ActiveGAN: Enhanced Image Classification with Active Generative Adversarial Network

Quan Kong[1,a)] Bin Tong[1] Klinkigt Martin[1] Yuki Watanabe[2] Naoto Akira[1]
Tomokazu Murakami[1]

Abstract: Sufficient supervised information is crucial for any machine learning models to boost performance. However, labeling data is expensive and sometimes difficult to obtain. Active learning is an approach to acquire annotations for data from a human oracle by selecting informative samples with a high probability to enhance performance. In recent emerging studies, a generative adversarial network (GAN) has been integrated with active learning to generate good candidates to be presented to the oracle. In this paper, we propose a novel model that is able to obtain labels for data in a cheaper manner without the need to query an oracle. In the model, a novel reward for each sample is devised to reflect the degree of *uncertainty*, which is obtained from a classifier trained with existing labeled data. This reward is used to guide a conditional GAN to generate informative samples with a higher probability for a certain label. With extensive evaluations, we have confirmed the effectiveness of the model, showing that the generated samples are capable of improving the classification performance in popular image classification tasks.

1. Introduction

Machine learning models including traditional ones and new emerging deep neural networks require sufficient supervised information, i.e., class labels, to achieve fair performance. In situations in which labeled data is expensive or difficult to obtain, these models degenerate in performance. Active learning [21] is proposed for handling such a problem. It aims to find the best approach to leverage a limited number of labeled data and to reduce the cost of data annotation. Active learning selects informative samples from a pool of unlabeled data and obtains their labels by involving a human oracle. In this paper, we investigate the problem of *lack of labeled data* from a new and different perspective. We propose a model to improve learning performance, which is able to make use of limited labeled data without using any additional unlabeled data nor involving any human oracle to acquire labels.

As to a classification model, informative samples are those that are able to better contribute to improve classification performance than other samples. For example, samples close to the hyperplane are often *uncertain* for a support vector machine (SVM) based classifier. Therefore, acquiring labels of those samples can reduce the *uncertainty*, thereby reducing classification errors. In the area of active learning, informative samples are selected from a pool of unlabeled data by using criteria, such as degree of *uncertainty*. The labels of the selected samples are obtained by querying a human oracle. Recently, there have been attempts [10], [29] which label informative samples generated from a generative adversarial network (GAN) [9]. In these works, GAN is used to generate samples with the same distribution as the unlabeled dataset. In [29], latent variables, which are able to generate samples that have small distances to the classification hyperplane, are selected. These latent variables are used to generate samples that are labeled by involving an oracle. In [10], a GAN is used to generate samples compound of two classes and the human oracle has to choose the sample which cannot be clearly assigned to either class. However, the above methods still need to use a pool of unlabeled data and query the human oracle.

Unlike the above methods, we investigate the problem of how to acquire labeled data to improve the classification performance by only using a limited number of labeled data. A straightforward way is to use conditional GAN [15], [18] for generating labeled samples. However, for a class of samples, most generated samples may fall inside its convex hull. Samples inside this convex hull are less discriminative to other classes, while samples along or even outside of the convex hull are informative to optimize the hyper-plane of the classifier. In the idea of active learning, a classifier trained with existing labeled data provides a signal to determine if a sample is uncertain to the hyper-plane of the classifier. In this work, we use this external signal to guide the conditional GAN in generating informative labeled samples with a higher probability that contribute to improving classification performance. This can be regarded as an optimization with a tradeoff between generation of samples with the same distribution as the training samples and generation of informative samples. This optimization philosophy is widely used in machine learning, such as penalizing complexity of parameters to avoid over-fitting. The contribution of our work is two-fold:

(1) We propose a model that provides a cheaper way than active

[1] Hitachi, Ltd. Research and Development Group
[2] Hitachi America, Ltd
[a)] quan.kong.xz@hitachi.com

learning to acquire labeled samples. Instead of querying the oracle, our model generates labeled samples with a higher probability that are informative to optimize the hyper-plane of a classifier.

(2) We propose a novel loss function for training generative network model to generate informative samples with a specific label that is inspired by the idea of policy gradient [25] in reinforcement learning. We regard generated samples and the external signal related to *uncertainty* as *action* and *reward*, and use this reward to update the parameters of network for generating informative samples.

2. Related Work

Handling a limited amount of labeled data is a long-standing and important problem in the area of machine learning. Different philosophies and problem settings exist for dealing with this problem, such as transfer learning [19] and active learning [21].

Transfer learning focuses on how to optimize a model with a limited amount of labeled data by transferring the knowledge from a similar yet different source task with sufficient labeled data, thereby reducing the cost of data annotation for the task at hand. Zero-shot learning [8], [13] is a variation of transfer learning, in which *unseen*, and therefore un-labeled object are expected to be recognized by transferring knowledge from *seen* classes [21]. Active learning [21] is based on a different philosophy. Typically, a pool of unlabeled data is available to this learning paradigm, in which the most informative samples from the pool of unlabeled data are selected to query an oracle. Typically active learning provides a schema to limited the number of queries by selecting the most informative samples to maximize the effect of the acquired labels. To determine the degree of *uncertainty* used in the query strategy, uncertainty sampling [11] is the most simple, yet widely used criterion to measure informativeness. Other criteria for the query strategy may include query by committee (QBC) [7], expected error reduction [16], [20] and density weighted methods [2], [6], [23].

A generative adversarial network (GAN) [9] is a neural network model trained in an unsupervised manner, aiming to generate new data with the same distribution as the data of interest. It is widely applied in computer vision and natural language processing tasks, such as generating samples of images [5] and generating sequential words [14]. One of its variants, conditional GAN [15], uses both label information and noisy latent variables to generate samples for a specified label. A variant of conditional GAN, called Auxiliary Classifier GAN (AC-GAN) [18], uses supervised information to generate high quality images at pixel level.

Recently, the GAN models have been used with transfer learning [1], [4], zero-shot learning [26], [27] and active learning [10], [29]. In most these studies, the GAN models are used to generate expected candidates that help to solve problems in zero-shot learning and active learning. Unlike the works [10], [29] in which the generated samples are presented to the oracle, this work focuses on directly generating informative labeled samples that might contribute to boosting learning performance. This does not require to query an oracle. To the best of our knowledge, this is the first study that uses a GAN to generate informative samples by incorporating a new devised factor to measures the degree of *uncertainty*. This study provides a new paradigm that augments labeled data to improve learning performance without using any other unlabeled data nor involving a human oracle.

3. Preliminary

In this section, we introduce the preliminaries of GAN and active learning that serves the basis to derive our new model. A GAN [9] consists of a generator G and discriminator D that compete in a turn-wise min-max game. The discriminator attempts to distinguish real samples from synthetic samples, and the generator attempts to fool the discriminator by generating synthetic samples looking like real samples. The D and G play the following game on $V(D,G)$

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}_i \in p_{data}(\mathbf{x})}[\log D(\mathbf{x}_i)] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z)))], \tag{1}$$

where $\mathbf{x}_i$ represents a sample. p_{data} and p_z represent the distribution of real samples and synthetic samples, and z represents a noise vector. A GAN tries to close p_z to p_{data}; that means the generated samples from G are desired to own a high likelihood to p_{data} which is also the distribution of training samples. In the original GAN model only z is used to generate samples. In a variation called conditional GAN (CGAN) [15], a condition y_i, which is a class label of $\mathbf{x}_i$, is included in addition to z to control the sample generation. The objective function becomes

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}_i \in p_{data}(\mathbf{x})}[\log D(\mathbf{x}_i|y_i)] + \mathbb{E}_{z \in p_z(z)}[\log(1 - D(G(z|y_i)))], \tag{2}$$

where y_i could be a one-hot representation of the class label. During training of the CGAN model, y_i is used to instruct the generator G to synthesize samples for this given class.

Active learning is a machine learning method that is able to interactively query an oracle to obtain labels for samples. These samples are selected from an unlabeled sample pool by using a criterion to measure if the selected sample is able to reduce the learning error. To be more specific, standard supervised learning problems assume an instance space of data X and labels Y. A mapping function $f : X \to Y$ is optimized by minimizing error:

$$f^* = \arg\min_{f \in F} \sum_Y L(f(X), Y), \tag{3}$$

where F represents a space over a predefined class of functions. The error is measured by a loss function L that penalizes disagreement between $f(X)$ and Y. In the typical setting of active learning [22], a pool of unlabeled samples $U = \{\mathbf{x}_1^u, \ldots, \mathbf{x}_n^u\}$ is given. Denote $M = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in X$ and $y_i \in Y$. Active learning performs in an iterative way: (1) training a classifier f on M; (2) using a query function $Q(f, M, U)$ to select an unlabeled sample i^* to label; (3) removing $\mathbf{x}_{i^*}^u$ from U and adding $(\mathbf{x}_{i^*}^u, y_{i^*})$ to M. The target of active learning is to choose samples i^* to be labeled by asking an oracle, and reduce the learning error with as few queries as possible. The selected samples are regarded as more informative than other unselected ones in terms of contribution in learning error reduction.

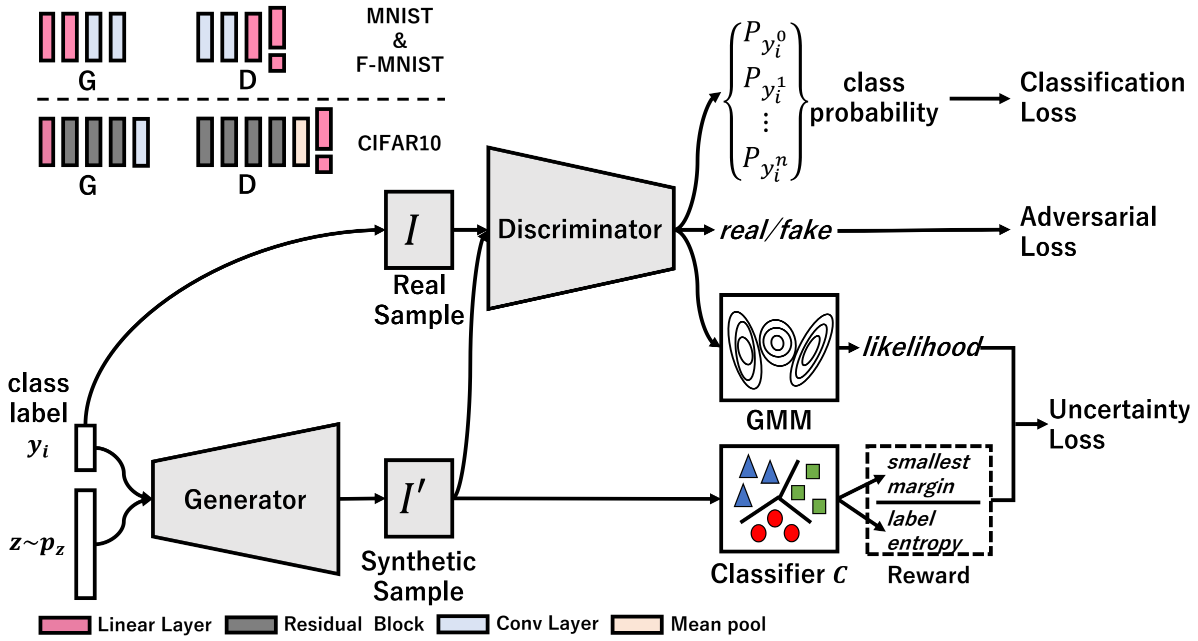


Fig. 1: Architecture of our proposed model. (1) a conditional GAN. It generates samples by training a generator and a discriminator with a class label one hot vector y_i from training sample as a condition. In our case, we introduce a Classification Loss follows AC-GAN [18] to make sure the generated sample is highly related to the specific given label condition. The discriminator also determines the possibility if a generated sample is fake used as our Adversarial Loss to make the distributions of generated and real samples similar. Simultaneously, (2) a classifier C trained with existing labeled data. It calculates a reward for a sample related to the degree of *uncertainty* includes smallest margin and label entropy. (3) a Gaussian mixture model (GMM). It calculates a likelihood of a sample to be generated from current distribution of synthetic samples. Uncertainly Loss consists of a likelihood and rewards of each generated sample, and we use it to train the generator for generating informative samples via policy gradient.

4. Proposed Method

In this section, we discuss the details of the proposed method. Without loss of generality, we take a classifier as the example of supervised learning, in which we are given a set of labeled data $S_l = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i$ is a sample, y_i is its corresponding label, and N represents the number of samples.

The overview of our proposed model is shown in Figure 1. This model mainly consists of a classifier C trained with existing labeled data, a conditional GAN, and a Gaussian mixture model (GMM). The conditional GAN is used to generate labeled samples. A novel reward is devised to measure the degree of informativeness for each generated sample. This reward is calculated according to a degree of *uncertainty* for a sample with respect to the hyper-plane of the pretrained classifier. In general, the more informative a sample is, the higher probability this sample is able to improve classification performance if it is included in the existing labeled data. The GMM model provides a likelihood of a generated sample to be generated from a recent set of generated samples. Together with the likelihood, the reward is used to update parameters for the generator in the conditional GAN. This model makes a trade-off between generating samples with the same distribution as the labeled data and generating informative samples to improve classification performance for the pretrained classifier.

4.1 Generation of labeled samples

Since we focus on image classification tasks, the proposed model uses a variant of conditional GAN, called AC-GAN [18], which shows its promising performance on generating images.

Given a set of labeled images $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, the AC-GAN model is used to generate labeled samples with the inputs of both a noise latent variable and a one-hot representation of a class label. In the AC-GAN, the generator G generates a synthetic sample $\widehat{\mathbf{x}}_i = G(z, y_i)$ with the noise latent vector z and a label y_i. The discriminator gives two kinds of probabilities. One is a probability distribution over sources, i.e., $P(\text{real}|\mathbf{x}_i)$ and $P(\text{fake}|\widehat{\mathbf{x}}_i)$. The other one is posterior probabilities over the class labels, i.e., $P(y_i|\mathbf{x}_i)$ and $P(y_i|\widehat{\mathbf{x}}_i)$. The objective functions of generator and discriminator in the AC-GAN are formulated as

$$\begin{aligned} L^D_{\text{AC-GAN}} = {} & \mathbb{E}[\log P(\text{real}|\mathbf{x}_i)] + \mathbb{E}[\log P(\text{fake}|\widehat{\mathbf{x}}_i)] + \\ & \mathbb{E}[\log P(y_i|\mathbf{x}_i)] + \mathbb{E}[\log P(y_i|\widehat{\mathbf{x}}_i)] \end{aligned} \qquad (4)$$

$$L^G_{\text{AC-GAN}} = \mathbb{E}[\log P(\text{real}|\widehat{\mathbf{x}}_i)] + \mathbb{E}[\log P(y_i|\widehat{\mathbf{x}}_i)]. \qquad (5)$$

The discriminator D is trained to maximize $L^D_{\text{AC-GAN}}$, and the generator G is trained to maximize $L^G_{\text{AC-GAN}}$. For the discriminator D, the first two terms in Equation 4 expect that both real and fake samples are classified correctly. The last two terms in Equation 4 expect that both real and fake samples have correct class labels. For the generator G, it is expected that generated samples are classified as fake, and have correct class labels as well.

4.2 Measure of uncertainty

In this subsection, we discuss how the degree of *uncertainty* is measured in the proposed model. Among the samples generated by the AC-GAN model, only informative samples might be able to contribute to improving classification performance. In the area of active learning, uncertainty sampling is the most widely used query strategy. The intuition behind uncertainty sampling is that if a sample is highly uncertain with a hyper-plane of a classifier, obtaining its label will improve the degree of discrimination among classes. In other words, this sample is considered to be informative in improving the classification performance. In our model, we use SVM as the classifier. In our paper, we mainly use two metrics based on the label probabilities to measure the uncertainty of a sample.

Smallest Margin Margin sampling is an uncertainty sampling method in the case of multi-class [21], which is defined as

$$\widehat{\mathbf{x}}_M = \arg\min_{\widehat{\mathbf{x}}_i}(P(y'_1|\widehat{\mathbf{x}}_i) - P(y'_2|\widehat{\mathbf{x}}_i)), \tag{6}$$

where y'_1 and y'_2 are the first and second most probable class labels of a generated sample $\widehat{\mathbf{x}}_i$ under the specified classifier, respectively. Intuitively, samples with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Samples with small margins are more ambiguous, thus knowing the true label will help the model to discriminate more effectively between them.

Label Entropy A more general uncertainty sampling strategy uses the entropy of posterior probabilities over class labels. In smallest margin, posterior probabilities of labels other than the two most probable class labels are simply ignored. To mitigate this problem, the entropy over all class labels is used, which is formulated as

$$\widehat{\mathbf{x}}_{LE} = \arg\max_{\widehat{\mathbf{x}}_i} - \sum_{y'} p(y'|\widehat{\mathbf{x}}_i) \log p(y'|\widehat{\mathbf{x}}_i). \tag{7}$$

4.3 Loss on uncertainty

In this subsection, we discuss how to devise a loss function for the generated samples based on the degree of *uncertainty* to update the parameters of the generator.

Policy gradient [25] has been successfully applied in reinforcement learning to learn an optimal policy. As one target of this work is to guide the generator to synthesize informative samples, we regard the degree of *uncertainty* and the generated samples as *reward* and *action*, respectively. In general, the higher the degree of *uncertainty* is, the higher the reward becomes. If a generated sample has a high degree of *uncertainty*, this sample is encouraged to be generated with a high probability. To the best of our knowledge, we are the first to use the idea from policy gradient to model the degree of *uncertainty* in active learning.

In the following we discuss how to convert the degree of *uncertainty* into a reward. With respect to smallest margin, for each generated sample $\widehat{\mathbf{x}}_i$, the reward is calculated by $\mathbf{r}_m(\widehat{\mathbf{x}}_i) = e^{(1-u_m)}$, where $u_m = P(y'_1|\widehat{\mathbf{x}}_i) - P(y'_2|\widehat{\mathbf{x}}_i)$. If the difference between the probabilities of the two most probable class labels for a generated sample is small, it means this generated sample is uncertain. This results in a larger value of $\mathbf{r}_m$ than other certain samples. With respect to label entropy, we can calculate the reward similar to $\mathbf{r}_{le}(\widehat{\mathbf{x}}_i) = e^{u_{le}}$, where $u_{le} = -\sum_{y'} p(y'|\widehat{\mathbf{x}}_i) \log p(y'|\widehat{\mathbf{x}}_i)$. The reward for a generated sample is calculated by combining the above two factors, which is formulated as

$$\mathbf{r}(\widehat{\mathbf{x}}_i) = \alpha \cdot \mathbf{r}_m(\widehat{\mathbf{x}}_i) + (1-\alpha) \cdot \mathbf{r}_{le}(\widehat{\mathbf{x}}_i), \tag{8}$$

where α is a parameter that balances the importance between the two metrics of smallest margin and label entropy. According to policy gradient, we devise the loss for generated samples formulated as follows:

$$L_{\text{uncertainty}} = \sum_{\widehat{\mathbf{x}}_i} \mathbf{r}(\widehat{\mathbf{x}}_i) P(\widehat{\mathbf{x}}_i|\Theta), \tag{9}$$

where $\widehat{\mathbf{x}}_i$ represents a generated sample from the generator $G(z, y_i)$, $P(\widehat{\mathbf{x}}_i|\Theta)$ represents the probability of $\widehat{\mathbf{x}}_i$ that is generated by the generator. However, the generator does not directly provide such a probability for each generated sample. Therefore, we have to estimate this probability based on a model with the parameters Θ. In our work, we choose GMM. To estimate this probability in a dynamic manner during the training of model, we set a buffer to store the latest m batches of generated samples to train the GMM model, and estimate the probabilities of generated samples in the current batch. The intuition to set a buffer is to make a trade-off between the instability of modeling GMM caused when using only a limited number of samples, and using the most recent samples only to represent the current parameters Θ accurately. During the training, the distribution of the generated samples becomes close to that of the original samples. Therefore, using the generated samples from earlier batches is improper to model the latest distribution of generated samples. The GMM models a mixture of Gaussians for the generated samples in the buffer with parameters $\Theta = \{\mu, \mathbf{\Sigma}\}$, where $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$ and $\mathbf{\Sigma} = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_K\}$, where μ_i and Σ_i are the mean and deviation of the i-th component in the mixture Gaussians, K is the number of mixture Gaussians. Note that the features of the generated samples are the output of the convolution layers in the discriminator D of the AC-GAN model. These features are shared by the two kinds of probabilities, which are probability distribution over sources and posterior probability over the class labels.

4.4 Algorithm

By integrating the loss measuring the degree of *uncertainty* for the generated samples, our proposed model, called ActiveGAN, has the following loss function for the generator, which is maximized.

$$\begin{aligned} L^G_{\text{ActiveGAN}} &= L^G_{\text{AC-GAN}} + \lambda L_{\text{uncertainty}} \\ &= \mathbb{E}[\log P(\text{real}|\widehat{\mathbf{x}}_i)] + \mathbb{E}[\log P(y|\widehat{\mathbf{x}}_i)] \\ &\quad + \lambda \mathbb{E}[\log P(\widehat{\mathbf{x}}_i|\Theta)\mathbf{r}(\widehat{\mathbf{x}}_i)], \end{aligned} \tag{10}$$

where $L^G_{\text{AC-GAN}}$ is the loss function of the generator in AC-GAN. The discriminator in Active-GAN is the same as that in AC-GAN, which is denoted by $L^D_{\text{ActiveGAN}}$. The notations Ψ_g and Ψ_d in Algorithm 1 represent the parameters of generator and discriminator in the ActiveGAN, respectively. λ is a parameter that balances the importance between the loss for the generator in the AC-GAN

Algorithm 1 ActiveGAN

```
Input training data $\mathbf{x}_i$ and its label $y_i$ where $i \in [1, \ldots, N]$.
Output $\Psi_d$ and $\Psi_g$
Initialize $\alpha$, $\lambda$, $\Theta$, $\Psi_d$ and $\Psi_g$.
Set the buffer size to be $M$
Train SVM with grid-search for best parameters
Train the generator $G$ and the discriminator $D$ with first $m$ iterations
Save generated samples in $m$ iterations into the buffer
repeat
    Generate a sample $\widehat{\mathbf{x}}_i \leftarrow G(z, y_i)$
    Use Equation 8 to calculate the reward $\mathbf{r}(\widehat{\mathbf{x}}_i)$ for $\widehat{\mathbf{x}}_i$.
    Use generated samples in the buffer to estimate parameters $\Theta$ of GMM
    Calculate the probability $P(\widehat{\mathbf{x}}_i|\Theta)$ for $\widehat{\mathbf{x}}_i$
    Use Equation 9 to calculate the loss $L_U$ related to the degree of
    uncertainty for $\widehat{\mathbf{x}}_i$
    Update parameters for the generator $G$: $\Psi_g \leftarrow \triangledown_{\Psi_g} L^G_{\text{ActiveGAN}}$
    Update parameters for the discriminator $D$: $\Psi_d \leftarrow \triangledown_{\Psi_d} L^D_{\text{AC-GAN}}$
    Update the buffer by adding the sample $\widehat{\mathbf{x}}_i$
until
```

model and the loss related to the degree of *uncertainty* for the generated samples. The larger the value of λ is, the more likely the model is forced to generate samples that contribute to improving the classification performance instead of generating samples whose distribution is the same as the training ones. The learning process of ActiveGAN is detailed as pseudo-code in Algorithm 1.

The evaluation of ActiveGAN is conducted as follows. We use the trained generator G to synthesize a specific number of samples, which we denoted by $\mathcal{S}_g$. Together with the labeled data $\mathcal{S}_l$, we retrain the SVM to examine improvements in the classification performance.

5. Experiments

5.1 Evaluation settings

We utilize three datasets CIFAR10 [12], MNIST [17] and Fashion-MNIST [28] for evaluation of the proposed model ActiveGAN. MNIST consists of 50,000 training samples, 10,000 validation samples and 10,000 testing samples of handwritten digits of size 28 × 28. CIFAR10 has colored images for 10 general classes. Again we find 50,000 training samples and 10,000 testing samples of size 32 × 32 in CIFAR10. Fashion-MNIST has a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28 × 28 grayscale image, associated with a label from 10 different classes associated with fashion items.

We used the same network structure for the generator and discriminator as in [18] for CIFAR10 and [3] for MNIST and Fashion-MNIST. To train a stable ActiveGAN, the parameters of the discriminator are updated once after those of the generator are updated for a specified number of iterations. Adam was used as the gradient method for learning parameters of the network. Its initial learning rate is searched in the set $\{0.0002, 0.001\}$. We used SVM as classifier, which is trained using grid-search for the best hyper-parameters. We used a pre-trained VGG-16 [24] to extract features for images for all datasets. The balancing parameter α in Equation 8 was set to 0.5. The balancing parameter λ in Equation 10 was set to 10 to guarantee that values of two terms $L^G_{\text{AC-GAN}}$ and L_U are in the same scale.

Table 1: F-score of models on CIFAR10, MNIST, Fashion-MNIST (F-MNIST). n represents the number of labeled images used for training SVM.

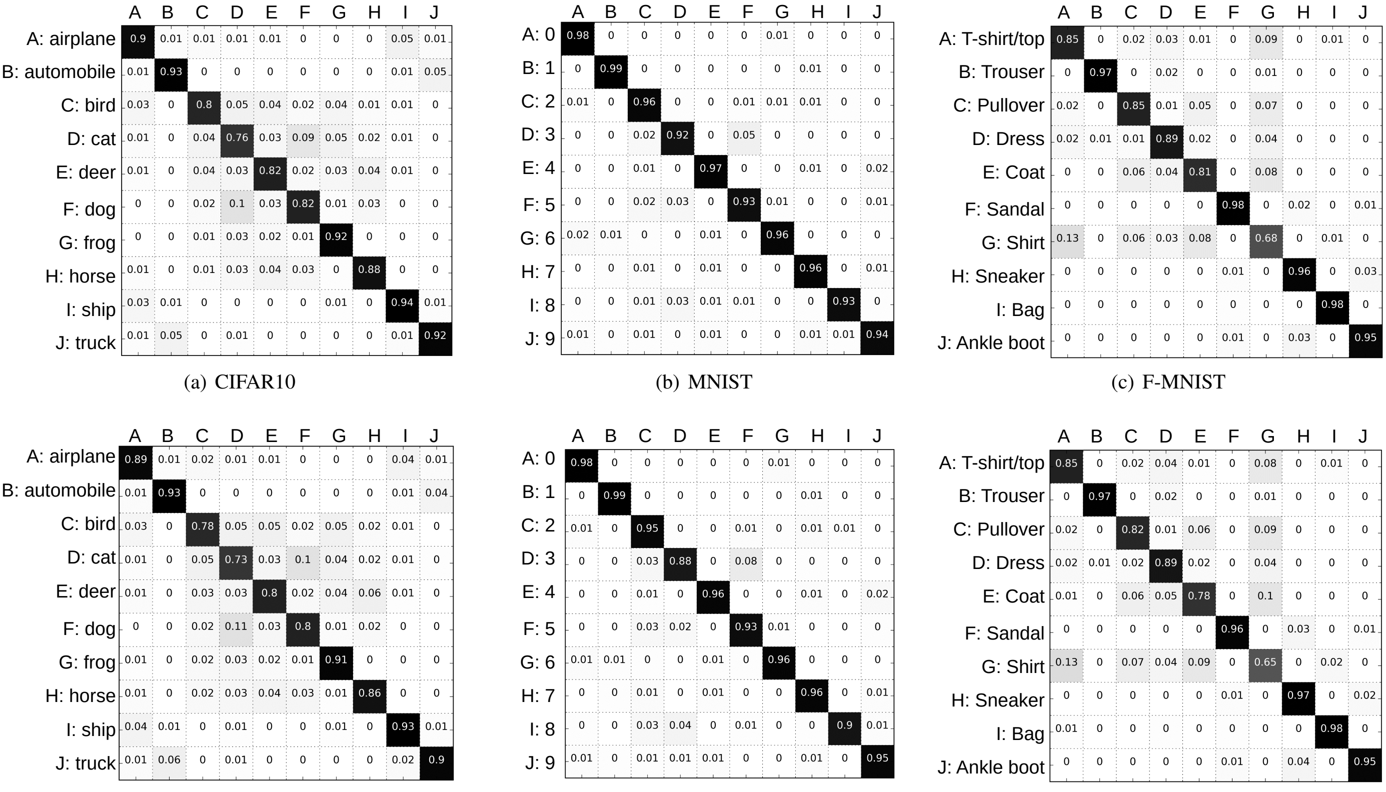
	CIFAR10		MNIST		F-MNIST	
Method	5k	10k	500	1k	5k	10k
SVM	83.4	85.3	94.6	96.2	87.1	88.1
AC-GAN	81.4	82.7	94.1	95.8	85.4	86.4
AC-GAN+F	82.5	83.2	94.5	95.9	86.2	87.3
ActiveGAN	83.3	84.8	94.9	96.1	86.8	87.8
ActiveGAN+F	**84.5**	**86.8**	**95.3**	**96.8**	**87.9**	**89.2**

We compare the performance of the proposed model for a number of settings. Together with images in the training set, the generated images from AC-GAN or ActiveGAN are used to retrain the SVM. According to the principles of AC-GAN and ActiveGAN, not all generated images are able to improve the classification performance. Due to this reason, we have two different settings for dealing with those generated images. The first setting is to use all generated images, and the second one is to apply a constraint to filter out images that are not regarded as informative by the criteria used in active learning. The margin over posteriors of most the two most probable class labels, which is calculated by Equation 6, is used as the constraint. The margin is also tuned in our evaluation. We use notations AC-GAN+F and ActiveGAN+F to represent that the margin constraint is applied to the generated samples from AC-GAN and ActiveGAN. As the image classification task in this evaluation is a multi-class problem, the F-score is used as the metric for evaluating performance. F-score is calculated by $\frac{2 \cdot P \cdot R}{P + R}$, where P are R are precision and recall, respectively.

5.2 Image Classification

Table 1 shows the classification performances of models for three different datasets. Note that the threshold for the margin constraint is tuned for each data set in the set $\{0.1, 0.15, 0.20, 0.25, 0.3\}$. The best F-scores are shown. MNIST is a simple dataset. Using a SVM with advanced features easily achieves almost 99% classification accuracy. In such a case, it is not easy to assess the effectiveness of the generated samples. Therefore, we artificially limited the number of labeled samples available for training to n chosen as 500 or $1,000$ for MNIST and as $5,000$ or $10,000$ for the other data-sets. Note that all testing samples are used in each dataset.

We can see that when the number of images used for training the SVM increases, F-scores are improved for every dataset, which is consistent with common sense. When the generated images from AC-GAN are used for training together with the existing labeled images, the classification performance even drops. For example, in CIFAR10, the classification performance drops by 2.0% and 2.5% for the case of n set to $5,000$ and $10,000$, respectively. By applying the margin constraint, the classification performance is slightly improved. However, the side effect is not fully mitigated, and performance is still lower as compared to the baseline SVM. This empirically explains that the generated samples from AC-GAN, whose distribution is similar to that of



	A	B	C	D	E	F	G	H	I	J
A: airplane	0.9	0.01	0.01	0.01	0.01	0	0	0	0.05	0.01
B: automobile	0.01	0.93	0	0	0	0	0	0	0.01	0.05
C: bird	0.03	0	0.8	0.05	0.04	0.02	0.04	0.01	0.01	0
D: cat	0.01	0	0.04	0.76	0.03	0.09	0.05	0.02	0.01	0
E: deer	0.01	0	0.04	0.03	0.82	0.02	0.03	0.04	0.01	0
F: dog	0	0	0.02	0.1	0.03	0.82	0.01	0.03	0	0
G: frog	0	0	0.01	0.03	0.02	0.01	0.92	0	0	0
H: horse	0.01	0	0.01	0.03	0.04	0.03	0	0.88	0	0
I: ship	0.03	0.01	0	0	0	0	0.01	0	0.94	0.01
J: truck	0.01	0.05	0	0.01	0	0	0	0	0.01	0.92

(a) CIFAR10

	A	B	C	D	E	F	G	H	I	J
A: 0	0.98	0	0	0	0	0	0.01	0	0	0
B: 1	0	0.99	0	0	0	0	0	0.01	0	0
C: 2	0.01	0	0.96	0	0	0.01	0.01	0.01	0	0
D: 3	0	0	0.02	0.92	0	0.05	0	0	0	0
E: 4	0	0	0.01	0	0.97	0	0	0.01	0	0.02
F: 5	0	0	0.02	0.03	0	0.93	0.01	0	0	0.01
G: 6	0.02	0.01	0	0	0.01	0	0.96	0	0	0
H: 7	0	0	0.01	0	0.01	0	0	0.96	0	0.01
I: 8	0	0	0.01	0.03	0.01	0.01	0	0	0.93	0
J: 9	0.01	0	0.01	0	0.01	0	0	0.01	0.01	0.94

(b) MNIST

	A	B	C	D	E	F	G	H	I	J
A: T-shirt/top	0.85	0	0.02	0.03	0.01	0	0.09	0	0.01	0
B: Trouser	0	0.97	0	0.02	0	0	0.01	0	0	0
C: Pullover	0.02	0	0.85	0.01	0.05	0	0.07	0	0	0
D: Dress	0.02	0.01	0.01	0.89	0.02	0	0.04	0	0	0
E: Coat	0	0	0.06	0.04	0.81	0	0.08	0	0	0
F: Sandal	0	0	0	0	0	0.98	0	0.02	0	0.01
G: Shirt	0.13	0	0.06	0.03	0.08	0	0.68	0	0.01	0
H: Sneaker	0	0	0	0	0	0.01	0	0.96	0	0.03
I: Bag	0	0	0	0	0	0	0	0	0.98	0
J: Ankle boot	0	0	0	0	0	0.01	0	0.03	0	0.95

(c) F-MNIST

	A	B	C	D	E	F	G	H	I	J
A: airplane	0.89	0.01	0.02	0.01	0.01	0	0	0	0.04	0.01
B: automobile	0.01	0.93	0	0	0	0	0	0	0.01	0.04
C: bird	0.03	0	0.78	0.05	0.05	0.02	0.05	0.02	0.01	0
D: cat	0.01	0	0.05	0.73	0.03	0.1	0.04	0.02	0.01	0
E: deer	0.01	0	0.03	0.03	0.8	0.02	0.04	0.06	0.01	0
F: dog	0	0	0.02	0.11	0.03	0.8	0.01	0.02	0	0
G: frog	0.01	0	0.02	0.03	0.02	0.01	0.91	0	0	0
H: horse	0.01	0	0.02	0.03	0.04	0.03	0.01	0.86	0	0
I: ship	0.04	0.01	0	0.01	0	0	0.01	0	0.93	0.01
J: truck	0.01	0.06	0	0.01	0	0	0	0	0.02	0.9

(d) CIFAR10-SVM

	A	B	C	D	E	F	G	H	I	J
A: 0	0.98	0	0	0	0	0	0.01	0	0	0
B: 1	0	0.99	0	0	0	0	0	0.01	0	0
C: 2	0.01	0	0.95	0	0	0.01	0	0.01	0.01	0
D: 3	0	0	0.03	0.88	0	0.08	0	0	0	0
E: 4	0	0	0.01	0	0.96	0	0	0.01	0	0.02
F: 5	0	0	0.03	0.02	0	0.93	0.01	0	0	0
G: 6	0.01	0.01	0	0	0.01	0	0.96	0	0	0
H: 7	0	0	0.01	0	0.01	0	0	0.96	0	0.01
I: 8	0	0	0.03	0.04	0	0.01	0	0	0.9	0.01
J: 9	0.01	0	0.01	0.01	0.01	0	0	0.01	0	0.95

(e) MNIST-SVM

	A	B	C	D	E	F	G	H	I	J
A: T-shirt/top	0.85	0	0.02	0.04	0.01	0	0.08	0	0.01	0
B: Trouser	0	0.97	0	0.02	0	0	0.01	0	0	0
C: Pullover	0.02	0	0.82	0.01	0.06	0	0.09	0	0	0
D: Dress	0.02	0.01	0.02	0.89	0.02	0	0.04	0	0	0
E: Coat	0.01	0	0.06	0.05	0.78	0	0.1	0	0	0
F: Sandal	0	0	0	0	0	0.96	0	0.03	0	0.01
G: Shirt	0.13	0	0.07	0.04	0.09	0	0.65	0	0.02	0
H: Sneaker	0	0	0	0	0	0.01	0	0.97	0	0.02
I: Bag	0.01	0	0	0	0	0	0	0	0.98	0
J: Ankle boot	0	0	0	0	0	0.01	0	0.04	0	0.95

(f) F-MNIST-SVM

Fig. 2: Visualization of confusion matrix of ActiveGAN+F (top) and SVM (bottom) for each dataset, CIFAR10, MNIST and Fashion-MNIST (F-MNIST). For ActiveGAN+F. The number of images used in the training for (a)(d), (b)(e) and (c)(f) are 10*k*, 1*k* and 10*k*, respectively. The thresholds for the margin constraint in (a), (b) and (c) are 0.2, 0.15, and 0.15, respectively.

images in the original training data, do not provide more information than the original training images already could.

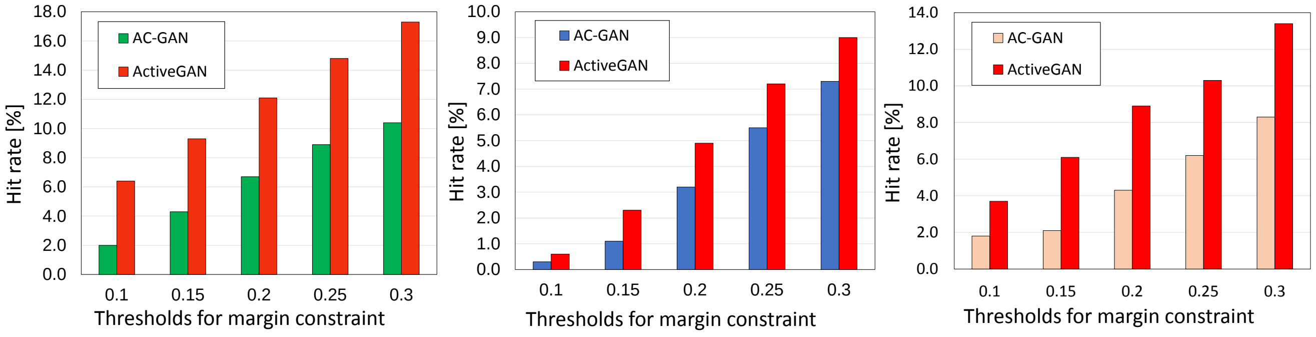
Interestingly, using the generated samples from ActiveGAN is able to achieve similar or even slightly better performance than the baseline SVM. For example, in MNIST, ActiveGAN is able to achieve slightly better performance than the case SVM by 0.3%. If the margin constraint is applied, the generated samples with the same distribution as the training images might be filtered out, only keeping informative samples that are considered to contribute to the learning performance. In such cases, ActiveGAN+F achieves a better performance than the base SVM for all datasets. For example, in both CIFAR10 evaluations, ActiveGAN+F reaches F-scores of 84.5% and 86.8% in different settings, which are improvements of 1.1% and 1.5% compared to the base SVM, respectively.

Due to the multi-class problem in the image classification task, we also show how the F-score of each category changes compared to the baseline SVM if ActiveGAN+F is used. Figure 2 shows the confusion matrix for each dataset. We can observe that the F-score of each category is evenly improved for MNIST. In CIFAR10, except the class 'automobile', the recall of all classes is improved. Among the improvements, the recall of the class label 'cat' is improved by 3%. In MNIST, the recall of the class '2', '3', '4', and '8' is improved. The recall of the class label '3' is improved by 4%.

5.3 Discussion and Analysis

First, we examine the ratio of informative images to all generated images. Since it is difficult to claim if a generated image is informative, we simply use the margin constraint to filter out images that are not treated as informative ones by setting the margin between the probabilities of the two most probable classes. The smaller value the margin is, the more uncertain the generated image is. Figure 3 shows the ratios of ActiveGAN and AC-GAN for each dataset. It can be seen that the ratio in ActiveGAN is always higher than that in AC-GAN at every threshold in each dataset. For example, in CIFAR10, the ratio is about 2% to 7% higher. In MNIST, the ratio is about 0.3% to 2% higher. It explains that ActiveGAN is more likely to generate informative images than AC-GAN, due to the objective function in ActiveGAN having a term to measure the degree of *uncertainty*. The informative generated images tend to contribute in improving the classification performance. But we empirically found it does not mean that the more informative the sample is, the more the classification performance is improved. In the image classification task, the best classification performance is not always achieved at the margin of 0.1. It might be because few samples are kept under a strong margin constraint, which is incapable of improving the classification performance. Formulated differently, not only informative samples are required to achieve good performance.

Second, to verify that ActiveGAN is more likely to generate informative images than AC-GAN, we visualize the features of

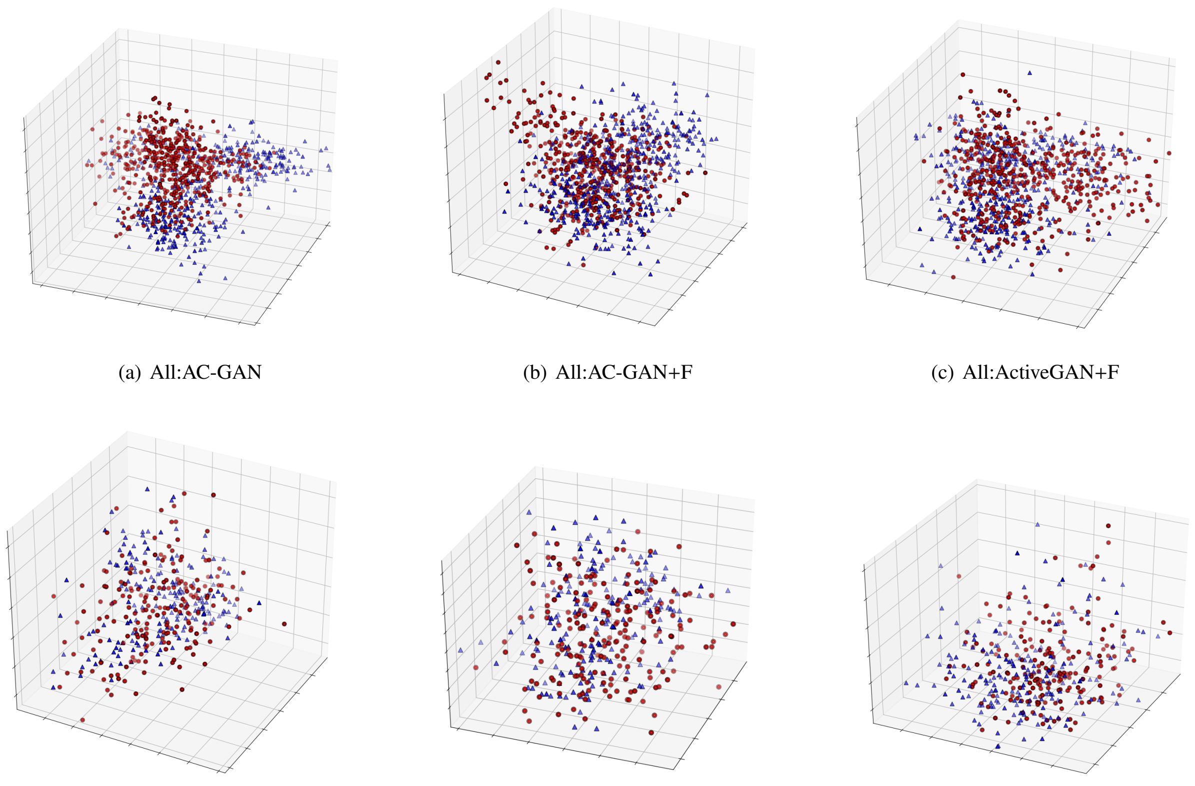


(a) CIFAR10

(b) MNIST

(c) F-MNIST

Fig. 3: Ratio of informative images to all training images at every margin threshold for each dataset.



(a) All:AC-GAN

(b) All:AC-GAN+F

(c) All:ActiveGAN+F

(d) Cat:AC-GAN

(e) Cat:AC-GAN+F

(f) Cat:ActiveGAN+F

Fig. 4: Visualization of features of training images and generated images. Blue points represent original training images and red points represent generated images. Top row: (a), (b) and (c) represent visualization for AC-GAN, AC-GAN+F, and ActiveGAN+F for all class labels, respectively. Bottom row: (d), (e) and (f) represent visualization for AC-GAN, AC-GAN+F and ActiveGAN+F for the class label 'cat', respectively. To ease visualization, we randomly choose 100 images from training images and generated images for each class.

training images and the generated images in CIFAR10 in a 3-dimensional space, as shown in Figure 4. The 3-dimensional space is obtained by principal component analysis (PCA). Let $\mathbf{c}_{\text{TRAIN}}$ denote the centriod of training images, and let $\mathbf{c}_{\text{AC-GAN}}$ denote the centroids of images from AC-GAN. Let $\mathbf{c}_{\text{AC-GAN+F}}$ and $\mathbf{c}_{\text{ActiveGAN+F}}$ denote the centroids of images from AC-GAN+F and ActiveGAN+F applied by the margin constraint, respectively. We calculated Euclidean distances for features of training images and generated images, as shown in Table 2. Two cases are shown, including all class labels and a specific class label. For a specific class label, we take the class label 'cat' as an example, since classification performance of this label is improved most after using ActiveGAN+F. The columns of 'centriod' represent the distances between centroids of features of generated images in each model and those of features in the training images. The columns of 'mean' and 'std' represent averaged distances and their standard deviations for each model. These distances are defined as ones between the centroids of original images and image features from each model. For cases of both all class labels and class label 'cat', $\mathbf{c}_{\text{ActiveGAN+F}}$ is higher than those of $\mathbf{c}_{\text{AC-GAN}}$ and $\mathbf{c}_{\text{AC-GAN+F}}$. This implicitly explains that the generated images from ActiveGAN+F are located slightly farther from training images than those from

Table 2: Statistics on Euclidean distances for features of training images and generated images.

	AC-GAN			AC-GAN+F			ActiveGAN+F		
Class label	centriod	mean	std	centriod	mean	std	centriod	mean	std
All labels	21.05	75.42	6.81	18.84	85.99	10.93	31.43	106.70	7.70
‘Cat’	8.59	49.30	8.44	11.07	58.09	6.39	13.43	69.77	7.49

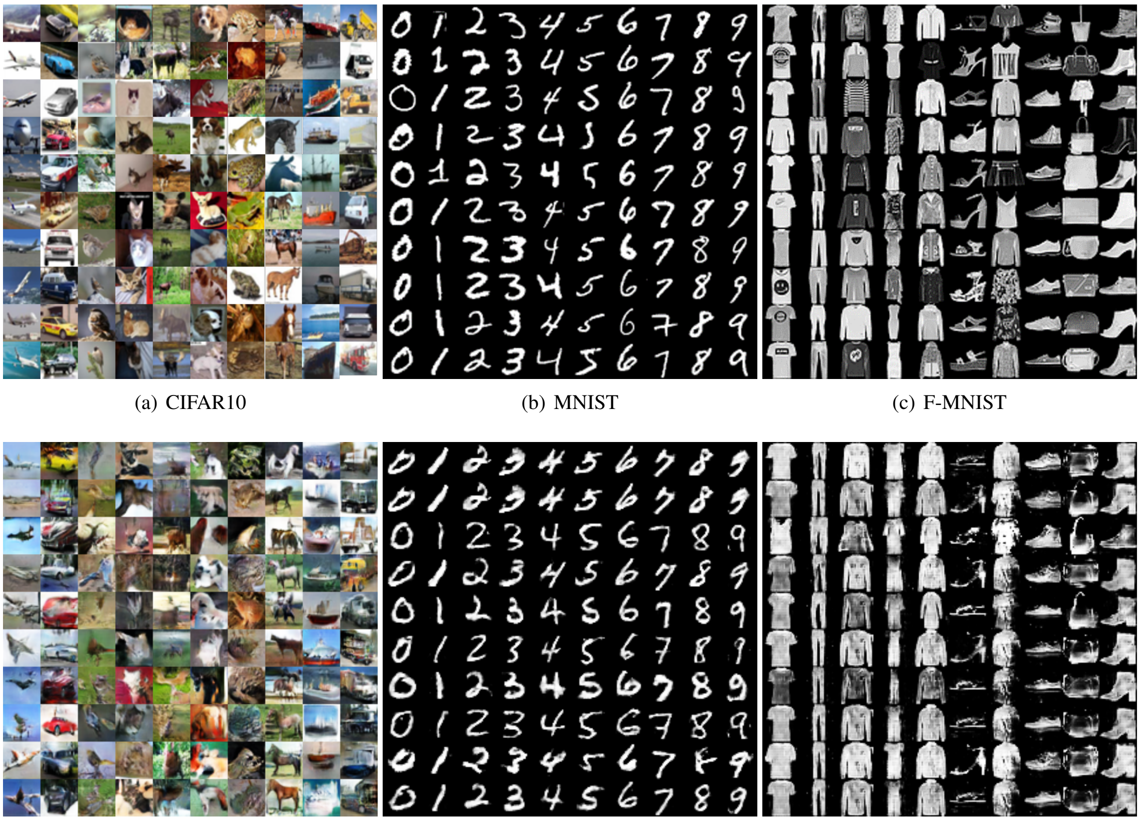

(a) CIFAR10
(b) MNIST
(c) F-MNIST
(d) CIFAR10-ours
(e) MNIST-ours
(f) F-MNIST-ours

Fig. 5: Samples of generated images (a), (b) and (c) are images randomly sampled from training images. (d), (e) and (f) are images generated from ActiveGAN. Each row shares same label and each column shares the same latent variables.

Table 3: Ablation study when α changed in Eq. (8) for n=10k.

Setting of α	$\alpha = 0$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 1.0$
F-score on CIFAR10	86.4	86.7	86.8	86.2	85.7

AC-GAN and AC-GAN+F.

Third, we show sampled images from original training images and ActiveGAN, as depicted in Figure 5. The generated images in ActiveGAN are the ones that satisfy the margin constraint. For MNIST and Fashion-MNIST, ActiveGAN is able to generate reasonably good quality of images. Due to more complex images in CIFAR10, the image generation for this dataset is more difficult than MNIST and Fashion-MNIST. The first columns of Figure 5(a) and Figure 5(d) represent images of class label ‘airplane’. We can see that some images in Figure 5(d) look like ‘bird’, which might serve as informative images to discriminate the classes of ‘airplane’ and ‘bird’.

Forth, to show our smallest margin and label entropy reward both helpful on generating informative samples. We choose CIFAR10 as an example, the performance of which is shown in Table 3 when the value of α in Equation (8) changes. When $\alpha = 0.0$, only label entropy was used as our reward, that outperforms about 3.2% compared with AC-GAN+F. When we only use smallest margin as our reward ($\alpha = 1.0$), the performance is boosted by 2.5%. When $\alpha = 0.5$, that we both use two kinds of rewards, we achieved the best performance, which is higher than $\alpha = 0.0$ and $\alpha = 1.0$ about 0.4% and 1.1%. It implies that both factors of smallest margin and label entropy are helpful.

6. Conclusion

In this paper, we investigate the problem of *lack of labeled data*, in which labels of data can be obtained without using any additional unlabeled data nor querying the human oracle. This is achieved in a cheaper manner than traditional active learning. In our proposed model, we use class-conditional generative adversarial networks (GANs) to generate images, and devise a novel reward related to the degree of *uncertainty* for generated samples. This reward is used to guide the class-conditional GAN to generate informative samples with a higher probability. Our empirical results on CIFAR10, MNIST and Fashion-MNIST demonstrate that our proposed model is able to generate informative labeled images that are confirmed to be effective in improving classification performance.

References

[1] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D. and Krishnan, D.: Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 95–104 (2017).

[2] Cebron, N. and Berthold, M. R.: Active learning for object classification: from exploration to exploitation, *Data Mining and Knowledge Discovery*, Vol. 18, No. 2, pp. 283–299 (2009).

[3] Chen, X., Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. and Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2172–2180 (2016).

[4] Choe, J., Park, S., Kim, K., Park, J. H., Kim, D. and Shim, H.: Face Generation for Low-Shot Learning Using Generative Adversarial Networks, *IEEE International Conference on Computer Vision Workshop (ICCVW)* (2017).

[5] Denton, E., Chintala, S., Szlam, A. and Fergus, R.: Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1486–1494 (2015).

[6] Donmez, P., Carbonell, J. G. and Bennett, P. N.: Dual Strategy Active Learning, *European Conference on Machine Learning (ECML)* (Kok, J. N., Koronacki, J., Mantaras, R. L. d., Matwin, S., Mladenič, D. and Skowron, A., eds.), pp. 116–127 (2007).

[7] Freund, Y., Seung, H. S., Shamir, E. and Tishby, N.: Selective Sampling Using the Query by Committee Algorithm, *Machine Learning*, Vol. 28, No. 2, pp. 133–168 (online), DOI: 10.1023/A:1007330508534 (1997).

[8] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A. and Mikolov, T.: DeViSE: A Deep Visual-Semantic Embedding Model, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2121–2129 (2013).

[9] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C. and Bengio, Y.: Generative Adversarial Nets, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672–2680 (2014).

[10] Huijser, M. W. and van Gemert, J. C.: Active Decision Boundary Annotation with Deep Generative Models, *International Conference on Computer Vision (ICCV)*, pp. 5296–5305 (2017).

[11] Jain, P. and Kapoor, A.: Active Learning for Large Multi-Class Problems, *Computer Vision and Pattern Recognition (CVPR)* (2009).

[12] Krizhevsky, A., Nair, V. and Hinton, G.: CIFAR-10 (Canadian Institute for Advanced Research), (online), available from ⟨http://www.cs.toronto.edu/ kriz/cifar.html⟩.

[13] Lampert, C. H., Nickisch, H. and Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer, *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 951–958 (2009).

[14] Li, J., Monroe, W., Shi, T., Ritter, A. and Jurafsky, D.: An Embarrassingly Simple Approach to Zero-shot Learning, *Empirical Methods on Natural Language Processing (EMNLP)* (2017).

[15] Mirza, M. and Osindero, S.: Conditional Generative Adversarial Nets, *CoRR*, Vol. abs/1411.1784 (online), available from ⟨http://arxiv.org/abs/1411.1784⟩ (2014).

[16] Moskovitch, R., Nissim, N., Stopel, D., Feher, C., Englert, R. and Elovici, Y.: Improving the Detection of Unknown Computer Worms Activity Using Active Learning, *Advances in Artificial Intelligence*, pp. 489–493 (2007).

[17] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A. Y.: Reading Digits in Natural Images with Unsupervised Feature Learning, *Workshop on Deep Learning and Unsupervised Feature Learning - Advances in Neural Information Processing Systems (NIPS)* (2011).

[18] Odena, A., Olah, C. and Shlens, J.: Conditional Image Synthesis with Auxiliary Classifier GANs, *International Conference on Machine Learning (ICML)*, pp. 2642–2651 (2017).

[19] Pan, S. and Yang, Q.: A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359 (2010).

[20] Roy, N. and Mccallum, A.: Toward Optimal Active Learning through Sampling Estimation of Error Reduction, *International Conference on Machine Learning (ICML)*, pp. 441–448 (2001).

[21] Settles, B.: Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009).

[22] Settles, B.: Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009).

[23] Shen, D., Zhang, J., Su, J., Zhou, G. and Tan, C.-L.: Multi-Criteria-based Active Learning for Named Entity Recognition, *Meeting of the Association for Computational Linguistics (ACL)*, pp. 589–596 (2004).

[24] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *CoRR*, Vol. abs/1409.1556 (2014).

[25] Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y.: Policy Gradient Methods for Reinforcement Learning with Function Approximation, *Advances in Neural Information Processing Systems (NIPS)*, pp. 1057–1063 (2000).

[26] Tong, B., Klinkigt, M., Chen, J., Cui, X., Kong, Q., Murakami, T. and Kobayashi, Y.: Adversarial Zero-shot Learning with Semantic Augmentation, *Association for the Advancement of Artificial Intelligence (AAAI)* (2018).

[27] Wang, W., Pu, Y., Verma, V. K., Fan, K., Zhang, Y., Chen, C., Rai, P. and Carin, L.: Zero-Shot Learning via Class-Conditioned Deep Generative Models, *Association for the Advancement of Artificial Intelligence (AAAI)* (2018).

[28] Xiao, H., Rasul, K. and Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017).

[29] Zhu, J. and Bento, J.: Generative Adversarial Active Learning, *CoRR*, Vol. abs/1702.07956 (2017).