasets and model architectures. We empirically show that LT-GAN results in a more steerable and disentangled latent space by performing latent space manipulation on CelebA-HQ and ImageNet datasets. We also compare our approach with the recently proposed self-supervised SS-GAN, which uses a rotation-based auxiliary task [4].

DATASET	METHOD	FID
CIFAR-10 (cond.)	BigGAN	14.73
	LT-BigGAN (ours)	11.01
	CR-BigGAN	11.48
	CR+LT-BigGAN (ours)	**9.80**
CIFAR-10 (uncond.)	SNDCGAN	25.39
	LT-SNDCGAN (ours)	22.10
	CR-SNDCGAN	18.72
	CR+LT-SNDCGAN (ours)	**17.56**
CelebA-HQ (uncond.)	SNDCGAN	25.95
	LT-SNDCGAN (ours)	19.63
	CR-SNDCGAN	16.97 (18.44*)
	CR+LT-SNDCGAN (ours)	**16.84**
	StyleGAN	11.43
	LT-StyleGAN (ours)	11.15
ImageNet (cond.)	BigGAN	$10.34^{\#}$
	LT-BigGAN (ours)	$9.94^{\#}$

Table 1: Comparison of self-supervised LT-GAN training approach with state-of-the-art GANs based on FID. * denotes our best reproduced result using the implementation of [1], which is different from the score reported in [50]. # denotes BigGAN Imagenet implementation of [2]

4.2. Results

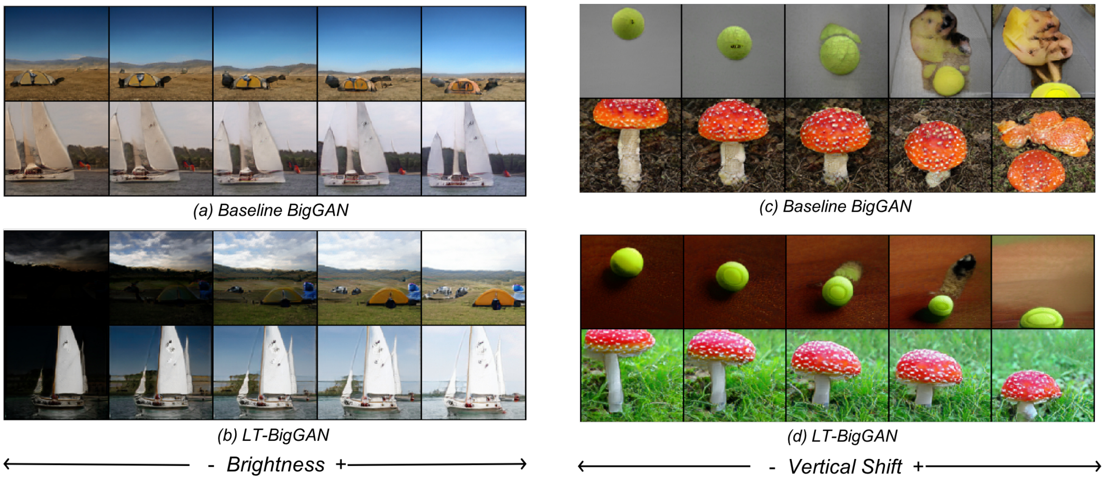
Unconditional GANs In the unconditional setting, we perform experiments on CIFAR-10 with SNDCGAN [29] architecture and CelebA-HQ with SNDCGAN and StyleGAN [23] architectures [1].

From Table 1, we can see that LT-GAN results in improved FID scores compared to the baseline models. Moreover, we also combine our self-supervision task with the current state of the art training methodology CR-GAN [50]. Using CR+LT-GAN results in further improvement of FID score on both datasets.

Conditional GANs In the conditional setting, both the generator and the discriminator are provided with the underlying class label information. We perform experiments on CIFAR-10 and ImageNet datasets [2] using the recent state-of-the-art BigGAN [2] model. We observe that for both

[1] We use the open source implementation of `https://github.com/google/compare_gan` for SNDCGAN and `https://github.com/rosinality/style-based-gan-pytorch` for StyleGAN

[2] Experiments on BigGAN use the implementation of `https://github.com/ajbrock/BigGAN-PyTorch`. For ImageNet, d_{step} is 1 instead of the more optimal setting of 2 as in [2] because of less computation requirements of the former.


(a) Baseline BigGAN

(b) LT-BigGAN

(c) Baseline BigGAN

(d) LT-BigGAN

Figure 2: Qualitative comparison of Brightness (left) and Vertical shift (right) using latent space manipulation on randomly generated images for baseline BigGAN and LT-BigGAN models.

datasets our self-supervision task improves the FID score as shown in Table 1. We also present experimental results of combining our self-supervision technique with the current state-of-the-art CR-GAN [50] on CIFAR-10 that further improves the FID score over CR-GAN.

4.3. Steerability in Latent Space

In this section, we empirically demonstrate that our proposed self-supervision task helps to learn a more steerable latent space. We analyse models trained on CelebA-HQ dataset using the framework of InterfaceGAN [42]. On ImageNet dataset, we use the methodology proposed in [37] to show that BigGAN [2] model trained with our approach helps in finding better edit directions in the latent space corresponding to image transformations like translation, brightness and scale.

ImageNet Dataset We analyze the latent space of generator trained on ImageNet by finding interpretable directions corresponding to parametrizable continuous factors of variation like translation, zoom and brightness using the framework of [37]. To this end, authors in [37] propose a novel reconstruction loss between randomly generated images and the transformed version of these images with varying intensity level (e.g. zoom at various scales) to first determine the latent code corresponding to transformed images. Using this training set of pairs of latent codes of original and transformed images, the direction in latent space corresponding to that particular transformation is learnt as explained in [37]. We use this methodology to discover latent space trajectories corresponding to the image transformations: brightness, scale, horizontal shift and vertical shift, for BigGAN [2] model trained on ImageNet dataset.

Fig. 2 shows the qualitative comparison of brightness and vertical shift direction vectors between baseline BigGAN and LT-BigGAN. We can see in the figure that our approach results in smoother and more meaningful transformations in the image space while preserving the content of the image and avoiding distortions at the extremes. More qualitative comparison on latent space steerability including horizontal shift and zoom is shown in the supplementary.

CelebA-HQ Dataset InterfaceGAN [42] provides a framework to find interpretable semantics encoded in the latent space of face synthesis GAN models. Using it, we discover directions in the latent space to smoothly vary facial attributes, namely age, gender, smile expression and eyeglasses. We use the following procedure as proposed in [42] to discover facial attribute boundaries for StyleGAN and SNDCGAN architectures:

- Randomly generate 500K images. Use a ResNet50 facial attribute detector to predict the value of each binary facial attribute for the generated images. For each binary attribute, sort the list of 500K images based on the predicted value of the attribute and collect top 10K and bottom 10K images. Out of these 20K images, randomly sample 14K images to use as the training set.

- For each attribute, train a linear SVM using the above collected 14K images to predict the value of the attribute (i.e. 0/1) given the latent code used to generate the image. The trained SVM represents a hyperplane that serves as a boundary in the latent space separating the two (-ve/+ve) classes of the binary attribute.

We report the accuracy of each of the trained SVMs on remaining set of images (i.e. 480K images). A higher SVM

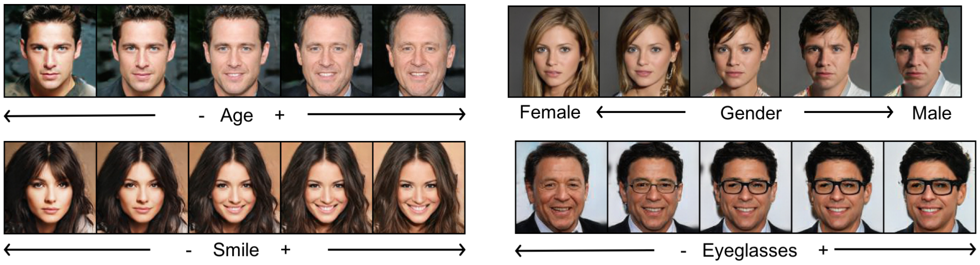


Figure 3: Manipulation of Age (top left), Gender (top right), Smile expression (bottom left)and Eyeglasses (bottom right) attributes by navigating in the latent space of LT-StyleGAN using InterfaceGAN [42] framework. Original images are in the centre and the left and right images are generated by moving the latent code in negative and positive directions respectively.

	SNDCGAN			StyleGAN	
	Baseline	SS-GAN	LT-GAN	Baseline	LT-GAN
A	63.89	67.28	**71.01**	68.66	**70.57**
E	77.36	82.85	**88.53**	70.95	**76.91**
G	64.64	68.34	**72.06**	73.78	**78.49**
S	86.10	85.76	**88.55**	64.75	**65.30**

Table 2: Classification accuracy (%) of separation boundaries in latent space with respect to different attributes of CelebA-HQ. Attributes are A: Age, E: Eyeglasses, G: Gender, and S: Smile expression.

	A	E	G	S
A	1./1.	0.373/**0.326**	0.466/**0.462**	-0.128/**-0.111**
E	-	1./1.	0.292/**0.262**	-0.096/**-0.088**
G	-	-	1./1.	-0.297/**-0.293**
S	-	-	-	1./1.

Table 3: Correlation matrix of synthesized attribute distributions of StyleGAN on CelebA-HQ. In each cell, the first value corresponds to baseline StyleGAN and the second value (following /) corresponds to LT-StyleGAN. Attributes are A: Age, E: Eyeglasses, G: Gender, and S: Smile expression.

accuracy indicates a more steerable latent space. As shown in Table 2, our self-supervision task improves upon the baseline accuracy on all four facial attributes (i.e. age, eyeglasses, gender and expression) for both SNDCGAN and StyleGAN architectures. We also compare LT-GAN against SSGAN [4], which is another self-supervision based GAN. LT-GAN achieves better accuracy on all four attributes compared to SSGAN. Fig. 3 shows some example images of attribute manipulation by moving the latent code in the direction normal to attribute boundary. We show more qualitative samples in the supplementary section.

5. Discussion and Ablation Studies

StyleGAN Latent Space Disentanglement To demonstrate that our proposed self-supervision task helps in achieving a more disentangled latent space, we adopt the InterFaceGAN framework [42] to measure the correlation between synthesized facial attributes distributions of StyleGAN trained on CelebA-HQ dataset. We synthesize 500K images by randomly sampling the latent space. Using a pre-trained ResNet50 facial attribute detector, we assign attribute scores to all 500K images for all four facial attributes (age, eyeglasses, gender, and smile). Treating each attribute score as a random variable, we can compute the correlation between two attributes using their distribution observed over the 500K generated images. The formula to compute correlation between two attributes X and Y is $\rho_{XY} = \dfrac{Cov(X,Y)}{\sigma_X \sigma_Y}$, where $Cov(\cdot,\cdot)$ denotes covariance and σ denotes standard deviation. Correlation values closer to zero indicate a more disentangled latent space. Table 3 shows the correlation values between attributes for both baseline StyleGAN and LT-StyleGAN. It can be observed that the correlation between attributes is more closer to 0 for LT-StyleGAN as compared to baseline StyleGAN.

Similar to [23], we also compute the perceptual path length for both latent spaces Z and W of StyleGAN. The idea is that a more disentangled latent space will result in perceptually smoother transitions in the image space as we interpolate in the latent space, and thus give lower perceptual path length. For the Z space, perceptual path length is 242.33 for baseline StyleGAN and 133.11 for LT-StyleGAN. For the W space, perceptual path length is 77.48 for baseline StyleGAN and 72.71 for LT-StyleGAN.

Choice of hyper-parameters σ_ϵ and λ. Hyper-parameter σ_ϵ controls the difficulty of self-supervision task. A large value of σ_ϵ makes the self-supervision task trivial (since it is easier to distinguish between latent space perturbations that are far apart). In contrast, a smaller value of σ_ϵ makes the pretext task too difficult and may cripple training. Hyper-parameter λ controls the ratio of weight assigned to self-supervision loss and adversarial loss in generator objective function. To study the effect of these hyper-parameters on model performance (i.e. FID), we perform ablation experiments by varying one hyper-parameter and fixing the other to its optimal value. We conduct this experiment on SNDCGAN architecture with CelebA-HQ dataset and the results are as shown in Fig. 4. It can be observed that minimum

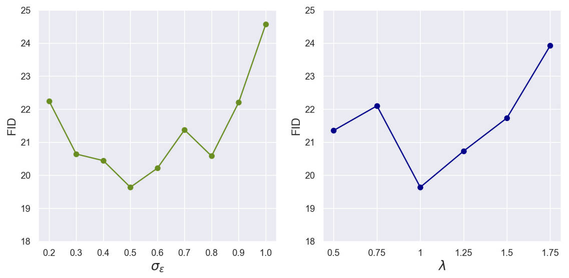


Figure 4: FID on varying σ_ϵ and λ for LT-SNDCGAN on CelebA-HQ

Methods	CelebA-HQ	CIFAR-10	STL-10
Baseline	25.95	25.39	35.74
SS-GAN	26.85	22.88	33.63
LT-GAN (ours)	**19.63**	**22.10**	**31.35**

Table 4: FID comparison of LT-GAN with SS-GAN on different datasets

FID is achieved at the optimal values of $\sigma_\epsilon = 0.5$ and $\lambda = 1.0$. FID increases as we move away from the optimal values and the graphs show a U-shaped trend.

Auxiliary network accuracy on generative transformations We validate the efficacy of our learned auxiliary network A in SNDCGAN CelebA-HQ setting with $\sigma_\epsilon = 0.5$. We vary the σ_ϵ of $p(\epsilon)$ and test the ability of the auxiliary network to distinguish between GAN-induced transformations. In Fig. 5, we report the binary classification accuracy on randomly generated 25K samples and their transformations from the trained generator. It can be observed that the auxiliary network classifies relatively well for transformations with σ_ϵ in neighbourhood of 0.5, on which it was trained, but performance decreases as σ_ϵ diverges from 0.5.

Comparison with SS-GAN We also compare LT-GAN with SS-GAN [4], which is a recently proposed technique to train GANs with rotation-based self-supervision. In contrast to SS-GAN, our self-supervision task is only defined wrt the generator and considers generative transformations instead of rotation transformations. We compare across 3 datasets: CIFAR-10 and CelebA-HQ on SNDCGAN architecture and STL-10 on ResNet architecture [29]. The results are shown in Table 4. We observe that LT-GAN performs better on CelebA-HQ and STL-10 and is comparable to SS-GAN on CIFAR-10. Since rotation transformation is less informative for datasets with single domain images like faces, SS-GAN performs worse than baseline on CelebA-HQ dataset. However, in comparison to SS-GAN, LT-GAN improves the FID score for all datasets.

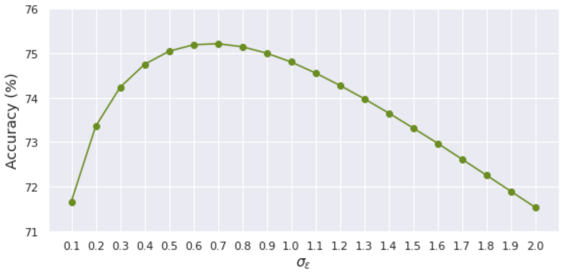


Figure 5: Accuracy (%) of our binary classification self-supervision task on varying σ_ϵ for LT-SNDCGAN on CelebA-HQ

Classification Accuracy Score (CAS) CAS [39] was recently proposed as an additional metric for evaluating conditional generative models on the downstream task of image classification. A standard image classification network is trained using images generated from the model as a training set. The trained model is used to predict labels on the test set of real images and the obtained test accuracy is the CAS metric (higher the better). It was shown that neither FID [17] nor IS[40] scores are predictive of CAS, and thus it serves as another independent evaluation metric. We compare the CAS score of LT-BigGAN and baseline BigGAN trained on Imagenet and CIFAR-10 datasets. For ImageNet dataset, we trained a ResNet-50 [16] classifier, similar to [39], using the generated samples and evaluated its performance on its validation set. LT-GAN achieves the top-5 accuracy of 49.24 as compared to 44.15 of the baseline model, outperforming it by over 5%. Furthermore, on closer examination of class level classification accuracy, we found that many classes in the baseline model suffer from severe mode collapse, which is alleviated to a large extent in LT-GAN. Sample images from those classes for both baseline and LT-GAN are shown in supplementary section. On CIFAR-10 dataset, a ResNet-56 model (as used in [39]) trained via samples generated from LT-BigGAN achieved a test accuracy of 79.93%, whereas the baseline model achieved 70.57%.

6. Conclusion

In this work, we present LT-GAN, a novel self-supervised technique for improving the image generation quality and diversity of GANs. The pretext task of distinguishing GAN-induced transformation helps the generator blocks of GANs to learn steerable latent feature representation and synthesise high-fidelity images. The experimental results demonstrate that when combined with strong GAN baselines [2, 23], our model LT-GAN improves the quality and diversity of generated images on several standard datasets. The performance on FID metric and controlled image editing highlights the effectiveness of LT-GAN in both unconditional and class-conditional GAN settings. We hope that this approach of leveraging latent transformation as a pretext task can be extended to other generative models.

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *ICLR*, 2019.

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

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[4] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *CVPR*, 2019.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *arXiv preprint arXiv:1606.03657*, 2016.

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[7] Emily Denton, Ben Hutchinson, Margaret Mitchell, and Timnit Gebru. Detecting bias with generative counterfactual face attribute augmentation. In *CVPR Workshop on Fairness Accountability Transparency and Ethics in Computer Vision*, 2019.

[8] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[9] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[10] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[12] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019.

[13] 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.

[14] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

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

[18] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *In Advances in Neural Information Processing Systems*, 1994.

[19] Rui Huang, Wenju Xu, Teng-Yok Lee, Anoop Cherian, Ye Wang, and Tim K. Marks. Fx-gan: Self-supervised gan learning via feature exchange. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3183–3191, 2020.

[20] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *ICLR*, 2020.

[21] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. Grasp2vec: Learning object representations from self-supervised grasping. In *CoRL*, 2018.

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

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[26] Andreas Geiger Lars Mescheder and Sebastian Nowozin. Which training methods for gans do actually converge? *CoRR*, abs/1801.04406, 2018.

[27] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *ICML*, 2019.

[28] Puneet Mangla, Mayank Singh, Abhishek Sinha, Nupur Kumari, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, 2020.

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018a.

[30] T. Nathan Mundhenk, Daniel Ho, and Barry Y. Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018.

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[32] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[33] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.

[34] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 207.

[35] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. *ECCV*, 2018.

[36] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[37] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020.

[38] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[39] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *NeurIPS*, 2019.

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[41] Nawid Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*, pages 228–243. Springer, 2018.

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

[43] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 7:3, 2017.

[44] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, and Ngai-Man Cheung. An improved self-supervised gan via adversarial training. In *arXiv preprint arXiv:1905.05469*, 2019.

[45] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Linxiao Yang, and Ngai-Man Cheung. Self-supervised gan: Analysis and improvement with multi-class minimax game. In *NeurIPS*, December 2019.

[46] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

[47] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *CVPR*, 2017.

[48] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[49] Donglai Wei1, Joseph Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[50] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *ICLR*, 2020.

[51] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by autoencoding transformations rather than data. In *CVPR*, 2019.

[52] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, 2016.

[53] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.