GAN-based Semi-supervised Learning On Fewer Labeled Samples

Takumi Kobayashi
takumi.kobayashi@aist.go.jp

National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan

Abstract

Semi-supervised learning is recently addressed by means of neural networks in the framework of deep learning. For the semi-supervised tasks where training samples are partially labeled, the generative adversarial networks (GANs) are applicable not only to augmentation of the training samples but also to the end-to-end learning of classifiers exploiting the unlabeled samples. It, however, is found that the previous GAN-based semi-supervised method is less effective on the smaller number of labeled samples, and thus in this paper, we propose a novel GAN-based method to effectively work on fewer labeled samples. In the GAN framework, through analyzing gradients of the discriminator which are fundamental to learn the network via back-propagation, we formulate a discriminator model and accordingly a generator loss to cope with the less discriminative classifier trained on the fewer labeled samples. The proposed model is also mixed with the original one to further improve discriminativity on the semi-supervised learning in an efficient way beyond the simple linear combination. The experimental results on semi-supervised classification tasks using MNIST, SVHN and CIFAR-10 datasets show that the proposed method exhibits favorable performance compared to the other methods.

1 Introduction

Building classifiers requires labeled training samples in the learning process, and thus it is effective to formulate the learning scheme in the semi-supervised framework utilizing both unlabeled samples and (the smaller number of) labeled ones, which reduces the human labor for annotating samples. The semi-supervised learning problem has so far been addressed mainly by exploiting the relationships among samples in a graph where pair-wise similarities are assigned to the edges [1, 2, 23, 25]. In recent years, it is tackled by means of neural networks in the deep learning framework, such as by improving the network architecture [9, 18] and/or the objective loss on which the network is learned [11, 14], without explicitly considering the relationships among the labeled and unlabeled samples.

In that framework, the generative adversarial networks (GANs) [5, 17] contribute to learning the classifier in a semi-supervised way, while they remarkably advance the field of image generation [6, 26]. The GANs can generate realistic images which are regarded as part of training data [20, 27] to effectively augment data for improving classification performance. While the methods [20, 27] leverage the (fake) images generated by separately pre-trained GANs to the data augmentation, there also exist the methods [16, 19, 21] that directly couple GANs with classifiers in an end-to-end manner utilizing the unlabeled samples.

Though the methods are effective without carefully controlling the labels of the generated images by GANs, we empirically found that they compromise performance in the case that labels are assigned only to the smaller number of samples. Such a case of the fewer labeled samples is practically desirable for significantly reducing human labor in label annotation, though leading to the less discriminative classifier due to lack of labels.

In this paper, we propose a method to further improve the performance of the GAN-based semi-supervised learning by coping with the less discriminative classifier especially on the smaller number of labeled samples. Through analyzing how the previous GAN-based method works on the semi-supervised learning from the viewpoint of *gradients*, the fundamentals to update network parameters, we present a novel discriminator model and accordingly a loss for training the generator to boost performance of the less discriminative classifier. Then, to bridge the gap between the proposed method and the previous one, we mix those two methods to further improve discriminativity in an efficient way beyond the simple linear combination which actually results in a performance drop.

2 GAN-based Semi-Supervised Learning

We first briefly review the GAN-based methods [16, 19] that we focus on, and then analyze how the network parameters are updated by those methods with discussing why it fails to work on the fewer number of labeled samples.

For classifying an image I to one of C classes, the *classifier* $\phi_{\boldsymbol{\theta}}$ is built by the neural network of L layers whose parameters are denoted by $\boldsymbol{\theta}$. It produces the posterior probabilities of an image I by means of softmax as

$$\mathrm{p}(c|I;\boldsymbol{\theta}) = \frac{\exp(x_c)}{\sum_{c'=1}^{C}\exp(x_{c'})}, \quad \forall c \in \{1,\cdots,C\}, \tag{1}$$

where x_c indicates the c-th neuron activation at the last layer to form the vector $\boldsymbol{x} \triangleq \phi^L_{\boldsymbol{\theta}}(I) \in \mathbb{R}^C$ as shown in Fig. 1. On the other hand, given a random vector $\boldsymbol{z} \in \mathbb{R}^d$, the *generator* G produces a *fake* image $\tilde{I} = \mathsf{G}_{\boldsymbol{\eta}}(\boldsymbol{z})$ with the parameters $\boldsymbol{\eta}$ in the same manner as the standard GANs to generate images [5, 17]. In the semi-supervised learning framework, the *discriminator* D is constructed by leveraging the classifier $\phi_{\boldsymbol{\theta}}$ to the discrimination between *real* and *fake* images as

$$\mathsf{D}_{\boldsymbol{\theta}}(I) = \frac{\sum_c^C \exp(x_c)}{1+\sum_c^C \exp(x_c)} = \mathrm{p}(\mathit{real}|I;\mathsf{D}_{\boldsymbol{\theta}}), \tag{2}$$

and accordingly $\mathrm{p}(\mathit{fake}|I;\mathsf{D}_{\boldsymbol{\theta}}) = 1-\mathsf{D}_{\boldsymbol{\theta}}(I)$. This is based on the $C+1$-class softmax where the *fake* class is added to the C classes of the classification targets and represented by "1" in the denominator without loss of generality [19]. From the above formulation, we can derive the following loss to be minimized w.r.t $\boldsymbol{\theta}$;

$$\min_{\boldsymbol{\theta}} E_{I\in\mathcal{L}}[-\log\{\mathrm{p}(c_I|I;\boldsymbol{\theta})\}] + E_{I\in\mathcal{U}}[-\log\{\mathsf{D}_{\boldsymbol{\theta}}(I)\}] + E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{1-\mathsf{D}_{\boldsymbol{\theta}}(\tilde{I})\}], \tag{3}$$

where c_I indicates the class label assigned to an image I drawn from the labeled image set $\mathcal{L}$, although such labels are not provided in the unlabeled image set $\mathcal{U}$ and the *fake* image set $\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}$ produced by $\mathsf{G}_{\boldsymbol{\eta}}$; note that the *fake* image set $\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}$ can be updated in accordance with the updated $\mathsf{G}_{\boldsymbol{\eta}}$. While both the classifier and the discriminator parameterized by $\boldsymbol{\theta}$ are trained in

(3), according to the standard protocol in GANs [5], the generator $\mathsf{G}_{\boldsymbol{\eta}}$ can be simultaneously learned by adversarially minimizing the following loss to fool the discriminator [16];

$$\min_{\boldsymbol{\eta}} E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{\mathsf{D}_{\boldsymbol{\theta}}(\tilde{I})\}]. \tag{4}$$

In [19], the generator loss is improved for semi-supervised learning by introducing the feature-matching loss of

$$\min_{\boldsymbol{\eta}} \|\boldsymbol{\mu}^l_{\boldsymbol{\theta}} - \tilde{\boldsymbol{\mu}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}}\|_2^2, \tag{5}$$

where $\boldsymbol{\mu}^l_{\boldsymbol{\theta}} = E_{I\in\mathcal{L}\cup\mathcal{U}}[\phi^l_{\boldsymbol{\theta}}(I)]$ and $\tilde{\boldsymbol{\mu}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}} = E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[\phi^l_{\boldsymbol{\theta}}(\tilde{I})]$ are the averaged neuron activations (feature vectors) at the l-th layer $\phi^l_{\boldsymbol{\theta}}$, $l < L$, over *real* and *fake* images, respectively; we use $l = L-1$, the second last layer as shown in Fig. 1. In [19], it is empirically found that the feature-matching loss (5) to train the generator works well in the semi-supervised learning. The GAN-based semi-supervised learning method alternately optimizes the classifier $\phi_{\boldsymbol{\theta}}$/discriminator $\mathsf{D}_{\boldsymbol{\theta}}$ in (3) and the generator $\mathsf{G}_{\boldsymbol{\eta}}$ in (4) or (5).

2.1 Analysis of Gradient

Although the GAN-based method effectively works by simply relating the classifier and the generator via the discriminator, we empirically found, as shown in Sec. 4, that the method is vulnerable to a less discriminative classifier especially trained on the smaller number of labeled samples. In this section, we explain the finding through analyzing the gradient-based updates of the network.

The unsupervised loss regarding the unlabeled images $I \in \mathcal{U}$ in (3) induces the following updates, corresponding to the negative gradients, with respect to the neurons $\{x_c\}_{c=1}^C$;

$$-\frac{\partial}{\partial x_c}[-\log\{\mathsf{D}_{\boldsymbol{\theta}}(I)\}] = \frac{1}{1+\sum_{c'=1}^C \exp(x_{c'})}\frac{\exp(x_c)}{\sum_{c'=1}^C \exp(x_{c'})} = \mathsf{p}(\mathit{fake}|I;\mathsf{D}_{\boldsymbol{\theta}})\mathsf{p}(c|I;\boldsymbol{\theta}). \tag{6}$$

The update (6) pushes the network to further enhance the current prediction $\mathsf{p}(c|I;\boldsymbol{\theta})$ no matter how it is correct or not; the neuron x_{c^*} associated with the predicted class $c^* = \arg\max[\mathsf{p}(c|I;\boldsymbol{\theta})]$ receives the largest update, being further encouraged. It should be also noted that the scaling factor $\mathsf{p}(\mathit{fake}|I;\mathsf{D}_{\boldsymbol{\theta}})$ is not small due to the adversarial training of the generator, ideally reaching $\frac{1}{2}$. Thus, this update works on the basis that the current classifier $\phi_{\boldsymbol{\theta}}$ is so *discriminative* as to produce fairly correct prediction $\mathsf{p}(c|I;\boldsymbol{\theta})$. Such a discriminative classifier, however, is obtained by using considerable amount of labeled samples and/or sufficiently proceeding the learning at the later epochs. It could degrade performance in case that the classifier is immature and less discriminative, producing wrong predictions which are unfavorably enhanced through (6). The issue would manifest itself in the tasks using fewer labeled samples on which the classifier is too poor to exhibit enough discriminativity power especially at the early learning stage. Therefore, we conjecture that due to the updating formula (6) the GAN-based method [19] produces a less effective classifier in the semi-supervised learning tasks of the smaller number of labeled samples.

The same discussion can be applied to the adversarial generator loss (4), though it is replaced with the feature-matching loss (5) in [19]. From the above-mentioned viewpoint, the loss (4) would be less effective for the semi-supervised learning since the prediction by the poor classifier unfavorably affects the updating process of the generator during training.

3 Proposed Method

To remedy the issue discussed in the previous section, we propose the discriminator model and accordingly the loss for training the generator such that their updates are not dependent on the predictions by the (poor) classifier during training. For further improving discriminativity power, we also present an effective method to *mix* the proposed model with the previous one that comprises the discriminator (2) and the generator loss (5); the overall procedure is shown in Algorithm 1.

3.1 Discriminator Model

The proposed discriminator is modeled in the following form:

$$\hat{\mathsf{D}}_{\boldsymbol{\theta}}(I) = \frac{\exp(\frac{1}{C}\sum_{c=1}^{C} x_c)}{1+\exp(\frac{1}{C}\sum_{c=1}^{C} x_c)} = \mathsf{s}\left(\frac{1}{C}\sum_{c=1}^{C} x_c\right) = \mathsf{p}(\mathit{real}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}}), \tag{7}$$

where s indicates a sigmoid function. The proposed discriminator (7) detects *real* images based on the averaged neuron activation, which induces the updates for the discriminator $\hat{\mathsf{D}}_{\boldsymbol{\theta}}$ on *real* images I and *fake* ones $\tilde{I}$ as

$$\begin{aligned} -\frac{\partial}{\partial x_c}[-\log\{\hat{\mathsf{D}}_{\boldsymbol{\theta}}(I)\}] = \frac{1}{C} - \frac{1}{C}\hat{\mathsf{D}}_{\boldsymbol{\theta}}(I) = \frac{1}{C}\mathsf{p}(\mathit{fake}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}}), \\ -\frac{\partial}{\partial x_c}[-\log\{1-\hat{\mathsf{D}}_{\boldsymbol{\theta}}(\tilde{I})\}] = -\frac{1}{C}\mathsf{p}(\mathit{real}|\tilde{I};\hat{\mathsf{D}}_{\boldsymbol{\theta}}). \end{aligned} \tag{8}$$

These update formulas do not resort to the current prediction, $\mathsf{p}(c|I;\boldsymbol{\theta})$ in (1), and all the neuron activations $\boldsymbol{x}$ get unbiased updates with the scale of $\mathsf{p}(\mathit{fake}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}})$ or $\mathsf{p}(\mathit{real}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}})$. These neurons are forced to be activated only on *real* images I, thereby forming a subspace for the features $\boldsymbol{x}$ of *real* images. As the generator provides more realistic *fake* images $\tilde{I}$, the subspace becomes tighter to facilitate learning the classifier which focuses on the labeled samples in the restricted subspace. Through this formulation, the semi-supervised loss (3) is clearly decomposed into two types of learning; one is for a *discriminator* $\hat{\mathsf{D}}_{\boldsymbol{\theta}}$ to differentiate *real* unlabeled images from *fake* images and the other is for a *classifier* $\phi_{\boldsymbol{\theta}}$ across C classes on the labeled samples. Actually, the updates (8) of the discriminator, which are constant over $\{x_c\}_c^C$, are perpendicular to the classifier, not affecting the softmax posterior $\mathsf{p}(c|I;\boldsymbol{\theta}) = \frac{\exp(x_c)}{\sum_c \exp(x_c)} = \frac{\exp(x_c+\beta)}{\sum_c \exp(x_c+\beta)}, \forall \beta \in \mathbb{R}$, while the previous discriminator model (2) makes some effects on the classifier through its updating formula (6). Thus, the proposed discriminator (7) just outlines the subspace of the *real* images so as to improve generalization performance of the classifier on it, leaving the enhancement of discriminativity to the supervised learning of the classifier based on the labeled samples $\mathcal{L}$.

On the other hand, the discriminator (7) is also applicable to the adversarial learning for the generator $\mathsf{G}_{\boldsymbol{\eta}}$. In contrast to the previous model (2,6), our model (7,8) is not dependent on the classifier prediction and thus is freed from the (poor) performance of the classifier during training. Therefore, we can directly apply it to the following generator loss:

$$\min_{\boldsymbol{\eta}} E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{\hat{\mathsf{D}}_{\boldsymbol{\theta}}(\tilde{I})\}]. \tag{9}$$

This loss for the generator $\mathsf{G}_{\boldsymbol{\eta}}$ is *adversarial* to that for the discriminator $\hat{\mathsf{D}}_{\boldsymbol{\theta}}$, which is consistent from the viewpoint of the adversarial learning [5]. The consistency would contribute to increasing stability in training networks.

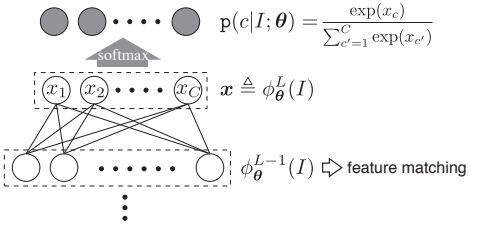

Figure 1: Layers of the network $\phi_{\boldsymbol{\theta}}$. The last output by the L-th layer is fed into softmax to produce $\mathsf{p}(c|I;\boldsymbol{\theta})$, while the $L-1$-th output contributes to the feature-matching loss.

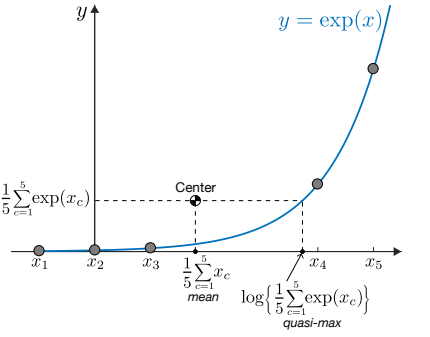

Figure 2: Arithmetic mean and *quasi* max over five features $\{x_c\}_{c=1}^{5}$. The discriminators (7) and (11) are built on them.

3.2 Mixing Two Models

The proposed discriminator (7) effectively copes with the poor classifier which is trained on the smaller number of labeled samples through immature learning at the earlier epochs, as discussed above. On the other hand, for a matured classifier exhibiting enough discriminativity such as at the later learning epochs, it would be beneficial to apply the classifier as a discriminator to exploit the discriminative information of the unlabeled samples via the updating (6), as in the previous model (2). We incorporate those positive aspects into the semi-supervised learning by mixing the two models.

Discriminator

It is straightforward to mix the two models (2) and (7) in terms of losses into

$$\begin{aligned} &wE_{I\in\mathcal{U}}[-\log\{\mathsf{D}_{\boldsymbol{\theta}}(I)\}] + (1-w)E_{I\in\mathcal{U}}[-\log\{\hat{\mathsf{D}}_{\boldsymbol{\theta}}(I)\}] \\ &+ wE_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{1-\mathsf{D}_{\boldsymbol{\theta}}(\tilde{I})\}] + (1-w)E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{1-\hat{\mathsf{D}}_{\boldsymbol{\theta}}(\tilde{I})\}], \end{aligned} \tag{10}$$

with the balancing parameter $0 \le w \le 1$. It, however, does not work at all as demonstrated in Sec. 4.1. The loss (10) implies that *real* and *fake* images are classified by two types of discriminators, which rather lacks consistency from the perspective of the adversarial learning based on single discriminator. Therefore, we here consider to mix them while keeping consistency as follows.

Without loss of generality, we first slightly modify the discriminator $\mathsf{D}_{\boldsymbol{\theta}}$ into

$$\mathsf{D}'_{\boldsymbol{\theta}}(I) = \frac{\frac{1}{C}\sum_{c=1}^{C}\exp(x_c)}{1+\frac{1}{C}\sum_{c=1}^{C}\exp(x_c)} = \mathsf{s}\left(\log\left\{\frac{1}{C}\sum_{c=1}^{C}\exp(x_c)\right\}\right) = \mathsf{p}(\mathit{real}|I;\mathsf{D}'_{\boldsymbol{\theta}}), \tag{11}$$

which provides the same updating as in (6); $-\frac{\partial}{\partial x_c}[-\log\{\mathsf{D}'_{\boldsymbol{\theta}}(I)\}] = \mathsf{p}(\mathit{fake}|I;\mathsf{D}'_{\boldsymbol{\theta}})\mathsf{p}(c|I;\boldsymbol{\theta})$. It should be noted that $\log\{\frac{1}{C}\sum_{c=1}^{C}\exp(x_c)\}$, input to the sigmoid s, is close to $\max_c(x_c)$ [3] as depicted in Fig. 2; thus we call it *quasi* max. From this viewpoint of the sigmoid-based discrimination, the discriminator (11) is sensitive to the maximum neuron activation via the *quasi max* while the proposed discriminator (7) is built on the *mean* activation. On the basis of this analysis, we can mix those discriminators consistently in the sigmoid function;

$$\bar{\mathsf{D}}_{\boldsymbol{\theta}}(I;w) = \mathsf{s}\left((1-w)\frac{1}{C}\sum_{c=1}^{C}x_c + w\log\left\{\frac{1}{C}\sum_{c=1}^{C}\exp(x_c)\right\}\right), \tag{12}$$

where the arithmetic mean and the quasi max are mixed with a linear weight w, followed by the sigmoid function to construct the discriminator. This mixed discriminator provides the following update formula:

$$-\frac{\partial}{\partial x_c}[-\log\{\bar{\mathsf{D}}_{\boldsymbol{\theta}}(I;w)\}] = \mathsf{p}(\mathit{fake}|I;\bar{\mathsf{D}}_{\boldsymbol{\theta}})\left[(1-w)\frac{1}{C}+w\mathsf{p}(c|I;\boldsymbol{\theta})\right], \tag{13}$$

which is regarded as the linear combination of (8) and (6), while the loss-based mixing (10) does not lead to such a formulation but result in

$$-\frac{\partial}{\partial x_c}[-(1-w)\log\{\hat{\mathsf{D}}_{\boldsymbol{\theta}}(I)\}-w\log\{\mathsf{D}'_{\boldsymbol{\theta}}(I)\}] = (1-w)\mathsf{p}(\mathit{fake}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}})\frac{1}{C}+w\mathsf{p}(\mathit{fake}|I;\mathsf{D}'_{\boldsymbol{\theta}})\mathsf{p}(c|I;\boldsymbol{\theta}), \tag{14}$$

where the two fundamental updates $\frac{1}{C}$ and $\mathsf{p}(c|I;\boldsymbol{\theta})$ are weighted by not only w but also $\mathsf{p}(\mathit{fake}|I;\hat{\mathsf{D}}_{\boldsymbol{\theta}})$ and $\mathsf{p}(\mathit{fake}|I;\mathsf{D}'_{\boldsymbol{\theta}})$ of which balance is dependent on the discrminative powers of the two discriminators. Compared with (14), we can see that the proposed mixed discriminator (12) favorably works to mix the two characteristics of the discriminators in learning; one is to tightly shape subspace of *real* images for facilitating the learning of the classifier (Sec. 3.1) and the other is toward discriminative learning (Sec. 2.1).

As discussed in Sec. 3.1, the discriminator (7) is effective at the earlier learning epochs when the classifier is immature, while the other one (11) works on the matured classifier at the later epochs. Thus, we can gradually increase the balancing weight w as the learning proceeds; in this work, w is increased by 0.01 every epoch. As a result, the loss for training a classifier/discriminator in semi-supervised learning is finally defined as

$$\min_{\boldsymbol{\theta}} E_{I\in\mathcal{L}}[-\log\{\mathsf{p}(c|I;\boldsymbol{\theta})\}]+E_{I\in\mathcal{U}}[-\log\{\bar{\mathsf{D}}_{\boldsymbol{\theta}}(I;w)\}]+E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{1-\bar{\mathsf{D}}_{\boldsymbol{\theta}}(\tilde{I};w)\}]. \tag{15}$$

Generator

For increasing stability in training GAN, we slightly modify the feature-matching loss (5) for the generator $\mathsf{G}_{\boldsymbol{\eta}}$ by introducing the matching loss regarding *standard deviation* as shown in (16). This is inspired by the style loss [13] in which two images of different styles are related through matching two kinds of statistics, mean and standard deviation. This helps to effectively move *fake* images to *real* ones especially from the viewpoint of *style*.

We have presented two types of generator losses in (9) and (5). In contrast to the case of mixing discriminators, both losses are defined at different layers (Fig. 1) to optimize the *identical* generator $\mathsf{G}_{\boldsymbol{\eta}}$ and thus can be simply mixed as in deep supervision [24];

$$\min_{\boldsymbol{\eta}}\ (1-w)E_{\tilde{I}\in\mathcal{F}_{\mathsf{G}_{\boldsymbol{\eta}}}}[-\log\{\hat{\mathsf{D}}(\tilde{I})\}]+w\{\|\boldsymbol{\mu}^l_{\boldsymbol{\theta}}-\tilde{\boldsymbol{\mu}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}}\|_2^2+\|\boldsymbol{\sigma}^l_{\boldsymbol{\theta}}-\tilde{\boldsymbol{\sigma}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}}\|_2^2\}, \tag{16}$$

where $\boldsymbol{\sigma}^l_{\boldsymbol{\theta}}=\sqrt{E_{I\in\mathcal{L}\cup\mathcal{U}}[(\phi^l_{\boldsymbol{\theta}}(I)-\boldsymbol{\mu}^l_{\boldsymbol{\theta}})^2]}$ and $\tilde{\boldsymbol{\sigma}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}}=\sqrt{E_{I'\in\mathcal{F}_{\mathsf{G}}}[(\phi^l_{\boldsymbol{\theta}}(I')-\tilde{\boldsymbol{\mu}}^l_{\boldsymbol{\theta},\boldsymbol{\eta}})^2]}$ are the standard deviations of the neuron activations at the l-th layer across *real* and *fake* images, respectively; $l=L-1$ in this study. It should be noted that the balancing weight w is shared with the mixed discriminator (12) so that the discriminator and the generator losses are varied correspondingly and gradually.

The two losses (15) and (16) are alternately minimized according to the standard adversarial learning protocol, finally producing the classifier $\phi_{\boldsymbol{\theta}^*}$; the training procedure is shown in Algorithm 1.

Algorithm 1 : GAN-based Semi-supervised Learning

Input: $\phi_{\boldsymbol{\theta}}$: classifier network, $\mathsf{G}_{\boldsymbol{\eta}}$: generator network, T: number of training epochs
$\mathcal{L}$: labeled image set $\{I, c_I\}$, $\mathcal{U}$: unlabeled set containing only images $\{I\}$,
$\mathcal{Z}$: random value generator for the input of the generator G

1: Initialize network parameters $\boldsymbol{\theta}$ of the classifier (discriminator) and $\boldsymbol{\eta}$ of the generator
2: **for** $t = 1$ to T **do**
3: Set the balancing weight as $w = \min[1, 0.01(t-1)] \in [0,1]$
4: **while** Mini-batch sampling w.r.t $\mathcal{L}, \mathcal{U}$ and $\mathcal{Z}$ **do**
5: [*Discriminator*] Update $\boldsymbol{\theta}$ to minimize the loss (15) based on the gradient (13)
6: [*Generator*] Update $\boldsymbol{\eta}$ to minimize the loss (16)
7: **end while**
8: **end for**

Output: $\boldsymbol{\theta}^*$: optimized network parameter for the classifier $\phi_{\boldsymbol{\theta}}$

4 Experimental Results

The proposed method is applied to semi-supervised classification tasks on MNIST [12], SVHN [15] and CIFAR-10 [10] datasets. We use the same network architectures as in [19]; on MNIST dataset, the classifier network $\phi_{\boldsymbol{\theta}}$ is simply based on multiple-layer perceptron (MLP) with 5 hidden layers and the generator $\mathsf{G}_{\boldsymbol{\eta}}$ is similarly designed as MLP with 2 hidden layers, while on SVHN and CIFAR-10 datasets the convolutional neural network (CNN) of 9 hidden layers is employed as a classifier with the DCGAN [17] generator. The networks are trained by Adam optimizer [8] over 600 epochs through gradually increasing the balancing weight $w = \min[1, 0.01(t-1)]$ where the epoch index is denoted by $t \in \{1, \cdots, 600\}$.

In each dataset, the classifier $\phi_{\boldsymbol{\theta}}$ is learned in a semi-supervised manner on the training set where the labeled samples $\mathcal{L}$ are randomly drawn and the others are regarded as the unlabeled ones $\mathcal{U}$. The random split for picking up the labeled samples is repeated five times on MNIST and three times on the other datasets. We report the averaged error rate on the provided test set. For fairly comparing the performance of semi-supervised learning, we use only the training samples provided in the datasets without augmentation.

4.1 Performance Analysis

We first analyze performance of the proposed method on MNIST dataset. The class labels are assigned only to 1, 3 and 5 training samples per class.

The proposed method mixes the two types of models; the quasi-max based discriminator (11) with feature-matching loss for generator and the mean based discriminator (7) with its adversarial generator loss, which are mixed by a balancing weight w as shown in (15,16). We compared the proposed mixed model with the other configurations in our framework; actually, various types of configurations can be realized by controlling the weight w *separately* in the discriminator loss (15) and the generator loss (16). The performance results of various models are shown in Table 1.

By comparing two methods without mixing (Table 1*i*), the previous method (*i-1*, Sec. 2) in which the parameter updates are affected by immature classifier predictions degrades performance on the smaller number of labeled samples, as described in Sec. 2.1. In contrast, our model (*i-2*, Sec. 3.1) effectively works by excluding its effect from the update formula through the averaged feature representation in the mean-based discriminator (7). Then, as

Table 1: Performance analysis (error rate, %) regarding configurations of discriminator and generator on MNIST. We configure various models (*i*∼*iii*) by setting the balancing weight such as to $w = 0$ or 1 in the discriminator loss (15) and the generator loss (16), separately, while the mixed models (*iv*,*v*) simultaneously increase the weight w from 0 to 1.

			number of labeled samples per class		
	Discriminator (15)	Generator (16)	1	3	5
i-1)	quasi-max ($w = 1$)	feat.-match ($w = 1$)	64.42 ± 2.95	32.08 ± 7.95	7.26 ± 2.22
i-2)	mean ($w = 0$)	adv. loss ($w = 0$)	27.58 ± 3.22	7.71 ± 2.49	4.85 ± 1.41
ii-1)	mix	feat.-match ($w = 1$)	46.10 ± 11.19	14.28 ± 15.70	1.76 ± 1.50
ii-2)	mix	adv. loss ($w = 0$)	41.34 ± 4.58	13.59 ± 5.85	2.33 ± 0.60
iii-1)	quasi-max ($w = 1$)	mix	66.59 ± 4.15	28.50 ± 6.38	7.75 ± 1.49
iii-2)	mean ($w = 0$)	mix	25.01 ± 3.08	6.31 ± 4.92	2.47 ± 1.30
iv)	simple mix (10)	mix	76.79 ± 2.69	68.80 ± 1.99	54.20 ± 6.23
v)	mix	mix	36.92 ± 3.88	9.02 ± 4.63	1.52 ± 0.62

Table 2: Performance comparison (error rate, %) on MNIST. The supervised methods (*i*,*ii*) are trained only on the labeled samples, while the others are semi-supervised methods.

		number of labeled samples per class		
	Method	1	3	5
i)	supervised NN	54.46 ± 5.16	43.39 ± 3.07	31.87 ± 2.81
ii)	HOG [4] + SVM [22]	38.47 ± 3.36	20.89 ± 1.96	13.06 ± 1.50
iii)	HOG [4] + TSVM [7]	46.12 ± 3.92	11.52 ± 4.87	3.54 ± 0.84
iv)	improved GAN-SS[19]	60.74 ± 4.02	22.24 ± 3.99	5.34 ± 2.66
v)	LadderNet [18]	34.10 ± 16.91	5.19 ± 4.65	1.48 ± 0.44
vi)	VAE-SS [9]	14.17 ± 4.61	5.95 ± 0.89	5.05 ± 1.10
vii-1)	Ours (mixed model)	36.92 ± 3.88	9.02 ± 4.63	1.52 ± 0.62
vii-2)	Ours + SVM [22]	15.40 ± 2.61	4.19 ± 2.47	1.29 ± 0.21
vii-3)	Ours + TSVM [7]	15.72 ± 3.33	3.76 ± 3.52	0.93 ± 0.07

to the generator loss, Table 1*ii* demonstrates that the adversarial loss (9) by the mean-based representation contributes to performance improvement on fewer labeled samples, while the feature-matching loss becomes effective as the more samples are labeled. This result implies that the consistency across the generator and discriminator losses is useful for the less-discriminative classifiers trained on the fewer labeled samples. As shown in Table 1*iii* comparing the discriminator models, the mean-based representation outperforms the quasi-max based one. Those two models can be effectively mixed to improve performance especially on the case of 5 labeled samples (Table 1*v*). In summary, by exploiting the mean-based discriminator and its adversarial loss for the generator, the classification performance is improved in the semi-supervised learning on the smaller number of labeled samples. We also show in Table 1*iv* the result by simply mixing the discriminator models in terms of *loss* (10). Such a simple mixing on top of the losses is significantly inferior to our mixing (Table 1*v*), even though the difference between those two models is only in the discriminator loss. Thus, we can conclude that it is important to mix the discriminator models (2,7) into the single model (12) via a sigmoid function for retaining consistency in discriminating *real* and *fake* images.

4.2 Comparison to Other Methods

The proposed method is then compared with the other methods; for supervised learning, (*i*) the classifier of the neural network $\phi_{\boldsymbol{\theta}}$ is trained only on the labeled samples and (*ii*) the hand-

Table 3: Performance comparison (error rate, %) on SVHN.

	Method	number of labeled samples per class 10	30	50
i)	supervised CNN	81.87 ± 1.43	66.18 ± 3.47	44.59 ± 3.80
ii)	HOG [9]+SVM [22]	36.32 ± 1.54	28.61 ± 1.36	26.41 ± 1.34
iii)	HOG [9]+TSVM [7]	33.62 ± 2.86	22.88 ± 0.99	22.17 ± 0.61
iv)	improved GAN-SS[19]	79.60 ± 2.23	29.62 ± 12.57	18.57 ± 2.51
v)	TempEns [11]	75.00 ± 3.74	35.91 ± 3.21	16.87 ± 8.76
vi)	VAT [14]	57.78 ± 5.47	15.70 ± 2.21	9.17 ± 0.94
vii-1)	Ours (mixed model)	62.80 ± 8.55	11.45 ± 1.64	7.16 ± 0.63
vii-2)	Ours + SVM [22]	47.54 ± 6.40	10.31 ± 0.81	7.75 ± 0.24
vii-3)	Ours + TSVM [7]	51.85 ± 8.12	8.35 ± 0.26	7.05 ± 0.38

crafted HOG feature [9] is combined with linear SVM [22] classifier, while semi-supervised methods include (*iii*) HOG feature with transductive SVM (TSVM) [7], (*iv*) GAN-based semi-supervised method [19] and (*v*,*vi*) the other neural-network based methods [9, 11, 14, 18]; we used the codes provided by the authors for the methods [9, 11, 14, 18, 19]. In addition to our classifier (*vii-1*) trained in an end-to-end semi-supervised manner, the supervised SVM (*vii-2*) and semi-supervised TSVM (*vii-3*) classifiers are also applied to the features $\phi^{L-1}_{\boldsymbol{\theta}^*}$ of the neuron activations at the $L-1$-th layer in our trained network $\phi_{\boldsymbol{\theta}^*}$. This can be fairly compared to the hand-crafted HOG features from the viewpoint of feature extraction. Note again that all these methods are trained on the same set of labeled samples whose number per class is varied over 1, 3 and 5 on MNIST dataset and 10, 30 and 50 on SVHN and CIFAR-10 datasets. We apply the proposed method of mixed model (Table 1*v*) with gradually increasing the balancing weight $w = \min[1, 0.01(t-1)]$.

The performance comparison results are shown in Table 2, Table 3 and Table 4 for MNIST, SVHN and CIFAR-10 datasets, respectively. On MNIST (Table 2) of less complex image patterns, *i.e.*, hand-written digits, the simpler methods produce favorable performance, such as hand-crafted HOG (*iii*) and VAE-SS (*vi*) which applies dimensionality reduction via variational auto-encoder as pre-processing, while the supervised neural network (*i*) deteriorates due to such a small amount of labeled training data. The proposed method (*vii-1*) exhibits superior performance especially on 3 and 5 labeled samples, improving the performance of the neural network (*i*). The features provided by our trained network effectively work with the supervised SVM (*vii-2*) and the semi-supervised TSVM (*vii-3*) classifiers, demonstrating that our semi-supervised learning endows the intermediate ($L-1$-th) layer with the discriminative power; the maximum-margin criterion by SVMs enables us to further boost the performance on the smaller number of labeled samples. We can also observe the similar performance comparison on the other datasets (Table 3&4), where although the supervised neural network (*i*) is inferior even to the HOG+SVM model (*ii*), it is significantly improved by our method to produce favorable performance in comparison to the other methods.

Our GAN-based method considers *out-of-sample* (*fake*) representation via GAN to make the classifier concentrate on *real* samples, while the other methods operate *within* the training (*real*) samples. Thus, it is noteworthy that our GAN-based method can compensate and work together with the other semi-supervised methods such as [11, 14, 18] for further improving performance, which is our future work.

Table 4: Performance comparison (error rate, %) on CIFAR-10.

		number of labeled samples per class		
	Method	10	30	50
i)	supervised CNN	75.26 ± 1.67	68.89 ± 0.44	64.40 ± 0.61
ii)	HOG [4]+SVM [22]	71.28 ± 0.29	65.49 ± 0.85	62.28 ± 0.40
iii)	HOG [4]+TSVM [7]	72.26 ± 1.01	66.63 ± 0.93	63.06 ± 0.14
iv)	improved GAN-SS[19]	42.00 ± 0.42	30.57 ± 2.12	24.70 ± 1.20
v)	TempEns [11]	78.14 ± 0.79	55.15 ± 1.98	49.69 ± 5.21
vi)	VAT [14]	51.89 ± 0.89	34.62 ± 2.82	27.77 ± 1.17
vii-1)	Ours (mixed model)	27.66 ± 1.20	23.58 ± 0.81	21.92 ± 0.21
vii-2)	Ours + SVM [22]	27.54 ± 0.49	22.77 ± 0.74	21.53 ± 0.12
vii-3)	Ours + TSVM [7]	27.98 ± 0.91	22.65 ± 0.08	21.91 ± 0.41

5 Conclusion

In this paper, we have proposed a method to learn a classifier by exploiting GAN in the framework of semi-supervised learning especially on the smaller number of labeled samples. Based on the analysis of the gradients in the discriminator model, we formulate an effective discriminator model by leveraging the mean of neuron activations to cope with a less-discriminative classifier trained on fewer labeled samples. The proposed model is then mixed with the previous one via the sigmoid-based representation of discriminator to further improve the discriminativity. In the experiments on semi-supervised classification tasks using MNIST, SVHN and CIFAR-10 datasets, the proposed method exhibits favorable performance compared to the other methods.

References

[1] M. Belkin and P. Niyogi. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, (48): 1–36, 2006.

[2] H. Cheng, Z. Liu, and J. Yang. Sparsity induced similarity measure for label propagation. In *International Conference on Computer Vision*, 2009.

[3] N. Cohen, O. Sharir, and A. Shashua. Deep simnets. In *CVPR*, pages 4782–4791, 2016.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages –, 2014.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017.

[7] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[9] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, pages 3581–3589, 2014.

[10] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[11] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. In *IJCAI*, pages 2230–2236, 2017.

[14] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv,* 1704.03976, 2017.

[15] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[16] A. Odena. Semi-supervised learning with generative adversarial networks. In *ICML Workshop*, 2016.

[17] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[18] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, pages 3546–3554, 2015.

[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242, 2016.

[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, pages 2242–2251, 2017.

[21] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR*, 2016.

[22] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[23] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan. Linear neighborhood propagation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1600–1615, 2009.

[24] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015.

[25] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007.

[26] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pages 5908–5916, 2017.

[27] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, pages 3774–3782, 2017.