Enhancing Classification Accuracy on Limited Data via Unconditional GAN

Chunsan Hong*
KAIST
Daejeon, South Korea
hoarer@kaist.ac.kr

Byunghee Cha*
Seoul National University
Seoul, South Korea
paulcha1025@snu.ac.kr

Bohyung Kim
CNAI
Seoul, South Korea
giantbalance@snu.ac.kr

Tae-Hyun Oh†
Dept. Electrical Eng. & GSAI, POSTECH, Pohang, South Korea
Institute for Convergence Research and Edu. in Adv. Tech., Yonsei Univ., Seoul, South Korea.
taehyun@postech.ac.kr

Abstract

Despite significant advances in Deep Neural Networks (DNNs), these models often fall short in real-world scenarios, particularly when faced with a scarcity of training data. In this paper, we introduce a novel method that capitalizes on the power of Generative Adversarial Networks (GANs) to enhance performance in image classification tasks. Our approach specifically involves training the classifier by enforcing a consistency rule across generated unlabeled data synthesized from unconditional GANs. Through the implementation of our proposed methodology, we observed a substantial increase in accuracy - approximately 8.68% on the CIFAR-10 dataset compared to the baseline (which had an accuracy of 54.54%) trained with 500 real images. This notable enhancement in accuracy demonstrates the superiority of our method using class unconditional GANs over the previous techniques aiming to enhance accuracy using class Conditional GANs.

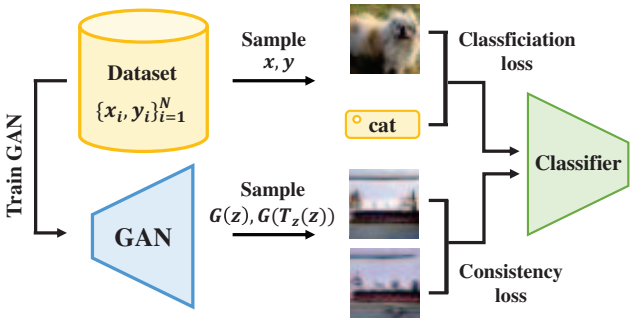


Figure 1: Illustration of our method. Initially, the GAN is trained from the Dataset, then keeps frozen. The classifier is then trained using two types of losses: 1) The classification loss, which is obtained by sampling an image x and its corresponding label y from the Dataset. 2) The consistency loss between the pair of images generated from two latents, where a latent z is sampled from the latent space and a transformed latent $T_z(z)$ is derived.

1. Introduction

Deep Neural Networks (DNNs) have achieved significant success across various domains. However, these models are data-hungry, often requiring large volumes of training data for optimal performance. With limited data, DNNs may excessively rely on specific features in the training data, *i.e.* overfitting, leading to diminished performance [40, 20]. The model fails to accurately predict data that fall slightly outside the distribution of the training set. To mitigate these problems, techniques such as dropout [35], data augmentation [5], and semi-supervised learning [39, 33] are employed.

Leveraging generative models such as Generative Adversarial Networks (GANs) to address these issues would be a quick remedy, but there is a scarce number of research that has shown successful outcomes with GANs in classification tasks. The work [11, 27, 19] reports that there barely exists prior work that achieves improvement in classification accuracy by utilizing GANs. Ealier works [32, 29, 30] conducted experiments to enhance classification accuracy

*These authors contributed equally to this work.
†Corresponding author.

Acknowledgment. T.-H. Oh is partially supported by IITP grant funded by the Korea government(MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network; No. 2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)).

by using GANs but reported only negligible improvements. Conventional approaches [25, 42] managed to enhance accuracy using GANs but their methodologies lack scalability.

In contrast, the potential of representations learned from GANs to aid classification tasks is immense. Numerous papers [3, 4, 31, 22, 38, 13, 8, 24, 36, 17, 41, 37, 14] have reported successful outcomes using GANs in various tasks. Furthermore, several studies [28, 7, 9] suggest that representations learned from GANs could benefit DNNs. Hence, we aim to utilize the representations learned by GANs for classification tasks.

Our proposed technique to improve classifier performance using GANs focuses on two key points: 1) Using class-unconditional GANs, and 2) Developing the learning strategy leveraging generated unlabeled images without altering the GANs training process. The employment of class-unconditional GANs is motivated by the potential risk of generated data falling into the convex hull of classes with class-conditional GANs, as Kong *et al*. [19] pointed out. Employing class-unconditional GANs may facilitate the generation of samples situated closer to the decision boundary, effectively filling the sparse region between the dense clusters of the training data that correspond to classes. Moreover, even if there are generated samples from class-conditional GAN that are near the classifier's decision boundary, it remains uncertain whether these samples have been correctly created corresponding to the conditioned class, *i.e*. the label of the generated sample can be wrong. We provide empirical evidence of the superiority of our technique over traditional methods using conditional GANs, demonstrating its effectiveness in enhancing classification accuracy.

Furthermore, the reason we do not manipulate the GANs training process is that the application of additional techniques to the GANs can easily destabilize its training, and there exists no relationship between the performance of the GANs and that of the classifier. Ravuri *et al*. [30] has pointed out that generated images from GAN with higher Fréchet Inception Distance (FID) do not necessarily lead to enhanced accuracy. Dai *et al*. [6] argue that utilizing relatively underperforming GANs, i.e. GANs that generate samples deviating from the training data distribution, is rather beneficial to enhance the performance of DNNs. They further substantiate this claim by providing mathematical proof. Hence, enhancing the performance of GANs is not our objective.

The images generated via an unconditional GAN lack class labels, necessitating learning methods other than conventional supervised learning. In the representation learning scenario, Jahanian *et al*. [14] provides the method which leverages only generated data from class-unconditional GANs (which will be referred as GenRep in the paper), successfully managing to achieve the performance of SimCLR [2] using real data. GenRep trains the model by backpropagating the contrastive loss between pairs of images generated from latent codes and from their perturbed counterparts. Based on the success of GenRep, we expand the idea to classification tasks. We devise a technique utilizing the classification loss of real labeled data and the consistency loss of unlabeled generated data pair. We use the consistency loss of FlexMatch [39] for our proposed method. Consistency loss of FlexMatch undergoes a thresholding procedure, which effectively inhibits the influence of low-quality generated data on the learning process. This in turn stabilizes the training and contributes to higher accuracy. We provide empirical evidence for these assertions. To the best of our knowledge, we are the first to employ a class-unconditional GAN for the purpose of enhancing classification accuracy. This novel approach takes us one step closer to achieving our goal of enhancing classifier performance with generated data.

We summarize our contributions as follows:

- We successfully improve the accuracy of the classifier in a limited data setting using GANs.
- We propose a novel method to utilize unlabeled generated images from class-unconditional GANs based on the consistency rule.
- We demonstrate that utilizing class-unconditional GANs may offer superior performance enhancement for classifiers compared to employing class-conditional GANs.

2. Related Work

There exists a large number of studies demonstrating the effectiveness of GANs in improving the performance of DNNs. These include applications in Semantic Segmentation [3, 31, 22], Human Pose Estimation [13, 38], Optical Flow [36, 8, 24], and Representation Learning [14]. Recent studies [41, 37] propose methods to enhance the performance of semantic segmentation by utilizing StyleGAN [17]. GenRep [14], the foundational work for our study, successfully leverages StyleGAN2 [18] in representation learning. We use StyleGAN2-ADA [16] for our base model, which was designed for limited data scenarios that align with our experimental setting.

While there have been attempts to leverage generated images to enhance the performance of classifiers, most of these methods are fundamentally archaic and lack scalability. Zhu *et al*. [42] improves classification accuracy by supplementing rare classes of the imbalanced dataset. CycleGAN trained to generate rare classes from sufficient classes is used for supplementing. On the other hand, this method is limited to imbalanced datasets. Mun *et al*. [25] proposed a technique to sequentially select generated images closer

to the SVM hyperplane. Nonetheless, this approach is confined to SVMs, necessitates a validation set, fails to utilize the entire training dataset, and possesses a complex training process that renders it non-scalable.

Two recent papers [27, 19] utilizing class-conditional GANs have conducted experiments in a practical setting. As noted in Sec. 1, the direct application of class-conditional GANs risks falling into the convex hull. Therefore, both papers aim to resolve it by incorporating uncertainty into the GAN's loss function, which results in generated samples being closer to the decision boundary. Uncertainty is defined in terms of Margin [15] or Confidence [23] calculated by the classifier output of generated images. Since the methods possess numerous hyperparameters related to uncertainty, improving classifier performance considering the stability of the GANs training process needs a lot of effort. Contrary to these two approaches that train the class-conditional GAN to generate images closer to the decision boundary, our method employs a class-unconditional GAN, which naturally generates a large number of samples closer to the decision boundary without additional training. This allows GANs to be trained independently of the classifier, which makes the training process to be stable. We opt to utilize work [27] as our benchmark, given that work [19] relies on a Support Vector Machine (SVM) for its classifier, which lacks scalability and is not suitable for a fair comparison.

Recently, with the surge of interest in diffusion models [12, 34], research [1] that enhances classification performance through these models has been reported. Nevertheless, the method is unsuitable for classification tasks of specific domains, since the method is based on pre-trained diffusion models. Furthermore, training diffusion models in a limited data setting is inappropriate since the generated samples tend to overfit the training data. We empirically validate this phenomenon in the experiment section. Hence, we opt for StyleGAN2-ADA over the diffusion model.

3. Method

3.1. Problem definition

The primary objective of our study is to improve the performance of the classifier C in a limited labeled data $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$, using class unconditional GAN. Generator model G is initially trained on $\mathcal{D}$ without label information, *i.e.* only utilizes $\{\boldsymbol{x}_i\}_{i=1}^{N}$ as train data. After initial training, the generator G is not trained further. We denote the unlabeled generated data created by G as $G(\boldsymbol{z})$, where $\boldsymbol{z}$ is a variable drawn from a standard normal distribution $\mathcal{N}(0, \sigma)$. Our goal is to train the classifier C using the following two losses: 1) the classification loss given by the pair $\boldsymbol{x}, \boldsymbol{y}$ sampled from $\mathcal{D}$, and 2) the consistency loss between $G(\boldsymbol{z})$ and $G(T_z(\boldsymbol{z}))$ sampled from G, where T_z is latent transformation.

3.2. Background

GenRep For contrastive learning, the infoNCE loss [26] is commonly used, which is defined as follows:

$$\mathcal{L}_{NCE} = -\,\mathbb{E}\left[\log \frac{e^{\tau F(\boldsymbol{x}_a)^T F(\boldsymbol{x}_p)}}{e^{\tau F(\boldsymbol{x}_a)^T F(\boldsymbol{x}_p)} + \sum_{k=1}^{K} e^{\tau F(\boldsymbol{x}_a)^T F(\boldsymbol{x}_n^K)}}\right]$$

where $\boldsymbol{x}_a$ is an anchor image, $F(\boldsymbol{x})$ is the model output of the input image $\boldsymbol{x}$, $\{\boldsymbol{x}_a, \boldsymbol{x}_p\}$ is a positive pair of images that should be entangled in feature space, and $\{\boldsymbol{x}_a, \boldsymbol{x}_n^K\}$ is a negative pair of images that should be disentangled in feature space. SimCLR [2], an outstanding self-supervised learning technique, generates a positive pair from a batch of images through the application of two separate pixel-wise transformations to each image in the batch. The rest of the data in the batch, barring the positive pair, is treated as negatives.

GenRep is a technique designed to effectively perform representation learning using only images generated from a pre-trained generator, without the need for real data. GenRep employs the infoNCE loss as well. Remarkably, it achieves comparable performance to SimCLR solely leveraging synthesized data, negating the need for real data samples. To elucidate further, GenRep treats a certain latent code $\boldsymbol{z}$ and a transformed version of the same latent code $T_z(\boldsymbol{z})$ as a positive pair. In contrast, images generated from a different latent code are treated as a negative pair. This method allows GenRep to leverage the infoNCE loss effectively.

GenRep successfully achieves high performance in representation learning scenarios without real data, utilizing latent transformation T_z. We adapt the concept into the limited data scenario, that image pairs generated from latent z and transformed latent $T_z(\boldsymbol{z})$ should be entangled in the feature space. For T_z, we use small Gaussian perturbations where GenRep reported that among various methods, Gaussian perturbation performed the best.

FlexMatch loss FlexMatch [39] is one of the techniques employed in semi-supervised learning, a task that aims to train a neural network using both labeled data $\mathcal{D}$ and unlabeled data $\mathcal{U} = \{\boldsymbol{u}_i\}_{i=1}^{M}$. The typical loss function of semi-supervised learning is defined as follows:

$$q(\boldsymbol{x}) = p_C(\boldsymbol{y}|\boldsymbol{x}),$$

$$\hat{q}_i(\boldsymbol{x}) = \begin{cases} 1 & \text{if } i = \arg\max q(\boldsymbol{x}) \\ 0 & \text{otherwise} \end{cases},$$

$$\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathcal{I}(q(\boldsymbol{u}_b)) \cdot \mathbb{H}\big[\hat{q}(\boldsymbol{u}_b), q(T_x(\boldsymbol{u}_b))\big]. \quad (1)$$

where $p_C(\boldsymbol{y}|\boldsymbol{x})$ is output of classifer C, and $\mathcal{I}(\cdot)$ is indicator function. To clarify the notation, $q(\boldsymbol{x})$ is the prediction of the classifier for input $\boldsymbol{x}$, and $\hat{q}(\boldsymbol{x})$ is the pseudo label for the prediction of input $\boldsymbol{x}$. The main idea of the loss is to regulate the model to make the same prediction of $T_x(\boldsymbol{u})$ as that of $\boldsymbol{u}$, based on the theorem that label information must be maintained. The indicator function $\mathcal{I}(q(\boldsymbol{u}))$ dictates whether to include a sample $\boldsymbol{u}$ in the learning process based on its corresponding output of the classifier. A common strategy is to apply a thresholding mechanism. The concept behind the threshold indicator function is based on the idea that the model should only pseudo-label and learn from the unlabeled data when it is sufficiently confident about it. For instance, the renowned FixMatch approach sets the value of $\mathcal{I}(\cdot)$ to 1 when the maximum confidence value surpasses a 0.95 threshold, and to 0 otherwise. FlexMatch modifies this method slightly, as follows:

$$\mathcal{I}(\boldsymbol{u}) = \mathbb{1}(\max(q(\boldsymbol{u})) > \mathcal{T}(\arg\max(q(\boldsymbol{u}))) \cdot \tau), \quad (2)$$

$$\mathcal{T}(c) = \frac{\beta(c)}{\max_c \beta},$$

$$\beta(c) = \sum_{i=1}^{M} \mathbb{1}(\max(q(\boldsymbol{u}_i)) > \tau \wedge \arg\max(q(\boldsymbol{u}_i)) = c).$$

The idea of FlexMatch stems from the limitations of the FixMatch technique, which applies the same threshold to samples from all classes. Considering that the difficulty can vary among classes, FlexMatch introduces the key idea of applying different thresholds to each class.

The distinguishing aspect between semi-supervised learning and our scenario is that unlabeled data $\{\boldsymbol{u}_i\}_{i=1}^{M}$ is replaced by generated data from G. Our choice of using FlexMatch as a consistency loss function is not only because it currently holds state-of-the-art performance in semi-supervised learning, but also due to its beneficial attributes in our scenario that we found experimentally. While training with generated data tends to be considerably more unstable, the thresholding approach of FlexMatch enables stable learning. We present experiments using various threshold indicator functions including FixMatch in Sec. 4.4.

3.3. Consistency loss for generated unlabeled data

Building upon the methodology presented, we introduce a consistency loss for the generated unlabeled data, inspired by GenRep [14] and FlexMatch [39]. While GenRep optimizes infoNCE loss between original and latent-transformed image, our method optimizes FlexMatch loss between those images. Integrating G and latent transformation into Flexmatch loss (Eq. 1 and Eq. 2), the formulation is given by:

$$\mathcal{L}_z = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathcal{I}(G(\boldsymbol{z}_i)) \cdot \mathbb{H}\big[\hat{q}(G(\boldsymbol{z}_i)), q(G(T_z(\boldsymbol{z}_i)))\big], \quad (3)$$

where each $\boldsymbol{z}_i$ is the latent code sampled from $\mathcal{N}(0, \sigma^2)$, and $T_z(\boldsymbol{z}_i)$ is also the latent code sampled from $\mathcal{N}(\boldsymbol{z}_i, \epsilon \cdot \sigma^2)$. One additional consideration is necessary when applying the FlexMatch technique to generated data. In the case of FlexMatch, all the predictions of the classifier for unlabeled data $\{\boldsymbol{u}_i\}_{i=1}^{M}$ are tracked when calculating $\beta(c)$. While in our approach, the unlabeled data is replaced with samples generated by G, and the count of these generated samples is infinite, making it impractical to track all of them. To address this issue, we compute $\beta(c)$ based on samples generated during the recent training batches. Finally, we can formulate the loss of our method as the weighted combination of supervised and consistency loss:

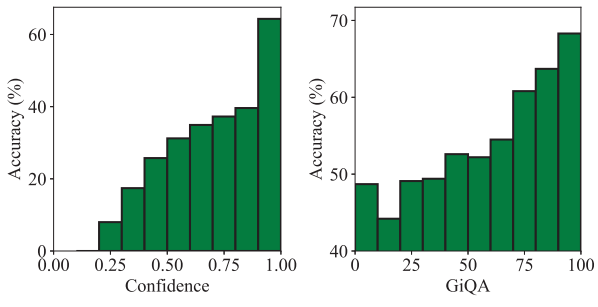
$$\mathcal{L}_{GenMatch} = \mathcal{L}_s + \lambda \mathcal{L}_z, \quad (4)$$

where $\mathcal{L}_s$ is the supervised loss on labeled data:

$$\mathcal{L}_s = \frac{1}{B} \sum_{i=1}^{B} \mathbb{H}[\boldsymbol{y}_i, q(\boldsymbol{x}_i)]. \quad (5)$$

The Eq.4 implies that our methodology contemplates the cross-entropy loss of real labeled data and the consistency loss of the unlabeled generated data which would result in entanglement of generated data in the classification space of C. The threshold indicator function in FlexMatch serves a dual purpose: it assists in filtering out the generated data that would contribute positively towards stabilizing the classifier, while also helping to even out the class distribution of the generated data being trained. This approach thereby harnesses the representations learned by the generator. The expected outcomes include an enhancement in model generalization and stability. We designate this method as GenMatch.

We experimentally confirmed that the performance of the classifier trained with GenMatch loss (Eq. 4) surpasses the performance of the classifier trained with traditional classification loss (Eq. 5). Despite this, Eq. 4 presents an area that needs refinement: there's no inherent justification for the model's prediction on $G(T_z(\boldsymbol{z}_i))$ to follow the pseudo-label of $G(\boldsymbol{z}_i)$. In Flexmatch, the prediction for the hard perturbed sample is guided to follow the original sample's prediction, as there is a higher probability that the original sample's prediction is accurate. Although the possibility of $T_z(\boldsymbol{z})$ being out-of-distribution compared to z increases due to its broader latent space ($p_{T_z(\boldsymbol{z})} = \mathcal{N}(0, (1+\epsilon) \cdot \sigma^2)$ compared to $p_z = \mathcal{N}(0, \sigma^2)$), it remains uncertain whether the sampled $\boldsymbol{z}$ would yield better classifier predictions than



(a) (b)

Figure 2: This plot illustrates the correlation between confidence, GiQA, and accuracy. The analysis made use of a WideResnet-28 model, which was trained on 500 actual samples from the CIFAR-10 training dataset, and leveraged samples from the CIFAR-10 test dataset. In Figure (a), confidence intervals are divided into increments of 0.1, and the corresponding accuracy of the test samples within each interval is calculated. In Figure (b), GiQA intervals are divided into deciles based on percentiles, and the corresponding accuracy for the test samples within each decile is again calculated.

$T_z(\boldsymbol{z})$. In other words, it is uncertain whether encouraging the prediction of the classifier for $T_z(\boldsymbol{z})$ to conform to the prediction for $\boldsymbol{z}$ would positively influence the performance improvement of the classifier.

To address this uncertainty, we define the samples in the classification space of the classifier that are more in-distribution as $\boldsymbol{z}_e$, and the out-of-distribution samples as $\boldsymbol{z}_h$. The purpose is to train a classifier regulating the prediction of $\boldsymbol{z}_h$ to follow that of $\boldsymbol{z}_e$.

$$\boldsymbol{z}_h = \begin{cases} \boldsymbol{z}, & \text{if } \xi(G(\boldsymbol{z})) > \xi(G(T_z(\boldsymbol{z})) \\ T_z(\boldsymbol{z}), & \text{otherwise} \end{cases},$$

$$\boldsymbol{z}_e = \begin{cases} T_z(\boldsymbol{z}), & \text{if } \xi(G(\boldsymbol{z})) > \xi(G(T_z(\boldsymbol{z}))) \\ \boldsymbol{z}, & \text{otherwise} \end{cases},$$

$$\mathcal{L}_z = \frac{1}{\mu B} \sum_{i=1}^{\mu B} \mathcal{I}(G(\boldsymbol{z}_{i,e})) \cdot \mathbb{H}\big[\hat{q}(G(\boldsymbol{z}_{i,e})), q(G(\boldsymbol{z}_{i,h}))\big]. \tag{6}$$

For the uncertainty measures ξ we consider using, it is required that they contain information about whether the given sample is in-distribution, and that lower uncertainty indicates the classifier making more accurate predictions for the given sample. Considering these two conditions, the chosen measures of uncertainty are the maximum-confidence value and the Generated Image Quality Assessment (GIQA) [10]. The maximum confidence value and the GIQA can be formulated as:

$$\xi_{conf}(x) = -\max(q(\boldsymbol{x})), \tag{7}$$

$$\xi_{giqa}(x) = -p(\boldsymbol{x}|\lambda^*) = -\sum_{i=1}^{M} w^i g(f_C(\boldsymbol{x})|\mu^i, \Sigma^i), \tag{8}$$

where $f_C(\boldsymbol{x})$ denotes the features derived from the classifier model C, $\lambda = \{w^i, \mu^i, \Sigma^i\}_{i=1}^M$ and $\lambda^* = \arg\max_\lambda \mathbb{E}_{\boldsymbol{x}\sim p(D)} \log p(f_C(\boldsymbol{x})|\lambda)$. GIQA is a metric designed to evaluate how well a generated image adheres to the training data distribution. It is known for its robust calibration performance. While GIQA typically employs a pre-trained network as the feature extractor in Eq. 8 for assessing generated image quality, we utilize f_C to evaluate the extent to which the generated image is in-distribution in the classification space with respect to the original training data. Despite its usefulness, GMM modeling is a time-consuming process that can potentially lengthen training times. To counteract this, we opt to perform GMM modeling once every 100 iterations. We denote the formula obtained by substituting confidence (Eq. 7) into Eq. 6 as GenMatch-Conf, and the formula obtained by substituting GIQA (Eq. 8) into Eq. 6 as GenMatch-GIQA. As shown in Fig. 2, it is clear that samples with higher levels of confidence or GIQA values tend to be more accurate. This insight has guided our choice of these two metrics for further exploration. Indeed, GenMatch-Conf and GenMatch-GIQA function as we intended, providing improved accuracy compared to GenMatch, which we elaborate on in Sec. 4.

4. Experiments

4.1. Dataset and Training Details

We conduct our experiments on the CIFAR-10 dataset, which includes 32×32 images corresponding to 10 classes and is comprised of 50,000 training data images and 10,000 test data images. For the limited data scenario, we randomly sample just 500 images from the training data for the experiment, and the test is conducted on the entire test data.

We use StyleGAN2-ADA as the generator G and WideResnet-28 as the classifier C. The learning rates for the generator and discriminator were carefully set at specific values, with a batch size chosen to maintain a balance between memory usage and the performance of the model. Another critical parameter, the Ada target, is adjusted to attain an equilibrium between image diversity and image quality. The StyleGAN2-ADA is trained with the logistic adversarial loss. We use Adam for optimization, with a learning rate of 0.0025 for both generator and discriminator, and settings of $\beta_1 = 0$ and $\beta_2 = 0.99$. Given the limited data size of only 500 instances, there exists a potential risk of the discriminator overpowering the generator, which could adversely impact the training process. To

Method	Number of Synth Images	Accuracy (%)
Baseline		
Supervised	None	54.54
Class-conditional GAN		
Supervised	500	54.63
	1000	53.85
GAN Distillation	∞	54.84
Unconditional GAN		
Ours (GenMatch-GIQA)	∞	**63.22**

Table 1: Results in CIFAR-10. Our approach shows an 8.68% improvement in performance compared to the traditional supervised method, while other techniques utilizing class-conditional GANs show minimal performance improvements.

mitigate this effect, we update the generator twice for each single update of the discriminator, successfully preventing such undesired occurrences. ADA is employed with an initial augmentation probability of zero, and Ada target probability of 0.6 throughout training. The augmentation policy is updated every 500 thousand images, with a frequency of every 4 training steps.

Following the training of StyleGAN2-ADA, the classifier is subsequently trained with GenMatch loss, using sampled image pairs from the generator where $\epsilon = 0.25$. Stochastic gradient descent (SGD) with a momentum of 0.9 is employed as the optimizer of the classifier. The learning rate starts from 0.03 and decays with a cosine learning rate schedule. The batch size on the labeled data is set to 64, while the batch size on the generated unlabeled data is set to 448. The weight between the supervised loss and the consistency loss, denoted as λ in Eq. 6, is 1. The parameter in the threshold indicator function, denoted as τ, is set to 0.95. For the GenMatch-GIQA, the feature dimensionality is diminished to 32 via PCA. For GMM, we use 10 components, which is the same number as the class number of CIFAR-10. During training, the exponential moving average of the GenMatch model is computed with a momentum of 0.999 and is used for the evaluation.

4.2. Main Experiments

We compare our technique with three baselines. The first baseline is the classifier trained by a standard supervised technique with 500 training data. The second baseline is also the classifier trained by standard supervised technique but with generated labeled data created by the class-conditional StyleGAN2-ADA to the origin training data. Two experiments were performed: the first utilizes 500 pieces of generated labeled data, and the second uses 1000 pieces, which include the initial 500 and an additional 500 generated labeled data. The third baseline is the experimental result of GAN Distillation [27], the most recent paper that improves the performance of the classifier using a class-conditional GAN. The version of our method used for comparison with other techniques is GenMatch-GIQA. The StyleGAN2-ADA used for the second and third baselines is class-conditional, while the StyleGAN2-ADA used for our method is class-unconditional. The hyperparameters used to train both GANs are set to be the same. GAN Distillation and our technique do not train the classifier with a fixed number of generated images, but an infinite number of those.

Method	Accuracy (%)
GenMatch	61.57
GenMatch-Conf	62.59
GenMatch-GIQA	**63.22**

Table 2: Experiments with selection of different uncertainty

As shown in Table 1, adding images generated by a class-conditional GAN hardly makes any difference in performance, and in fact, it slightly lowers it. This aligns with the results provided in prior works [32, 29, 30], implying that merely employing a class-conditional GAN without any additional techniques is insufficient to enhance accuracy. Furthermore, GAN Distillation only slightly improves the accuracy compared to a classifier trained solely on real data. In contrast, our method yields an improvement of 8.68%, which is notable.

4.3. Effect of Uncertainty Selection

In Sec. 3.3, we proposed uncertainty as a method to effectively integrate GenRep and Flexmatch. In this paragraph, we provide experiments on how the accuracy varies depending on different uncertainty selections. We compare the following three approaches: 1) applying Eq. 3 directly without considering uncertainty, 2) using confidence as the uncertainty in Eq. 4, and 3) utilizing GIQA as the uncertainty in Eq. 4. All hyperparameters and training schemes are set as same as the main experiment. As shown in Table 2, considering uncertainty improves performance compared to not considering it, and using GIQA as the uncertainty is better than using confidence. Nevertheless, GenMatch exhibits a 7.03% higher accuracy compared to standard supervised learning, indicating that the application of a consistency rule to generated data alone is sufficient to transfer the representations learned by the GAN to the classifier. The superior performance of GenMatch-Conf and GenMatch-GIQA over GenMatch suggests, as we antici-

Method	Accuracy (%)
Pseudo-Label	52.17
FixMatch	60.61
GenMatch-GIQA	**63.22**

Table 3: Experiments with the selection of different Threshold Indicator Function. "Pseudo-Label" denotes the result of substituting Eq. 9 into GenMatch-GIQA, and "FixMatch" denotes the result of substituting Eq. 10 into GenMatch-GIQA.

pated, that it is beneficial to configure the classifier to follow the prediction of the generated data that has a higher probability of being correct. In the following section, we provide empirical evidence demonstrating the premise that the classifier renders more precise predictions when dealing with samples with higher Confidence and GIQA values.

4.4. Effect of Threshold Indicator Function

The method we propose, GenMatch, is grounded on the threshold indicator function of FlexMatch (Eq. 2). In Sec. 3.2, we argued the choice of the FlexMatch indicator function was due to the stability it offers in learning. In this section, we present experimental results using different indicator functions. Semi-supervised techniques encompass a wide variety of threshold indicator functions, and we conduct additional experiments on two of them: one being a Pseudo-Label [21] technique that always sets the threshold indicator function to 1, and the other being a FixMatch [33] technique that sets the threshold indicator function to 1 only when max-confidence exceeds 0.95, and to 0 otherwise. We provide experimental results for these two cases. The two cases can be expressed in the form of equations as follows:

$$\mathcal{I}_{Pseudo}(\boldsymbol{u}) = 1, \tag{9}$$

$$\mathcal{I}_{FixMatch}(\boldsymbol{u}) = \mathbb{1}(\max(q(\boldsymbol{u})) > \tau). \tag{10}$$

where τ is set to 0.95. We provide the results when the indicator function of GenMatch-GIQA is replaced with the indicator function of Pseudo-Label, i.e., integrating Eq. 6, 8, and 9, and when the indicator function of GenMatch-GIQA is replaced with the indicator function of FixMatch, i.e., integrating Eq. 6, 8, and 10. The results are shown in Table 3. As shown in Table 3, we confirm that the performance is poorest when the indicator function of Pseudo-Label is used. In fact, the performance is even worse than that of the conventional supervised method presented in Table 1, which does not use any generated data at all. This suggests that the threshold indicator function not only stabilizes the learning process but also serves to filter out the generated data that are not beneficial for learning. The accuracy is lower when using the indicator function of FixMatch compared to our method. This is likely due to the tendency of FixMatch to focus on learning the easier classes.

Model	Train FID	Test FID
Unconditional StyleGAN2-ADA	84.59	35.33
Conditional StyleGAN2-ADA	86.94	37.66
Unconditional DDPM	81.52	55.84

Table 4: Measurement of FID across various generative models. The train FID refers to the FID measured on a sample of 500 images taken from the CIFAR-10 train set, which was used in our experiments. The test FID refers to the FID measured on the entirety of the CIFAR-10 test set. The reason for the relatively lower test FID compared to the train FID is because the sample size of the train data is 500, whereas the test data comprises 10,000 images.

4.5. FID Comparison

In this section, we validate the justification for our main experiment and the selection of class-unconditional StyleGAN2-ADA over class-unconditional DDPM through FID measurements. As shown in Table 4, the FID of class-unconditional GAN and class-conditional GAN are measured similarly in both the train and test sets. This indicates that there is not a significant performance difference between the class-unconditional GAN and class-conditional GAN used in our main experiment, suggesting that the results of the main experiment are not dictated by the performance difference between the GANs. Comparing the results of class-unconditional GAN and class-unconditional DDPM, the train FID is relatively lower for class-unconditional DDPM while the test FID is much higher. This indicates that class-unconditional DDPM induces significant overfitting in the limited data scenario, which is why we chose StyleGAN2-ADA over DDPM as the base model for our technique.

5. Conclusion

We have proposed a method to enhance the accuracy of classifiers in limited data scenarios by leveraging class-unconditional GANs. The method integrates the latent transformation and the consistency loss, serving as a way for classifiers to learn rich representations with class-unconditional GANs. To better apply the FlexMatch loss to generated data, we further introduce uncertainties by GIQA and max-confidence. We demonstrate that effective utilization of class-unconditional GANs can improve classifier accuracy, yielding better results than those using class-conditional GANs. The limitation is the time consumption

required for training, caused by continuous image generation by the GAN and the inherently slow speed of semi-supervised techniques. In future research, it would be an interesting direction to address these issues to develop a more practical and scalable technique.

References

[1] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[3] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *International Conference on Learning Representations (ICLR)*, 2017.

[10] Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Giqa: Generated image quality assessment. In *European Conference on Computer Vision (ECCV)*, 2020.

[11] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2014.

[14] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[15] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[16] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Quan Kong, Bin Tong, Martin Klinkigt, Yuki Watanabe, Naoto Akira, and Tomokazu Murakami. Active generative adversarial network for image classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[21] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning Workshops(ICMLW)*, 2013.

[22] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[23] Mingkun Li and Ishwar K. Sethi. Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28:1251–1261, 2006.

[24] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] Seongkyu Mun, Sangwook Park, David K. Han, and Hanseok Ko. Genrative adversarial network based acoustic scene training set augmentation and selection using svm hyperplane. In *In Detection and Classification of Acoustic Scenes and Events Workshop*, 2017.

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[27] Matteo Pennisi, Simone Palazzo, and Concetto Spampinato. Self-improving classification performance through gan distillation. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[29] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[30] Suman Ravuri and Oriol Vinyals. Seeing is not necessarily believing: Limitations of bigGANs for data augmentation. In *International Conference on Learning Representations Workshops (ICLRW)*, 2019.

[31] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *European Conference on Computer Vision (ECCV)*, 2018.

[33] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(56):1929–1958, 2014.

[36] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[37] Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[38] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[40] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[42] Xinyue Zhu, Yifan Liu, Jiahong Li, Tao Wan, and Zengchang Qin. Emotion classification with data augmentation using generative adversarial networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018.