Pseudo Conditional Regularization for Inverse Mapping of GANs

SANG-HEON SHIM and JAE-PIL HEO, (Member, IEEE)

1Department of Artificial Intelligence, Sungkyunkwan University, Suwon 16419, South Korea
2Department of Computer Science and Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jae-Pil Heo (jaepilheo@skku.edu)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea Government Ministry of Science and ICT (MSIT) (Reconstruction of Non-Line-of-Sight Scene for VR/AR Contents) under Grant 2020-0-00973, and in part by the Artificial Intelligence Graduate School Support Program (Sungkyunkwan University) under Grant 2019-0-00421.

ABSTRACT Inverse mapping of the Generative Adversarial Networks (GANs) which projects data to latent space have been recently introduced, and it is shown that the inverse mapping models trained by the bidirectional adversarial learning can enable novel and practical operations including interpolation between real data. However, existing techniques still do not ensure the consistent mapping between the data and their latent representation so that the models are hardly converged throughout training steps. Our discussion begins with empirical investigations on the inconsistency issue of the prior techniques, and we further propose a novel adversarial learning method, Pseudo Conditional Bidirectional GAN (PC-BiGAN), for training the inverse mapping of GANs with a high degree of consistency and similarity-awareness. Our models are specifically guided by the pseudo conditions defined by the proximity relationship among data in unsupervised learned feature space. We demonstrate that our novel bidirectional adversarial learning frameworks improve the performance in sample reconstruction, generation, and interpolation.

INDEX TERMS Deep learning, generative adversarial networks, inverse mapping of GANs, latent representation of GANs.

I. INTRODUCTION

Generative Adversarial Networks (GANs) [1] have been recognized as one of the most powerful frameworks to learn the data generating distributions. In GAN frameworks, a generative model, generator, is trained to imitate real data by learning a mapping from a latent distribution to the data space. On the other hand, an adversarial model, discriminator, is concurrently trained to distinguish between generated and real samples. Training two adversarial models is formulated as a two-player minimax game and leads the generator to nicely approximate the real data distribution when the discriminator becomes difficult to distinguish between true and fake samples.

The ability of GANs is sufficiently validated in both theoretical and practical aspects. For example, GANs trained on image datasets have provided magnificent achievements [2]–[6]. In particular, interpolations and arithmetic among latent variables [7]–[9] produce smooth semantic variations and attribute changes via the generator, respectively. Since such impressive results confirm that the learned latent space reflects the semantic information of the data [10]–[13].

Recently an interesting question has been arisen and investigated: can inverse mapping of GANs to project data back to the latent space be learned and utilized? The Bidirectional GANs (BiGAN) [14] and Adversarially Learned Inference (ALI) [15] have been proposed to learn an additional model, encoder, that maps samples in the real data distribution to the latent space. The discriminator is then trained to distinguish between two joint samples; 1) a real data and its latent representation predicted by the encoder and 2) a fake data and its latent variable fed to the generator. On the other hand, the goal of both the generator and the encoder is fooling the discriminator. These bidirectional approaches are well formulated to match two joint distributions of the generator and encoder.

Nonetheless, Li et al. [16] have demonstrated a scenario where the ALI and BiGAN produce inconsistent inverse mappings across the training steps, and present a solution, ALICE (ALI with Conditional Entropy), based on the conditional entropy expressed by the cycle-consistency [17]–[19]. The ALICE relieved the non-identifiability problem of the
ALI and improved the sample reconstruction performance, however, there is lack of consideration on the similarity-preserving mapping. For instance, if two samples are considered as similar (or close) in the data space, then their corresponding latent vectors are also preferred to be located in close proximity in the latent space. However, this similarity-based criteria is not explicitly considered in BiGAN, ALI, and ALICE.

In this paper, we propose a novel bidirectional learning method, Pseudo Conditional BiGAN (PC-BiGAN), that aims to produce latent representations preserving the similarity among data by guiding the encoder to form a similarity-aware latent space. We first compute initial features of data based on an unsupervised model. Although our method does not have any restriction on choice of a feature extraction model, in this paper we consistently utilize a trained encoder of the BiGAN to clearly validate the benefits of our approach over the BiGAN. Once we have the initial representations of the data, we perform a clustering to define pseudo classes and assign a pseudo label to each data. Those pseudo labels coarsely guide the encoder to map samples within the same pseudo class to latent vectors in close proximity. As a learning strategy, we simultaneously train two models, PC-BiGAN and its inference model PC-BiGAN-Inf, where the generator and encoder of a PC-BiGAN are explicitly regularized by the pseudo labels whereas ones of PC-BiGAN-Inf are not. However, the discriminator of PC-BiGAN-Inf is frequently synchronized with the PC-BiGAN. Since the generator and encoder are trained under the judgment of discriminator, we can learn the generator and encoder of PC-BiGAN-Inf to have similar joint distributions with ones of the PC-BiGAN without explicit guidance by pseudo conditions through the synchronization strategy.

We extensively evaluated our proposed method, especially PC-BiGAN-Inf, in sample reconstruction, generation, and interpolation tasks. The experimental results validate that our method achieves higher performance compared to the prior techniques specialized for the inverse mapping of GANs.

This paper is organized as follows. In Sec. II, we first summarize the literature related to inverse mapping of GANs and conditional GANs. In Sec. III, we further explain details on the bidirectional GANs, and discuss about their inconsistent mapping across training steps. We then present our methods alleviating the aforementioned issue; similarity-preserving mapping in bidirectional GANs frameworks. Experimental comparisons against prior bidirectional GANs frameworks are reported in Sec. V. We finally conclude our paper in Sec. VI.

II. RELATED WORKS

A. INVERSE MAPPING OF GANs

Inverse mapping from data space to latent space of GANs have been explored in recent years [14]–[16], [20]–[23]. Notable examples are BiGAN [14] and ALI [15]. They added an additional parametric model to GAN frameworks that infers latent representations of data samples. Li et al. [16] proposed to regularize the ALI by cycle-consistency [17] to improve the sample reconstruction performance. Our proposed method also aims to regularize bidirectional GANs, however, we guide models to be aware of a global structure of latent space induced by the data similarity. The merits of our global guidance scheme over sample-by-sample approach [16] are clearly validated in Sec. V.

One alternative approach is to compute latent representation based on the back-propagation [20]–[23]. Those methods first generate a fake sample from a random latent code and then update the latent code by the back-propagation according to predefined loss functions. Creswell and Bharath [20] utilized a pixel-level reconstruction loss for the back-propagation. On the other hand, Yeh et al. [22] used both weighted pixel-wise difference and adversarial losses. Although those methods have certain strength that they can be applied to any pretrained GANs, the predicted latent representations are sensitive to low-level features since their objectives mostly concentrate on the pixel-wise reconstruction. Furthermore, they require a huge computational time due to iterative updates based on the back-propagation which is inevitable to produce a latent code for a single data.

B. CONDITIONAL GANs

Conditional GANs (cGANs) [4], [24]–[26] have shown successful results in class-specific data generation. Mirza and Osindero [26] first proposed cGANs framework and validated that the learned generator can produce samples according to given conditional labels. The cGANs framework is further applied to various applications [27]–[31] such as image generation from text [32]–[34] and image-to-image translation [35]. Note that, all the aforementioned methods are based on supervised labels or supervised feature representations. At a high-level, our proposed method can be seen as a conditional BiGAN framework since we regularize the BiGAN by conditional labels. However, the main difference between our technique and the ordinary cGANs is that we define and utilize pseudo labels defined by totally unsupervised manner based on the data similarity.

III. BACKGROUND AND MOTIVATION

A. BACKGROUND

Let us define notations that we will use throughout this paper. We use a distribution of data $x \in \Omega_X$ as $P_X(x)$. The goal of GAN [1] is to approximate $P_X(x)$. The GAN framework is consist of two parametric models: a generator $G$ and a discriminator $D$. The generator maps a latent distribution $P_Z(z)$ for $z \in \Omega_Z$ into $P_X(x)$. The discriminator distinguishes whether a particular $x$ is drawn from $P_X(x)$ or $P_Z(z)$.

The BiGAN [14] has an additional parametric model, encoder $E$, for the inverse mapping from the data $x$ to the latent representation $E(x)$. The discriminator $D$ then predicts whether a given sample $x$ is drawn from real joint distribution $P_{XE}(x, z)$ or fake joint distribution $P_{GZ}(x, z)$. The objective of
BiGAN is described as follows: training steps.

These graphs report $I(t, k)$ (Eq. 4) measuring how many neighboring latent codes are preserved between two consecutive training epochs when $k=100$ and $k=1000$. The results confirm that the learned latent representations of ALI, ALICE, and BiGAN are inconsistent across training steps.

BiGAN is described as follows:

$$L_{\text{BiGAN}} = \min_{G, E} \max_D V(D, E, G),$$

where

$$V(D, E, G) := E_{x \sim P_{D}} [\mathbb{E}_{z \sim P_{z}(\cdot | x)}[\log D(x, z)]] + \mathbb{E}_{z \sim P_{z}} [\mathbb{E}_{x \sim P_{x}(\cdot | z)}[\log (1 - D(x, z))]]$$

It is theoretically verified that the perfect inverse relationship between the generator and the encoder (i.e. $G^{-1} = E$) can be reached with the optimal discriminator $D^*$. The BiGAN, however, hardly provides the sample reconstruction $G(E(x)) = x$ in practice.

Note that, the Adversarially Learned Inference (ALI) [15] is an identical model with BiGAN. In this paper we explain our models and experimental results on top of BiGAN for the sake of succinct explanations.

**B. MOTIVATION**

The BiGAN is hardly converged to a perfect inverse mapping mainly because the latent representation $E(x)$ is significantly changing during training steps. Since those latent representation $E(x)$ are fed to the discriminator $D$ (Eq. 2), such inconsistency disturbs training of $D$ that distinguishes between real and fake samples. It sequentially makes the learning the encoder $E$ difficult because the encoder is trained based on the judgement of $D$. As a result, the inconsistent mapping degrades the inverse mapping accuracy of BiGAN.

We experimentally validate the inconsistency mappings during training of the BiGAN. In the experiment, we investigate the latent space by measuring the change of the neighborhood codes. Given a set of real data samples $X = \{x_1, \ldots, x_n\}$, we compute their latent representations $L^{t} = \{E^{t}(x_1), \ldots, E^{t}(x_n)\}$ predicted by the trained encoder $E^{t}(\cdot)$ at the $t^{th}$ training epoch. For each $E^{t}(x_i)$, we then investigate $k$-nearest neighbors among the latent codes by the following measure:

$$N(i, k, t) = \{j | R(E^{t}(x_i), E^{t}(x_j), L^{t}, d(\cdot, \cdot)) \leq k\},$$

where $R(y_i, y_j, Y, d(\cdot, \cdot))$ is a function that computes a ranking of $y_j$ within a set $Y$ according to the distance measure $d(\cdot, \cdot)$ in non-decreasing order. In this experiment, $L_2$ distance $\| y_i - y_j \|$ is used. Thus, $N(i, k, t)$ is a set of $k$-nearest neighbor indices of $E^{t}(x_i)$ among all the latent codes predicted by the encoder at the $t^{th}$ training epoch. We specifically measure how many local neighboring latent codes are preserved in two consecutive training epochs $t-1$ and $t$ to investigate the consistency of learned latent representations during the training steps as follows:

$$I(t, k) = \frac{1}{n} \sum_{i=1}^{n} |N(i, k, t-1) \cap N(i, k, t)|,$$

where $| \cdot |$ is the cardinality operator (i.e. the number of elements in a set). For instance, $I(3, 10) = 0.6$ indicates that 60% of 10 neighboring samples in the learned latent space at the second training epoch still remain as 10-nearest samples at the third epoch on average. One expected trend of the $I(t, k)$ across training epochs may be that it starts with a low value while setting up the global structure of latent space and is getting increased as the models converge. However, the experimental results show that the neighborhood in the learned latent space is hardly preserved across the training epochs (Fig. 1). Specifically, the intersection rates of BiGAN, ALI, and ALICE were fluctuating and the values were below 0.4. These results imply that the latent distribution induced by the encoder $E(x)$ is changed drastically and is hardly converged even for the local neighborhood. Note that we conducted the experiment on the CIFAR-10 benchmark.

In this paper, we specifically aim to alleviate the aforementioned inconsistency problem of the prior techniques for the inverse mapping of GANs. As shown on Fig. 1, our proposed method predicts more consistent latent representations during adversarial learning, and we further experimentally validate that it leads to improve the inverse mapping performance over the baseline methods in Sec. V.

**IV. OUR APPROACHES**

We propose to regularize our encoder to locate latent vectors of similar data in close proximity in learned latent space. In order to train an encoder under the regularization, we propose a joint regularization scheme for both the generator and encoder. Note that, all of our learning strategies utilize fully-supervised pseudo labels to guide a global structure of the latent space.

Specifically, the similarity among data is defined by initial feature representations of data. Even though our method does not have any restriction on the initial feature representations, we explain our method when it utilizes the latent vectors predicted by the encoder of an ordinary BiGAN.
Let us denote a latent vector (or arbitrary feature representations) of a sample $x$ extracted by a trained encoder of the BiGAN (or any unsupervised feature extractor) as $f(x)$. We construct pseudo classes according to the similarity defined with the features $f(x)$. The $k$-means clustering [36] is, especially, performed to define the pseudo classes.

### A. PSEUDO CONDITIONAL BiGAN (PC-BiGAN)

One essential condition for the optimal inverse mapping $E = G^{-1}$ is that the encoder and generator share an unified latent space. To achieve this, we consider a way to jointly regularize the latent space of generator as well as encoder in an explicit manner. To provide a guidance with explicit and unified ways, we introduce a regularization method based on conditional GAN architecture. We specifically define a fixed global structure of the latent space based on the pseudo classes and guide both encoder and generator to learn under the predefined space (see Fig. 2).

We compute a single pseudo label $y_i$ of each data $x_i$ and $k$ similarity groups $S_1, \ldots, S_k$ based on pseudo labels as follows:

$$ y_i = \arg\min_j d(f(x_i), c_j) \quad \text{and} \quad S_j = \{x_i | y_i = j\}, \quad (5) $$

where $c_j$ is the centroid of $j$-th cluster computed by the $k$-means clustering on the initial features $f(x)$, and $d(A, B)$ denotes the Euclidean distance between $A$ and $B$. For each similarity group $S_j$, we compute the mean $\mu_j$ and covariance matrix $\Sigma_j$. Those mean and covariance explicitly describe the regions corresponding to similarity groups in the latent space. Specifically, we guide the encoder to map a real data $x_i$ to a latent vector near $\mu_j$ and the generator map a latent variable $z$ drawn from a multivariate Gaussian distribution $\mathcal{N}(\mu_j, \Sigma_j)$ to a sample similar to data in $S_j$. Note that, the $k$-means clustering is performed only once as a pre-computation to define pseudo labels of training samples in our training stage.

We also define two functions $h_i(x)$ and $h(z)$ that compute the pseudo labels of a data $x$ and latent representation $z$, respectively.

$$ h_i(x) = \arg\min_j d(f(x), c_j) \quad (6) $$

$$ h(z) = \arg\min_j d(z, c_j), \quad (7) $$

### 1) PSEUDO CONDITIONAL BiGAN

Once we determine a fixed global structure of the latent space based on $(\mu_i, \Sigma_i)$, we train our encoder to map a similarity group $S_i$ to latent variables belong to a single fixed region bounded by $(\mu_i, \Sigma_i)$. To do this, we modify the BiGAN framework by incorporating the conditional GAN [26] so that all models to get a pseudo label as an additional input (see Fig. 3). Specifically, the generator, encoder, and discriminator receive joint inputs $(\mu_i, z), (x, h(x)), \text{and} (x, z, h)$. $h$ is pseudo label index. Note that, the $z$ fed to the generator is a random variable drawn from $\mathcal{N}(\mu_i, \Sigma_i)$ for all $1 \leq i \leq k$, and the input fed to the discriminator is either of a real tuple $(x, E(x), h(x))$ or a fake tuple $(G(z), z, h(z))$. The adversarial learning with those joint inputs trains the encoder and generator to produce their outputs strongly related to the predefined global structure based on the pseudo labels likewise conditional GANs in supervised settings. Consequently, we formulate our adversarial loss as follows:

$$ L_{adv}^{PC-BiGAN} = \min_{G_p, E_p} \max_{D_p} V(D_p, E_p, G_p), \quad (8) $$

where $V(D_p, E_p, G_p) := \mathbb{E}_{x \sim P_x}[\mathbb{E}_{z \sim P_{z\text{G}}(x, h(x))}[\log D_p(x, z, h(x))]]$

$$ + \mathbb{E}_{z \sim P_z}[\mathbb{E}_{x \sim P_{xG}(z, h(z))}[\log(1 - D_p(x, z, h(z)))]$$

and models $D_p, G_p$ and $E_p$ denote discriminator, generator and encoder of our pseudo conditional BiGAN (PC-BiGAN), respectively. Note that $z$ denote a randomly sampled vector from a multivariate Gaussian distribution $\mathcal{N}(\mu_z, \Sigma_z)$.

In order to further regularize the encoder to map a sample $x \in S_{h(x)}$ to a latent vector nearby $\mu_{h(x)}$, we add the Euclidean distance between $E(x)$ and $\mu_{h(x)}$ as a distance-based loss $L_{dist}$ as follows:

$$ L_{dist} = \mathbb{E}_{x \sim P_x}[\| E_p(x) - \mu_{h(x)} \|^2 ] \quad (9) $$
As a result, our objective function to train the pseudo conditional BiGAN is given as follows:

$$L_{\text{PC-BiGAN}} = L_{\text{PC-BiGAN}}^{adv} + \lambda_{\text{dist}} L_{\text{dist}},$$

(10)

where $\lambda_{\text{dist}}$ is a small constant to balance the distance-based loss with adversarial loss. In all the experiments, we consistently use $\lambda_{\text{dist}} = 0.01$.

2) BINARY PSEUDO LABEL

One straightforward way to provide the pseudo labels to models as their input is encoding the labels as one-hot vector [24], [37], [38]. However, it significantly increases the number of parameters as we define more similarity groups. In practice, we set the number of similarity groups $K$ as 500 for all experiments. In order to reduce the model complexity, we convert each pseudo label index into a binary string with a fixed length and feed to models. For instance, an one-hot vector corresponding to a data within the similarity group index 3, $[0 \ 0 \ 1 \ 0 \ldots \ 0]^T \in \{0, 1\}^{500}$, is converted to 9-dimensional binary vector $[00 \ldots 011]^T \in \{0, 1\}^9$. We found that converting pseudo labels to binary representations and feeding them to the models significantly reduce the training time without any loss in accuracy.

B. INFEERENCE MODEL WITHOUT PSEUDO LABELS (PC-BiGAN-INF)

Although our pseudo conditional BiGAN can learn inverse mapping in more consistent manner compared to the BiGAN, it has certain limitation that it requires explicit pseudo labels $h_i(z)$ and $h_j(x)$ when generating samples (i.e. $G_P(z, h_i(z))$) or mapping a real data back to the latent space (i.e. $E_P(x, h_j(x))$). For instance, when we encode $x$ to a latent vector we need to compute a pseudo label corresponding to $x$ by extracting the initial feature $f(x)$ and specify the similarity group where $x$ belongs to. The pseudo label assignment in the inference stages has non-negligible computational costs since it basically requires to compute distances from $f(x)$ to all the cluster centers $\{c_i\}_{i=1}^N$ (Eq. 5). In order to remove such overhead in the inference stages, we propose an alternative learning strategy to train inference models which does not require any explicit pseudo labels.

The generator $G_I(x)$ and encoder $E_I(x)$ of our inference models, PC-BiGAN-Inf, have almost identical architectures with $G_P(z, h_i(z))$ and $E_P(x, h_j(x))$ except the inference models do not receive the pseudo labels as their input. On the other hands, the discriminator of PC-BiGAN-Inf, $D_I(x, z, h)$, has the same structure with $D_P(x, z, h)$. Since only $G_I$ and $E_I$ are utilized in the inference time, we allow $D_I$ to have explicit pseudo labels as one of its input.

Specifically, we simultaneously train the PC-BiGAN and PC-BiGAN-Inf. If the adversarial learning based on Eq. 8 enables to form regularized latent space with $G_P$, $E_P$, and $D_P$, then it seems to be also possible to train $G_I$ and $E_I$ by an adversarial learning with $D_P$ to have a similar latent space with $G_P$ and $E_P$. In other words, $D_P$ becomes in charge of discriminating tuples $(x, z, h)$ predicted by four models $G_P$, $E_P$, $G_I$, and $E_I$. However, this strategy has a problem. From a $D_P$’s point of view, $D_P(x, z, h)$ should classify tuples predicted by $E_I$ without pseudo conditions, $(x, E_I(x), h_I(x))$, to a real joint sample. Since $E_I$ does not receive any conditional labels, it hardly produces latent representations $E_I(x) \sim \mathcal{N}(\mu_{h_I(x)}, \Sigma_{h_I(x)})$ due to lack of sufficient guidance that $E_P(x, h_J(x))$ relies on. Such poorly predicted samples of $E_I$ having drastically different distribution from the output of $E_P$
obliviously disturb \(D_P\) to be converged. Similarly, fake tuples estimated by \(G_I\) confuse \(D_P\) to learn the decision boundaries between real and fake samples.

Instead of playing an adversarial game with 5 models \(G_P, E_P, G_I, E_I\) and the judge \(D_P\), we allow \(D_I\) to join the game. The role of \(D_I\) is to discriminate samples produced by \(G_I\) and \(E_I\), whereas \(D_P\) is a referee for \(G_P\) and \(E_P\). However, learning \(D_I\) only based on the inference models has the aforementioned problem. In order to reduce the problem, we frequently synchronize \(D_I\) with \(D_P\) by copying the learned parameters of \(D_P\) to \(D_I\). This way prevents to disturb \(D_P\) to learn a regularized latent space by separating it from \(G_I\) and \(E_I\). Moreover, \(D_I\) which is a copy of \(D_P\) provides appropriate guidance to \(G_I\) and \(E_I\). In practice, we synchronize two discriminators in every 10 training epochs.

As a result, our minimax game to train PC-BiGAN-Inf is governed by the following objective:

\[
\mathcal{L}_{\text{adv}}^{\text{PC-BiGAN-Inf}} = \min_{D_I} \max_{E_I, G_I} V(D_I, E_I, G_I),
\]

where 

\[
V(D_I, E_I, G_I) := \mathbb{E}_{x \sim P_X} [\mathbb{E}_{z \sim P_Z(x)} [\log D_I(x, z, h_z(x))]] + \mathbb{E}_{z \sim P_Z} [\mathbb{E}_{x \sim P_X(z)} [\log(1 - D_I(x, z, h_z(z)))]
\]

We also apply the same regularization loss \(\mathcal{L}_{\text{dist}}\) with equal amount \(\lambda_{\text{dist}}\) to the \(E_I\). Then, our final objective for PC-BiGAN-Inf is given as follows,

\[
\mathcal{L}_{\text{PC-BiGAN-Inf}} = \mathcal{L}_{\text{adv}}^{\text{PC-BiGAN-Inf}} + \lambda_{\text{dist}} \mathcal{L}_{\text{dist}}
\]

**Balanced Mini-Batch:** If we feed the real data \(x \sim P_X\) and latent variables \(z \sim P_Z\) in totally random manner, the number of training samples with respect to the pseudo class indices become imbalanced. For instance, it is highly likely to have no pair of \(x\) and \(z\) where \(h_z(x) = h_z(z)\) in a mini-batch. We found that such totally random sampling approach leads the training often failed. To overcome this problem, we first prepare real data \(\{x_1, \ldots, x_m\}\) randomly when the mini-batch size is \(2m\). We then draw latent variables \(\{z_1, \ldots, z_m\}\) so that the mini-batch is pseudo class-wise balanced based on the following criteria:

\[
z_i \sim \mathcal{N}(\mu_{h_z(x)}), \Sigma_{h_z(x)}).
\]

Recall the experimental study reported Fig. 1, our proposed PC-BiGAN-Inf better preserves the local neighborhood compared to the prior techniques across training epochs, thanks to our method that regularizes based on predefined and fixed latent space defined by the data similarity.

**V. EVALUATION**

In this section, we intensively evaluate our proposed method against the state-of-the-art bidirectional adversarial learning techniques including BiGAN, ALI, and ALICE.

Note that, the reported quantities with a \(\dagger\) are taken from the literature. In order to conduct experiments with baselines whose results are unavailable in the literature yet, we have utilized the source codes released by the authors and the hyper-parameters suggested in each paper for fair and reproducible experiments.

**A. DATASETS AND IMPLEMENTATION DETAILS**

1) **DATASETS**

We conduct experiments on CIFAR-10 and CIFAR-100 benchmarks [39] to validate the benefits of our proposed method.

[39] https://www.cs.toronto.edu/kriz/cifar.html
method. The CIFAR-10 dataset contains images of 10 object classes, while the CIFAR-100 dataset has instances of 20 coarse and 100 fine-grained classes. Both benchmarks contain 60,000 images at a resolution of $32 \times 32$, and the images are separated into 50,000 and 10,000 images for training and testing sets, respectively. We utilize the training set without supervised labels when training methods and measure all the quantitative and qualitative performances on the testing set.

2) IMPLEMENTATION DETAILS
In all experiments, we set the dimensionality of the latent space to 200. We set the number of similarity groups $k$ as 500 for all experiments. For every 10 epochs, we synchronize the learning parameters of discriminators. We use the ADAM optimizer [40] with learning rate of 0.0002 and momentum $\beta_1 = 0.9$. We set the mini-batch size to 256 and trained our models during 500 epochs. As data augmentation, we scale data to $[-1, 1]$ and perform random horizontal flipping. Although there is no restriction on choice of neural network architecture, we use DCGAN [2] structure for fair comparisons. We also feed latent vectors or binary pseudo labels to all input and intermediate layers of generators, discriminators, and encoders as did in the BiGAN.

B. RECONSTRUCTION AND GENERATION
We first evaluate the performance of our encoder and generator by the data reconstruction and generation tasks. For those experiments, we use the CIFAR-10 dataset.

1) SAMPLE RECONSTRUCTION
The sample reconstruction based on the sequence, $x \to E(x) \to G(E(x))$, is an appropriate way to evaluate the performances of both the encoder and generator together. We report the pixel-level reconstruction error measured by Mean Squared Error (MSE) between the original data $x$ and its reconstruction $G(E(x))$. As reported in Table 1, our method achieves significantly lower reconstruction error over the baseline techniques. We also report the Structural Similarity (SSIM) scores [41], and the results show that our method improves the second best models ALI and BiGAN in a large margin. Those improvements over the prior techniques confirm the merits of our technique that provides a guidance of a global structure of the latent space. Moreover, our method even outperformed ALICE which is optimized for the sample reconstruction by incorporating the cycle-consistency loss to their objective. These results validate that having structural consistency of the latent space is more beneficial than samples-wise one-to-one constraints.

We also report qualitative comparison of reconstructed images in Fig. 4. We observe that our method provides more accurate sample reconstruction over the state-of-the-art techniques in terms of colors and global shapes.

2) SEMANTIC RECONSTRUCTION
We also have investigated the semantic consistency of the reconstructed samples. In order to quantitatively measure the performance of semantic reconstruction, we specifically utilize a mis-classification rate of reconstructed samples, denoted by Semantic Reconstruction Error (SRE):

$$SRE(x) = \begin{cases} 0 & \text{when } f_{e}(G(E(x))) = Y^{C}(x) \\ 1 & \text{when } f_{e}(G(E(x))) \neq Y^{C}(x), \end{cases} \quad (14)$$

where $f_{e}$ is an external classification model trained with supervision [42] which is only utilized for the evaluation, and $Y^{C}(x)$ is the true label of the sample $x$. As reported in Table 1, our method significantly outperforms over the tested baselines. Those results confirm that our similarity based guidance boosts the latent space to better reflect the semantics, and strongly imply that our methods can capture the semantic classes by the latent codes even in the unsupervised settings. While our guidance globally regularizes the latent space to locate similar data in close proximity, the adversarial objective allows local structures of the latent space to exploit the semantic juice. On the other hand, the ALICE performs even worse than its base model ALI since the cycle-consistency expressed by the pixel-wise $L_2$ distances tends to more focus on learning low-level features rather than high-level ones. Note that, the classification accuracy of the external classifier $f_{e}$ utilized in this experiment on the real samples is 90%.

### TABLE 1. Quantitative comparison of reconstruction and image generation tasks on CIFAR-10 test set. The scores with $\dagger$ are taken from the literature.

| Method | Reconstruction | Image Quality |
|--------|---------------|---------------|
|        | MSE (↓) | SSIM (↑) | SRE (↓) | ICP (↑) | FID (↓) |
| ALI    | 0.67↓$\dagger$ | 0.11 | 0.62 | 5.93↑$\dagger$ | 61.5 |
| ALICE  | 0.41↑$\dagger$ | 0.09 | 0.66 | 6.01↑$\dagger$ | 55.7 |
| BiGAN  | 0.31 | 0.11 | 0.61 | 7.35 | 24.4 |
| PC-BiGAN-Inf | 0.22 | 0.15 | 0.50 | 7.79 | 23.1 |

### TABLE 2. Quantitative comparison of image generation performance between PC-BiGAN and PC-BiGAN-Inf on CIFAR-10 test set. Note that, we feed random pseudo labels to PC-BiGAN to generate samples.

| Method     | ICP (↑) | FID (↓) |
|------------|---------|---------|
| PC-BiGAN-Inf | 7.79   | 23.1 |
| PC-BiGAN   | 5.01   | 19.9 |

### TABLE 3. Quantitative comparison of reconstruction performance on CIFAR-100 test set. Note that, the coarse labels of CIFAR-100 are utilized to measure SRE.

| Method | MSE (↓) | SRE (↓) |
|--------|---------|---------|
| BiGAN  | 0.53   | 0.82   |
| PC-BiGAN-Inf | 0.23 | 0.72 |

Note that, the classification accuracy of the external classifier $f_{e}$ utilized in this experiment on the real samples is 90%.
3) SAMPLE GENERATION

We further have studied the sample generation performance by measuring the Inception Score (ICP) [43] and Frechet Inception Distance (FID) [44] to evaluate both quality and diversity of the generated samples. We specifically generate 50K and compute ICP and FID on those generated samples. As shown in Table 1, our method provides the highest ICP and lowest FID among the tested methods.

3) SAMPLE GENERATION

We further have studied the sample generation performance by measuring the Inception Score (ICP) [43] and Frechet Inception Distance (FID) [44] to evaluate both quality and diversity of the generated samples. We specifically generate 50K and compute ICP and FID on those generated samples. As shown in Table 1, our method provides the highest ICP and lowest FID among the tested methods.

We further claim that our conditional generator \( G_F \) produces higher quality of samples, since we more explicitly regularize our PC-BiGAN to generate samples belonging to their corresponding similarity groups compared to its inference model PC-BiGAN-Inf. In order to validate this claim, we report ICP and FID scores of PC-BiGAN which requires explicit computation of the pseudo labels and PC-BiGAN-Inf without any condition in Table 2. The result shows that our PC-BiGAN further improves image generation quality. It confirms that our regularization based on the data similarity in a more explicit way provides higher performance. Note that, the pseudo labels to measure image quality are randomly selected for a fair comparison between our PC-BiGAN and PC-BiGAN-Inf.

C. INTERPOLATION AND VECTOR ARITHMETIC

We also validate that the latent space produced by our method captures semantic variations. We specifically report quantitative and qualitative results of the sample interpolation and vector arithmetic operations among latent vectors.

1) SAMPLE INTERPOLATION

One may suspect that the guidance for global structures of latent space disturbs to exploit semantic juice of GAN. To show that our similarity based regularized latent space preserves the attributes of semantic juice, we estimate mean softmax values across interpolated samples. Specifically, we first select two real samples \( x_1 \) and \( x_2 \) from different categorical classes. Those samples are encoded to the latent representation \( E(x_1) \) and \( E(x_2) \), then decoded to the sample \( G(z_t) \) with an interpolated latent vector parameterized by \( t \)

\[
z_t = (1 - t)E(x_1) + tE(x_2)
\]

for \( t \in [0, 1] \). We measure mean softmax values of their own categorical classes across interpolation steps \( t \) on the top of external classifier. As shown in Fig. 5, the mean softmax values of our reconstructed samples \( t = 0 \) for \( x_1 \) and \( t = 1 \) for \( x_2 \) are higher than BiGAN. Although those values are output of softmax activation, it shows that the classifier predicts category of those reconstructed samples with higher confidence than the baseline method. On the other hands, we also found that the derivatives of our softmax values are dramatically changed from \( t = 0.3 \) to \( t = 0.5 \). It validates that our regularized latent space is semantically well-clustered since the semantic changes are focused on the middle of interpolation (i.e. boundary between different clusters). Note that, the mean softmax values of real samples are 0.87.

Fig. 6 shows the qualitative results of the sample interpolation. Specifically, we interpolate between two real samples which have different semantic attributes. Although we predefine the global structure of the latent space, those results confirm that our model captures underlying semantic attributes such as viewpoint and shape. Furthermore, the semantic...
We first examine the generated data \( G \) ing \[45\]. We conduct experiments with \emph{car} and \emph{truck} classes on real data as did in natural language processing. One interesting property of the inverse mappings of GANs is that it enables to perform attribute-level arithmetic operations among latent vectors well thanks to our novel the bi-directional adversarial learning framework regularized by the structure by structure guidance.

D. Reconstruction on CIFAR-100

In order to test the scalability of our method in terms of the complexity of data distribution, we evaluate MSE and SRE of our method against BiGAN on CIFAR-100 benchmark. As reported in Table. 3, our method significantly outperforms BiGAN in both pixel and semantic-level reconstructions. The performance improvement over BiGAN is consistent with respect to the complexity of datasets. Note that, we use 20 coarse labels to measure SRE. For the external classifier, we utilized ResNet-18 model \[46\] which achieves classification accuracy of 0.83 on real samples.

The qualitative results of the sample reconstruction on CIFAR-100 are provided in Fig. 8. We found that our method robustly reconstructs the samples with various shapes and classes.

E. Ablation Studies

In this section, we discuss about our method with various ablation studies. We specifically analyze the impacts of initial features, frequency of weight synchronization between \( D_P \) and \( D_I \), regularization methods, and incorporation of cycle-consistency.

1) Initial Feature Extractor

We first perform analysis on the impact of initial feature extractor utilized to compute pseudo labels. We test our method with the bottleneck representations of Auto Encoder (AE) as the initial features instead of latent vectors of BiGAN, since AE is one of the most popular unsupervised feature extractors \[47\]–\[50\]. As reported in Table. 4, our method w/ AE clearly outperforms BiGAN across all the tested reconstruction tasks. Although AE generally more focuses on the local features such as colors or textures, our method w/ AE more accurately reconstructs semantic features and provides a lower SRE compared to BiGAN. This ablation study strongly implies that our learning strategy has a low sensitivity on choice of initial feature extractors unless they can capture the data similarities. Note that, the experiments are conducted on CIFAR-10 test data.

2) Weight Synchronization Frequency

Table. 5 shows SRE scores of our method with various frequencies for weight synchronization between two discriminators \( D_P \) and \( D_I \). We observed that our method with the

\[
\varepsilon^k = \frac{1}{3}(E(x^k_1) + E(x^k_2) + E(x^k_3)),
\]

for each \( k \in \{a, b, c\} \). Note that, \( E(\cdot) \) is the latent representation of the real data predicted by the tested techniques including ours PC-BiGAN-Inf.

While all tested methods generated quite reasonable sample in terms of the attributes and classes, our PC-BiGAN-Inf produced the clearest image. Note that, the term \texttt{neutral} represents that many attributes are mixed together so it is hard to pick a single description \[2\].

We further study on the arithmetic operations among latent vectors. We specifically compute a latent vector \( \varepsilon^d \) as following equation:

\[
\varepsilon^d = \varepsilon^a - \varepsilon^b + \varepsilon^c,
\]

where \( \varepsilon^a, \varepsilon^b, \) and \( \varepsilon^c \) are computed by Eq. 15. The last rows (4th rows) of Fig. 7(a) and 7(b) show the generated samples from \( \varepsilon^d \) by ALI, ALICE, BiGAN, and our PC-BiGAN-Inf methods. As reported, our method produces the most accurate samples in terms of attribute, class, and appearance. Especially in Fig. 7(c), ALI and ALICE produced trucks, and BiGAN generated left-heading car, while the result of our PC-BiGAN-Inf correctly reflects the desired attribute and class.

Those experimental results validate that our inverse mapping and latent space support the vector arithmetic operators among latent vectors well.
frequencies from 10 to 25 epochs consistently reached at 0.50 of SRE score. Furthermore, the frequencies shorter or longer than the best setting such as 5 or 50 epochs did not cause noticeable performance degradation, and our method achieved higher accuracy than BiGAN regardless of the frequency (0.52 at 5 epochs vs 0.61 of BiGAN). Those results confirm that our method has a low sensitivity to the weight synchronization frequency. Note that, we train the models up to 500 epochs, and the frequency of 10 epochs is consistently utilized in all the other experiments.

### 3) REGULARIZATION METHODS

Our proposed method regularizes the BiGAN by providing pseudo labels to both generator and encoder (see Sec. IV). One may find that an alternative approach could be modifying only the encoder to produce a structured latent space. Specifically, we can stack a linear classification layer on the top of the encoder to predict the pseudo label. This way can implicitly form a clustered latent space according to the pseudo labels. We now name this alternative approach as PC-BiGAN-Cls.

We conduct experiments to compare our original method and the alternative one PC-BiGAN-Cls, and the results are reported in Table. 6. Interestingly, PC-BiGAN-Cls outperforms the BiGAN in both SRE and MSE scores. This result is notable since classifying the pseudo labels by the encoder does not provide direct guidance to the generator to have a matched latent distribution with the encoder but marginally alleviates the inconsistent mappings of latent codes across training steps. As a result, guiding the BiGAN framework to have consistent mappings during training helps to improve the performance of the inverse mappings. Meanwhile, our proposed method outperforms the PC-BiGAN-Cls and BiGAN in both measures MSE and SRE by a large margin. Those results confirm the unique merits of our method that guides both the encoder and generator to produce a shared global structure of the latent space. Note that, all tested models have exactly same neural network architecture in the inference phase.

#### 4) CYCLE-CONSISTENCY

Our method regularizes the BiGAN by the guidance of the global structure of the latent space based on the pseudo conditions defined by the data similarities, while the ALICE utilizes the cycle-consistency expressed by the sample-wise reconstruction. Although our method outperforms the ALICE even in the sample reconstruction task (Table. 1), we investigate our method when incorporating the cycle-consistency in addition to our objectives. To do this, we add the following cycle-consistency loss $L_{cycle}$ to our objective with different conditions defined by the data similarities, while the ALICE utilizes the cycle-consistency expressed by the sample-wise reconstruction. Although our method outperforms the ALICE even in the sample reconstruction task (Table. 1), we investigate our method when incorporating the cycle-consistency in addition to our objectives. To do this, we add the following cycle-consistency loss $L_{cycle}$ to our objective with different

$$
L_{cycle}(G_P, E_P) = \lambda_{cycle} \left( \| G_P(E_P(x), h_z(x)) - x \|_1 + \| E_P(G_P(z), h_z(z)) - h_z(z) \|_1 \right)
$$

Similarly, the following cycle-consistency loss is added to the objective of PC-BiGAN-Inf:

$$
L_{cycle}(G_t, E_I) = \lambda_{cycle} \left( \| G_t(E_I(x)) - x \|_1 + \| E_I(G_t(z)) - z \|_1 \right)
$$

Table. 7 shows the MSE and SRE scores when we add the cycle-consistency loss to our objective with different $\lambda_{cycle}$. As reported, the cycle-consistency is not beneficial to our method. Moreover, our models are even hardly converged or generate blurry or semantically meaningless images if we increase the $\lambda_{cycle}$ larger than $10^{-3}$. This is mainly because our regularization scheme based on the sample-wise guidance conflicts with the sample-wise way. More importantly, we would like to highlight that our original method without any explicit objective on the sample reconstruction significantly outperforms the ALICE as shown in Table. 1.

### VI. CONCLUSION

In this paper, we have proposed a novel pseudo conditional bidirectional adversarial learning framework to train inverse
mapping of GANs so that the resulting latent space can reflect the similarity among data and exploit the semantic juice.

While the prior techniques hardly provide consistent mapping between the data and their latent representations, we guide the models to have predefined the global structure of the latent space formed by the proximity relationship among data in unsupervised learned feature space. We specifically define pseudo labels based on clustering on such feature space of an unsupervised model. We train our pseudo conditional bidirectional GAN (PC-BiGAN) with the pseudo labels so that the encoder maps data to certain region corresponding to the pseudo label in the latent space. Since PC-BiGAN requires pseudo labels as its input, we further propose to learn its inference model that does not require any explicit feed of pseudo labels. We extensively evaluate our method against the state-of-the-art bidirectional learning approaches, and the experimental results validate that our method consistently outperforms all the tested techniques in various tasks including sample reconstruction, generation, and interpolation. The experimental results confirm that the unique merits of our work to have a fixed guidance based on the sample similarity for the latent space in the bidirectional adversarial learning.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–16.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[4] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–35. [Online]. Available: https://openreview.net/forum?id=B1xqsL09Fm

[5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 214–223.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 7354–7363.

[7] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and O. Mastropietro, and A. Courville, "Adversarially learned inference," in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 9047–9057.

[8] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5485–5493.

[9] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 597–613.

[10] T. Miyato and M. Koyama, "cGANs with projection discriminator," in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[11] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4099–4108.

[12] Y. Yan, J. Xu, B. Ni, W. Zhang, and X. Yang, "Skeleton-aided articulated motion generation," in Proc. 25th ACM Int. Conf. Multimdia, 2017, pp. 199–207.

[13] C. Chen, G. Lample, N. Usunier, A. Bordes, L. Denoyer, and O. Mastropietro, and A. Courville, "Adversarially learned inference," in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 9047–9057.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–16.

[16] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[17] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–35. [Online]. Available: https://openreview.net/forum?id=B1xqsL09Fm

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 214–223.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 7354–7363.

[20] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and O. Mastropietro, and A. Courville, "Adversarially learned inference," in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 9047–9057.

[21] N. Hadad, L. Wolf, and M. Shahar, "A two-step disentanglement method," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 772–780.

[22] Z. Zheng and L. Sun, "Disentangling latent space for VAE by label relevant/relevant dimensions," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12154–12163.

[23] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised GANs via auxiliary rotation loss," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 12154–12163.
[38] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, “Invertible conditional GANs for image editing,” in Proc. NIPS Workshop Adversarial Training, 2016, pp. 1–9.

[39] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

[40] D. P. Kingma and J. Lei Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., Dec. 2014, pp. 1–15.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” 2014, arXiv:1412.6806. [Online]. Available: http://arxiv.org/abs/1412.6806

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, “Improved techniques for training GANs,” in Advances in Neural Information Processing Systems, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, 2016, pp. 2234–2242.

[44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 6626–6637. [Online]. Available: http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf

[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 3111–3119.

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[47] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, “Understanding and improving interpolation in autoencoders via an adversarial regularizer,” in Proc. 7th Int. Conf. Learn. Represent. (ICLR), New Orleans, LA, USA, May 2019, pp. 1–19.

[48] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J. Mach. Learn. Res., vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[49] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in Proc. 2nd Int. Conf. Learn. Represent. (ICLR), Banff, AB, Canada, Apr. 2014, pp. 1–14.

[50] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” in Proc. 31st Int. Conf. Mach. Learn., 2014, pp. 1278–1286.

SANG-HEON SHIM received the B.S. degree in computer engineering from Sejong University, Seoul, South Korea, in 2017, and the M.S. degree in electrical and computer engineering from Sungkyunkwan University (SKKU), Suwon, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in artificial intelligence. His research interests include generative adversarial networks, machine learning, computer vision, and deep learning.

JAE-PIL HEO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 2008, 2010, and 2015, respectively. He was a Researcher with the Electronics and Telecommunications Research Institute (ETRI). He is currently an Assistant Professor with Sungkyunkwan University (SKKU), South Korea. His research interests include scalable algorithms for high-dimensional data, computer vision, machine learning, and deep learning.