Imbalanced Data Classification via Cooperative Interaction Between Classifier and Generator

Hyun-Soo Choi, Dahuin Jung, Siwon Kim, and Sungroh Yoon, *Senior Member, IEEE*

***Abstract*— Learning classifiers with imbalanced data can be strongly biased toward the majority class. To address this issue, several methods have been proposed using generative adversarial networks (GANs). Existing GAN-based methods, however, do not effectively utilize the relationship between a classifier and a generator. This article proposes a novel three-player structure consisting of a discriminator, a generator, and a classifier, along with decision boundary regularization. Our method is distinctive in which the generator is trained in cooperation with the classifier to provide minority samples that gradually expand the minority decision region, improving performance for imbalanced data classification. The proposed method outperforms the existing methods on real data sets as well as synthetic imbalanced data sets.**

***Index Terms*— Classification, decision boundary, deep learning, generative adversarial networks (GANs), imbalanced data, supervised learning.**

I. Introduction

THE imbalanced data problem is a phenomenon in which the number of samples in minority and majority classes has a large gap in training data. The medical domain is representative fields of the imbalanced data problem [1]. Domains, including biology, network intrusion, and fraud detection, also suffer from the same phenomenon [2]–[5]. The imbalance ratio (IR) between minority and majority classes varies depending on the application, and in severe cases, the IR may be as high as 100 000 [6], [7]. In many applications, it is more costly and important to classify the minority than the majority class [8], [9]. Since imbalanced data causes severe performance degradation in machine learning, it is an important research topic in both academia and industry [5]. The decision boundary learned by standard machine learning with imbalanced data can be strongly biased by the majority class, causing low precision of the minority class. Ultimately, the goal of addressing the imbalanced data problem is to increase the classification performance on the minority class.

Various methods have been proposed to overcome the imbalanced data problem [10]. Among existing methods, the data-level balancing approach has been widely used to balance training samples [6], [7], [11]–[19] The loss-based (cost-sensitive) balancing approach, which gives larger weights on minority samples than the majority samples, has also been widely used [20]–[22]. The classifier-design approach for balancing is to design algorithmic techniques embedded in a classifier to overcome the class-imbalance problem inherently [23]–[26]. These conventional methods have been effectively applied to the shallow learning classifier using handcrafted features, such as SHIFT [27] and SURF [28]. In recent years, deep learning classifiers outperform the shallow learning classifiers by a large margin [29]. Hence, even when the shallow learning classifiers adopt the imbalanced learning scheme, they are difficult to exceed the baseline performance of the deep learning classifier without an imbalanced learning scheme. In this article, we aim to develop a method that can be applied to a deep learning classifier that uses images as inputs directly without handcrafted features.

Recently, to tackle the imbalanced problem in deep learning classifiers, generative adversarial networks (GANs) [30] have been used to generate high-dimensional synthetic samples in the minority class [20]–[22], [31], [32]. Most of the existing GAN-based methods do not consider the effect on a classifier when training a generator and a discriminator of GAN, thus limiting improvement opportunities for the generated samples. To handle this issue, the concept of TripleGAN [33] has been adopted to address imbalanced data classification [34]. However, since TripleGAN was proposed for semisupervised learning, they have an adversarial relationship between a classifier and a discriminator, which limits the performance improvement.

In this article, instead of the adversarial relationship between classifier and GAN (generator/discriminator), we propose a novel cooperative relationship between classifier and GAN (generator/discriminator). In deep classifiers, implicit feature embedding techniques via multiple layers are used to obtain discriminative features that are easily separable between classes. Nevertheless, when the training samples are not sufficient or imbalanced, the deep features may not be embedded well enough to form the trained borderline similar to the true borderline between classes. In this perspective, to mitigate the problem of training the deep network with imbalanced

Manuscript received 6 March 2020; revised 2 October 2020; accepted 12 December 2020. Date of publication 2 February 2021; date of current version 4 August 2022. This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (Ministry of Science and ICT) under Grant 2018R1A2B3001628 and in part by the Brain Korea 21 Plus Project in 2021. *(Corresponding author: Sungroh Yoon.)*

Hyun-Soo Choi is with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea, also with the Vision AI Labs, SK Telecom, Seoul 04539, South Korea, and also with the Department of Computer Science and Engineering, Kangwon National University, Gangwon-Do 24341, South Korea.

Dahuin Jung and Siwon Kim are with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea.

Sungroh Yoon is with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea, also with ASRI, INMC, ISRC, Seoul National University, Seoul 08826, South Korea, and also with the Institute of Engineering Research, Seoul National University, Seoul 08826, South Korea (e-mail: sryoon@snu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3052243.

Digital Object Identifier 10.1109/TNNLS.2021.3052243

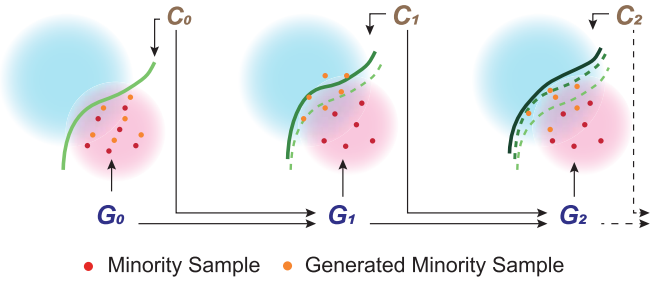

Fig. 1. Cooperative interaction between a generator and a classifier to expand the skewed decision region of minority class.

data, we aim to balance the samples in the overlapping region around the trained borderline by the proposed cooperative structure of classifier, generator, and discriminator along with a borderline regularization.

For an imbalanced data including between- and within-class imbalance, we assume that the samples in the overlapping region are also imbalanced and yield a biased training of the classifier. In particular, when training with the within-class imbalanced data, the imbalance would become more severe around the borderline adjacent to the within-class minority region. When training the deep classifier without any consideration of the imbalance, the trained borderline on the overlapping region can be biased toward the minority region. Hence, as shown in Fig. 1, our approach tries to move the trained borderline toward the majority region by training the classifier with the generated minority samples via the proposed cooperative training of classifier, generator, and discriminator, along with the borderline regularization.

Our key concepts and contributions are as follows.

1) A three-player structure (a classifier, a discriminator, and a generator) is proposed in a cooperative relationship between the generator and the classifier to address imbalanced data learning.
2) A novel regularization term is embedded to expand the decision boundary of the minority class in a cooperative interaction of the generator and the classifier.
3) We develop an alternating optimization strategy, along with a regularization decaying scheme to prevent over-generalization, in which the generator and the classifier are trained alternately to learn a desirable distribution.
4) The proposed method is validated experimentally using in-depth self-analysis as well as comparing with the existing methods.

II. Related Work

A. Imbalanced Data Classification

1) Data-Level Balancing Approach: The data-level balancing approach is divided into three categories: undersampling, oversampling, and hybrid methods. The undersampling methods, such as clustering centroids (C-Centroids) and condensed nearest neighbor (CN-Neighbor), balance the training data by removing majority samples [11]. The oversampling methods generate synthetic minority samples to balance training data. Representative methods are the synthetic minority oversampling technique (SMOTE) [6]. Several variants have been proposed to overcome the limitations of SMOTE. Borderline-SMOTE (B-SMOTE) [12], neighborhood rough set boundary-SMOTE (NRSB-SMOTE) [35], and the adaptive synthetic sampling approach (ADASYN) [13] adaptively generates samples considering the proportions of adjacent majority data. To ensure that the generated samples belong to a minority class, majority weighted minority oversampling technique (MWMOTE) [7] effectively identifies minority sample-dominated clusters that become sources of oversampling. Gaussian SMOTE (G-SMOTE) [15] achieves sample diversity by replacing the uniform distribution of SMOTE with Gaussian distribution. Real-value negative selection oversampling (RSNO) [16] synthesizes a minority sample without accessing minority samples. The hybrid methods, such as SMOTE editing the nearest neighbor (SMOTENN) [17] and SMOTE-Iterative partitioning filter (SMOTE-IPF) [18], filter out unsafe samples after SMOTE-based oversampling. However, most of these data-level methods consider only local information; therefore, they cannot reflect the entire data distribution [21]. Furthermore, these methods are based on interpolation with simple distance metrics (e.g., Euclidean), and therefore, they only consider numerically featurized data, which do not successfully address other types of data such as image [21].

2) Loss-Based (Cost-Sensitive) Balancing Approach: The cost-sensitive approach modifies the existing classification loss (cost) function (i.e., cross-entropy loss) to give additional considerations on minority class samples. Representative methods include class rectification loss (CRL) [36], max-pooling loss (MPL) [37], and focal loss (Focal) [38]. In detail, CRL rectifies the incremental class bias in the model by making use of batchwisely selected hard positive and negative samples of the minority classes. In the case of MRL, it indirectly addresses both interclass and intraclass imbalance by performing a generalized max pooling of pixel-specific losses. In the case of the focal loss, it reshapes the loss function to downweight samples of the majority classes. The advantage of the cost-sensitive approach is that it can be simply applied to a training procedure for a deep learning network. However, as illustrated in our experiments, the performance improvements of cost-sensitive methods are bounded because the amount of samples of the minority classes that can be used for learning data distributions is still limited.

3) Classifier-Design Approach for Balancing: This approach is to design algorithmic techniques embedded in a classifier to overcome the class-imbalance problem inherently. The representative methods in this approach are the regression-based linear classifier minimizing one-pass area under the receiver operating characteristic curve (AUC) convex loss [23], the kernelized online imbalanced learning (KOIL) of support vector machine (SVM) [24], the ensemble strategy of SVMs [25], and Random forests combining multiple decision trees to learn highly imbalanced medical data [26]. However, since the above methods belong to shallow learning, these methods cannot access extremely high-dimensional data, such as raw images without handcraft features. Recently, deep-network classifiers have shown more

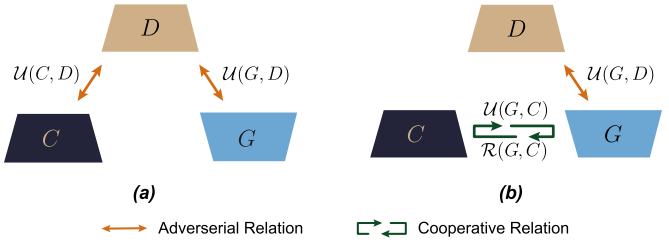

Fig. 2. Difference between (a) TripleGAN and (b) Proposed.

remarkable performance by directly learning raw images than the shallow learning methods [39]. Thus, our work intends to propose an imbalanced data learning approach applicable to a deep-network classifier.

4) GAN-Based Balancing Approach: In recent years, to reflect the actual distribution of a minority class in learning a deep-network classifier, GAN has been used as oversampling method. GAN-based works exploit the GAN model, such as deep convolutional GAN (DCGAN) [20], conditional GAN (cGAN) [21], or cycleGAN [22] to restore the actual distribution by synthesizing data. Balancing GAN (BAGAN) [31] is a slightly modified version of an auxiliary classifier GAN [40] that specializes in the generation of minority class samples. In all of these studies, the process of generating samples through GAN and the process of learning a classifier with the generated samples are independent. To handle this issue, the concept of TripleGAN [33] has been adopted to address imbalanced data classification [34]. However, TripleGAN and its variant, Enhanced TripleGAN (E-TripleGAN) [41], are originally proposed for semisupervised learning, and it has an adversarial relationship between a classifier and a discriminator for pseudolabeling. This adversarial relation is of no use for imbalanced data learning as unlabeled samples are not present in an imbalanced data problem. Hence, we remove the adversarial relationship and propose a novel cooperative relationship between the generator and the classifier.

B. Effectiveness of Samples Near Class Boundary

The samples near the decision boundary play an important role in training classifiers. For this reason, various research works have attempted to utilize the concept of the decision boundary, such as knowledge distillation via decision boundary transfer [42], classifier training robust to adversarial attacks [43], and out-of-distribution detection problems [44]. To the best of our knowledge, however, our work is the first attempt to address an imbalanced classification problem by generating samples with GAN to expand the decision boundary of the minority region. The novelty of our study is the decision boundary regularization with its decay, which promotes the convergence of the alternating optimization in training our three-player structure for mitigating the imbalance issue.

III. Proposed Method

To formulate our concept for expanding the minority decision region to have a desirable distribution, we design a three-player structure for imbalanced data learning

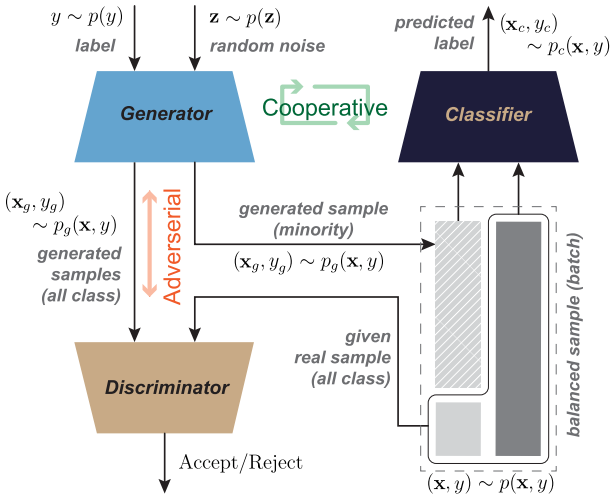

Fig. 3. Proposed GAN architecture with notations.

(see Section III-A) and develop an alternating training scheme with a cooperative training loop between the generator and the classifier (see Section III-B).

A. Three-Player Structure for Imbalanced Data Learning

1) Motivation: As mentioned in Section II-A4, TripleGAN [33] and E-TripleGAN [41] are designed to generate pseudolabels for unlabeled samples for facilitating semisupervised learning. The discriminator (D) in TripleGAN discriminates a given true label and a false label generated by the classifier (C). TripleGAN has an adversarial relationship $\mathcal{U}(C, D)$ between D and C, as shown in Fig. 2(a). In the proposed model, a cooperative relationship is developed between the generator (G) and C to ensure that both G and C are benefitted by joint training. For developing the cooperative relationship, additional utility terms $\mathcal{U}(G, C)$ and $\mathcal{R}(G, C)$ have been proposed, as shown in Fig. 2(b). The proposed three-player structure is designed to expand the minority region by generating minority samples toward the borderline between the majority and the minority in the early stage of training and finally to provide densely distributed samples within an expended minority region. In Section III-A2, we describe the details of the proposed utility function and discuss its impact on imbalanced data learning.

2) Utility Function: To describe our utility function for our three-player structure shown in Fig. 3, we define notations. $\mathbf{x}$ denotes the input data and y denotes the output label. Then, $\mathbf{x} = G(\mathbf{z}, y)$ denotes a generated sample from the randomly generated $\mathbf{z}$ and y values. It is assumed that the observed training samples are sampled from unknown $p(\mathbf{x}, y)$ and that samples from both $p(\mathbf{z})$ and $p(y)$ can be easily obtained by using simple known distributions (normal or uniform and so on.) during training. The classified label is denoted by $y = C(\mathbf{x})$ and the output of D is denoted by $D(\mathbf{x}, y)$ for given $\mathbf{x}$ and y. In addition, the joint distributions $p_g(\mathbf{x}, y)$ and $p_c(\mathbf{x}, y)$ are defined as

$$p_g(\mathbf{x}, y) := p(y)p_g(\mathbf{x}|y) = p(y)p(G(\mathbf{z}, y)|y) \quad (1)$$

$$p_c(\mathbf{x}, y) := p(\mathbf{x})p_c(y|\mathbf{x}) = p(\mathbf{x})p(C(\mathbf{x})|\mathbf{x}) \quad (2)$$

where $p_g(\mathbf{x}|y) = p(G(\mathbf{z}, y)|y)$ in (1) indicates the distribution of synthetic samples generated by G for a given label y and $p_c(y|\mathbf{x}) = p(C(\mathbf{x})|\mathbf{x})$ in (2) indicates the distribution of labels, determined by C, for the given samples (generated or observed).

Our goal is to design a utility function $\mathcal{U}(C, D, G)$ for imbalanced data learning in three-player game given by

$$\min_{C,G} \max_{D} \mathcal{U}(C, D, G). \tag{3}$$

In this article, the utility function for imbalanced data learning is proposed as

$$\mathcal{U}(C, D, G) = \mathcal{U}_g(D, G) + \mathcal{U}_{c_1}(C) + (1-\lambda)\mathcal{U}_{c_2}(G, C) + \lambda\mathcal{R}(G, C) \tag{4}$$

where the last two terms are distinctive aspects against Triple-GAN and they take key roles for cooperative training of G and C in our method. The third term is for jointly training of G and C, whereas the fourth term $\mathcal{R}(G, C)$ is for minority region expansion. These two terms are linked by a hyperparameter λ for tradeoff scheduling between the two terms (for details, see Sections III-A3 and III-B3). Each term is defined formally in the following.

The term $\mathcal{U}_g(D, G)$ is well known utility function of cGAN [21], which is defined as

$$\mathcal{U}_g(D, G) = \mathbb{E}_{p(\mathbf{x},y)}[\log D(\mathbf{x}, y)] + \mathbb{E}_{p_g(G(\mathbf{z},y),y)}[\log(1 - D(G(\mathbf{z}, y), y))]. \tag{5}$$

The term $\mathcal{U}_{c_1}(C)$ is for training C with only the observed (real) data, whereas $\mathcal{U}_{c_2}(G, C)$ is for joint training of G and C, which are defined as

$$\mathcal{U}_{c_1}(C) = \mathbb{E}_{p(\mathbf{x},y)}[-\log p_c(y|\mathbf{x})] \tag{6}$$

$$\mathcal{U}_{c_2}(G, C) = \mathbb{E}_{p_g(G(\mathbf{z},y),y)}[-\log p_c(y|G(\mathbf{z}, y))]. \tag{7}$$

In particular, $\mathcal{U}_{c_2}(G, C)$ makes C be trained to well classify the samples generated by G, whereas G be trained to generate extra samples helpful for C.

Lastly, $\mathcal{R}(G, C)$ is introduced for expansion of the minority region. To define $\mathcal{R}(G, C)$, the classification scores for the minority and majority classes are denoted by $C_{\text{mi}}(G(\mathbf{x}))$ and $C_{\text{ma}}(G(\mathbf{x}))$, respectively, and a generated sample is denoted by $\mathbf{x}_g$. Using these terms, $\mathcal{R}(G, C)$ is defined as

$$\mathcal{R}(G, C) = \mathbb{E}_{p_g(\mathbf{x},y)}[s_g] \tag{8}$$

where

$$s_g = \begin{cases} [C_{\text{mi}}(\mathbf{x}_g) - C_{\text{ma}}(\mathbf{x}_g)]^2, & \text{if } C_{\text{mi}}(\mathbf{x}_g) > C_{\text{ma}}(\mathbf{x}_g) \\ 0, & \text{otherwise} \end{cases}$$

where C_{mi} and C_{ma} have values between 0 and 1 since the classifier uses the softmax activation in the output layer. The role of $\mathcal{R}(G, C)$ is presented in Section III-A3.

3) Effect to Imbalanced Data Learning: In $\mathcal{R}(G, C)$ of (8), if the generated sample $(\mathbf{x}_g)$ is placed in the minority region of the current classifier (C), the minority score $(C_{\text{mi}}(\mathbf{x}_g))$ is greater than the majority score $(C_{\text{ma}}(\mathbf{x}_g))$. This is the case of upper condition in (8); in this case, the minimization of $\mathcal{R}(G, C)$ forces to generate samples near the boundary

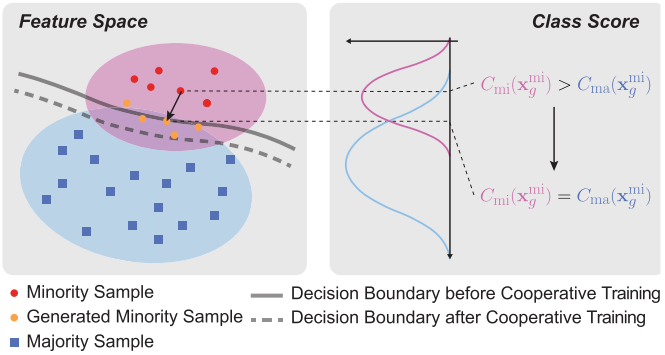

Fig. 4. Through cooperative training, G is trained to generate minority samples (yellow) crossing the decision boundary between the minority class and the majority class. As indicated by the dashed line, the samples generated by the tuned G contribute to the expansion of the minority class region for the next training of C.

satisfying $C_{\text{mi}}(\mathbf{x}_g) = C_{\text{ma}}(\mathbf{x}_g)$, as shown in Fig. 4. These generated samples take a role in the next training of the classifier to expand the decision boundary toward the majority region, as shown in Fig. 1. If the generated sample $(\mathbf{x}_g)$ is already placed in the majority region [the lower condition in (8)], this sample does not move to prevent the harmful effect to the majority class. Hence, as shown in Fig. 4, minimizing $\mathcal{R}(G, C)$ plays a role in generating samples to expand the minority region in the direction of the majority region in training C.

However, the endless expansion of the minority region might cause overgeneralization of the minority class, i.e., potential overlapping issues, which degrades the classification performance. To mitigate the potential overlapping problem, we introduce a hyperparameter λ for a tradeoff scheduling between $\mathcal{U}_{c_2}(G, C)$ and $\mathcal{R}(G, C)$. By reducing λ gradually to zero during the alternate training of C and G, the role of $\mathcal{R}(G, C)$ vanishes and thus the overgeneralization of the minority class stops. This implies that G is trained for the expansion of the minority class decision region in the early training stage only.

As λ decays, the cooperative term of $\mathcal{U}_{c_2}(G, C)$ contributes to the generation of minority samples achieving sufficiently balanced distribution within an expanded minority region by utilizing the shareable low-level features from the majority samples. When training the generator via the term $\mathcal{U}_{c_2}(G, C)$, the majority samples can provide shareable local feature information (edge, blob, texture, and so on) helpful to the minority sample generation, which can mitigate the overfitting problem encountered when an only small number of minority samples are used for generator training. This claim is supported by the study [45], which reports that the lower (input-side) layers of a convolution network learn the local features shareable among various classes. When generating images, the generator such as DCGAN [20] stores the local feature information in the higher (output-side) layers contrary to the convolution network used for a classifier. More details about the decaying scheme are described in Section III-B.

Theorem 1 shows that the proposed utility function has an equilibrium when λ decays to zero.

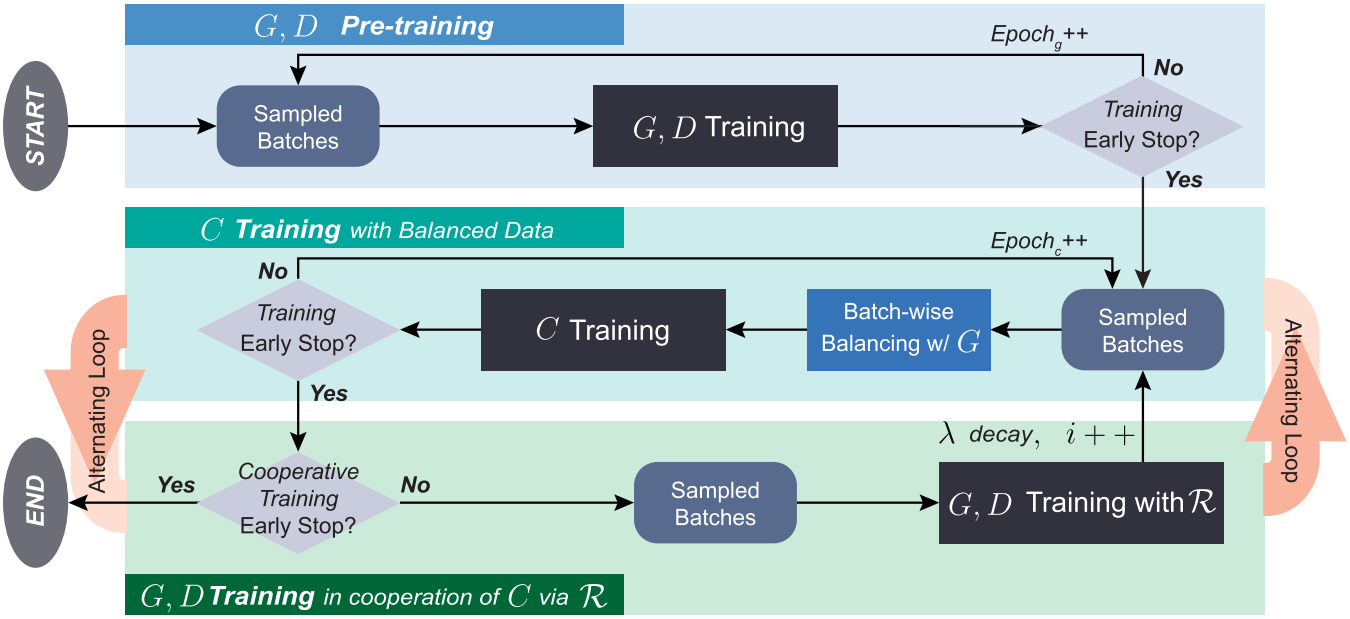

Fig. 5. Overall scheme of alternating training of C an G/D. For details on each block, see the pseudocode in Algorithm 1.

TABLE I
EMPIRICAL UTILITY SUBFUNCTIONS

$$\widetilde{\mathcal{U}}_g(\theta_g,\theta_d)=\frac{1}{m}\sum_{\mathbf{x}\in B_r}\log\widetilde{D}\left(\mathbf{x},y;\theta_d\right)+\frac{1}{m_g}\sum_{\widetilde{G}(\mathbf{z},y;\theta_g)\in B_g}\log\left(1-\widetilde{D}\left(\widetilde{G}(\mathbf{z},y;\theta_g),y;\theta_d\right)\right)$$

$$\widetilde{\mathcal{U}}_{c_1}(\theta_c)=\frac{1}{m}\sum_{\mathbf{x}\in B_r}y\log C(\mathbf{x};\theta_c)+(1-y)\log(1-C(\mathbf{x};\theta_c))$$

$$\widetilde{\mathcal{U}}_{c_2}(\theta_g,\theta_c)=\frac{1}{m_g^{\mathrm{mi}}}\sum_{\widetilde{G}(\mathbf{z},y^{\mathrm{mi}};\theta_g)\in B_g}y\log C(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c)+(1-y)\log(1-C(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c))$$

$$\mathcal{R}(\theta_c,\theta_g)=\frac{1}{m_g}\sum_{\widetilde{G}(\mathbf{z},y;\theta_g)\in B_g}s_g,\ \text{where}$$

$$s_g=\begin{cases}\left[\widetilde{C}_{\mathrm{ma}}(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c)-\widetilde{C}_{\mathrm{mi}}(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c)\right]^2, & \text{if } \widetilde{C}_{\mathrm{mi}}(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c)>\widetilde{C}_{\mathrm{ma}}(\widetilde{G}(\mathbf{z},y;\theta_g);\theta_c)\\ 0, & \text{otherwise,}\end{cases}$$

Note) m: # of observed samples (B_r), m_g: # of generated samples (B_g), m_g^{mi}: # of generated minority samples in B_g.
Note) The tilde $\widetilde{\ }$ notation indicates an empirical function calculated from the training samples.

Theorem 1: The equilibrium of $\mathcal{U}(C, D, G)$ with $\lambda = 0$ is achieved if and only if

$$p(\mathbf{x}, y) = p_g(\mathbf{x}, y) = p_c(\mathbf{x}, y) = p_c(G(\mathbf{z}|y), y). \tag{9}$$

Note that $p_c(\mathbf{x}, y) = p_c(G(\mathbf{z}|y), y)$ means that the training of C relies on the distribution of samples generated by G at the equilibrium. Hence, how well G learns the true distribution dominates the performance of C. The proof is shown in Appendix A.

B. Training Scheme

1) Overall Scheme: To promote cooperation between G and C, along with $\mathcal{U}_{c_2}(G, C)$ and $\mathcal{R}(G, C)$, in the optimization process, we adopt an alternating optimization between the training of G/D and the training of C. The overall scheme of the proposed method is outlined in Fig. 5. To stop the training of G/D, C, or alternating loop, we adopt the validation-based early stopping rule [46]. Before starting alternating optimization, we pretrain G/D with the observed imbalanced data for the initial generator. As the first step of the alternating loop, C is trained with a balanced batch generated by fixed G/D. Thereafter, G/D is trained in cooperation with C, along with the decision boundary regularization $\mathcal{R}(G, C)$. These two optimizations are repeated iteratively in an alternating loop. Each optimization is described in the following. The alternating loop induces G to generate minority samples that help C expand the minority region during the initial training phase. As λ decays with increasing of alternating iterations, the joint term $\mathcal{U}_c(G, C)$ plays a major role in achieving a desirable distribution within each decision region determined by the trained C.

The training parameters of G, D, and C are denoted by θ_g, θ_d, and θ_c, respectively. Then, letting $\widetilde{\mathcal{U}}(\cdot)$ be an empirical utility function that is parameterized from $\mathcal{U}(\cdot)$ in (4), the total empirical utility function $\widetilde{\mathcal{U}}(\theta_d, \theta_g, \theta_c)$ of $\mathcal{U}(D, G, C)$ is denoted by

$$\widetilde{\mathcal{U}}(\theta_d,\theta_g,\theta_c)=\widetilde{\mathcal{U}}_g(\theta_d,\theta_g)+\widetilde{\mathcal{U}}_{c_1}(\theta_c)+(1-\lambda)\widetilde{\mathcal{U}}_{c_2}(\theta_c,\theta_g)+\lambda\mathcal{R}(\theta_c,\theta_g) \tag{10}$$

where a decaying rule of λ is designed as λ_i for the ith iteration in Section III-B3. The details of each term are given in Table I. The details for the training of θ_c with a balanced batch generated by G and the training of θ_g/θ_d in cooperation with C are described in Sections III-B2 and III-B3.

2) Training of C With Balanced Batch by G: In this stage, only C is trained using the empirical utility function, $\widetilde{\mathcal{U}}(\theta_d, \theta_g, \theta_c)$, in (10), that is, only the parameter vector θ_c of C is updated after fixing θ_g and θ_d. As θ_g is fixed, θ_c is updated by descending the empirical utility function in (10) along its stochastic gradient with respect to θ_c. The samples of minority class for balancing are generated by the trained G in a batchwise manner, whereas the existing GAN-based balancing methods adopt a one-shot balancing policy. In a one-shot balancing policy, the fixed number of minority samples is generated before training C as a preprocessing step. In batchwise balancing, however, new samples are generated for each batch. Batchwise balancing is advantageous because it can fully utilize G by generating an unlimited number of samples, as new samples are generated repeatedly in a batch-wise manner until C converges. Another advantage is memory efficiency. Unlike one-shot balancing, batchwise balancing requires only a small amount of memory for as much as one batch size.

3) Training of G/D in Cooperation With C Along With $\mathcal{R}$: This training stage is designed to train G/D to pursue a balanced distribution by expanding the minority decision region and generating sufficient samples within the decision region. To prevent overgeneralization of the minority region, we designed a decaying rule of λ in the utility function (10). Specifically, λ_i for the ith iteration is exponentially reduced by multiplying hyperparameter $\gamma \in (0, 1]$ every iteration loop (i.e., $\lambda_i = \gamma \lambda_{i-1}$ for the ith iteration). The value of γ is empirically selected in experiments. In each alternating loop, by fixing θ_c, θ_g/θ_d are updated by descending/ascending $\widetilde{\mathcal{U}}(\theta_d, \theta_c, \theta_g)$ in (10) along their stochastic gradient with respect to θ_g/θ_d. Note that θ_g/θ_d can also be trained for several epochs in each loop, but one epoch was empirically sufficient. The pseudocode of the proposed alternating training scheme is given in Algorithm 1.

Algorithm 1 Alternating Training Scheme of C and G/D

Notation:
λ : the trade-off control parameter between $\mathcal{U}_{c_2}(\theta_g, \theta_c)$ and $\mathcal{R}(\theta_g, \theta_c)$
$\gamma \in (0, 1]$: hyper-parameter for λ decaying by $\lambda_i = \gamma \lambda_{i-1}$
Procedure:
Initialize $\lambda_0 = 1$ and $i = 1$
[G/D Pre-Training]
while *not converge by early stop during training* θ_g/θ_d **do**
  Sample a batch from the given data
  Train θ_g and θ_d by solving min-max problem with $\widetilde{\mathcal{U}}_g(\theta_d, \theta_g))$
end while
[Alternating Loop]
while *not converge by early stop during alternating loops* **do**
  Set $\lambda_i = \gamma \lambda_{i-1}$
  [C Training with Balanced Data]
  while *not converge by early stop during training* θ_c **do**
    Sample a batch from the given data
    Balance the batch with minority samples generated by G
    Train θ_c by minimizing the utility in (10)
  end while
  [G/D Training in cooperation of C via $\mathcal{R}$]
  while *not converge by early stop during training* θ_g/θ_d **do**
    Sample a batch from the given data
    Train θ_g and θ_d by solving min-max problem in (10)
  end while
  $i++$
end while

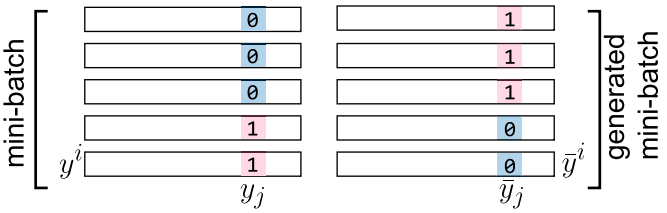

Fig. 6. Way how to choose multilabel for generate the samples.

C. Extension to Multiclass and Multilabel

To apply our method to multiclass problems, we expand a decision boundary between a minority and its neighboring majority class, which is the most influential class to the minority class. Hence, the majority class is determined by the class i^*, where $i^* = \text{argmax}_{i, i \neq \text{mi}} C_i(\mathbf{x}_g^{\text{mi}})$.

For multilabel classification, let the multilabel vector for the ith sample be denoted by $\mathbf{y}^i = [y_1^i, \ldots, y_j^i, \ldots]$, where j is an attribute index. For CelebA, either 0 or 1 is assigned to y_j^i. For balancing with the mini-batch generation, we make the multilabel vectors as inputs for G. Let $\mathbf{y}_j = \{y_j^i | i = 1, \ldots, N\}$ be a set of the jth elements of label vectors in a mini-batch, as shown in Fig. 6. Note that $\mathbf{y}$ and $\bar{\mathbf{y}}$ indicate a given training batch and a generated batch, respectively. The minority labels in $\bar{\mathbf{y}}_j$ are randomly chosen by a probability $1 - p_j$, where p_j is the ratio of minority samples in $\mathbf{y}_j$ of the training data set. The remaining elements are assigned to the majority labels. During G/D training with C, the utility function is obtained by summation of all utilities, $\sum(\mathcal{U}_j + \mathcal{R}_j)$, from each attribute.

IV. Experimental Results

A. Data Sets

In our evaluation, we utilized CIFAR-10 [47], ImageNet [29], Dementia diagnosis [48], and CelebA (multi-label) [49] data sets. CIFAR10 is a low-resolution image data set, and ImageNet is a large data set including high-resolution images. Dementia is a diagnosis data set for binary classification (control versus patient) of neuropsychological assessment profiles, where the number of control subjects is six times more than that of dementia (IR (IR) = 6). CelebA is a data set including portraits with multilabels, and some attributes (labels) are extremely imbalanced, such as baldness or hat wearing. The aspects of the data sets for our experiments are given in Table II.

Dementia and CelebA are inherently imbalanced data sets, but CIFAR10 and ImageNet are not imbalanced. Hence, we artificially constructed imbalanced data sets by subsampling the original data set of CIFAR10 for a minority class. For ImageNet, we constructed an imbalanced data set where the majority class is chosen by a class including many subclasses and the minority class is chosen by a class including a few subclasses.

To construct the minority class, two factors should be considered so that they cannot be easily classified from the

TABLE II
SUMMARY OF EVALUATION DATA SET

Dataset	Majority	Minority	IR	Training samples		Test samples		Dimension
				Majority	Minority	Majority	Minority	
CIFAR10-BC	Car	Truck	20	6,000	3,00	1,000	1,000	32 x 32
ImageNet-BC	Dog (117 sub-classes)	Cat (4 sub-classes)	29.25	152k	5,200	5,850	200	128 x 128
Dementia-BC	Control	Patient	6	2,132	349	533	87	92
CIFAR10-MC	1, 3, 5, 6, 7 classes	0, 2, 4, 6, 8 classes	20	30,000	1,500	5,000	5,000	32 x 32
CelebA-ML	Do not have an attribute (e.g, non-Bald)	Have an attribute (e.g, Bald)	upto 41	183k (label-wise IR)		20k (label-wise IR)		64 x 64

BC: Binary-class; MC: Multi-class; ML: Multi-label.

majority classes. The first factor is the degree of similarity between classes. Learning will be easy if both classes are distinct from each other even if a considerably small number of samples is provided for the minority class. Therefore, constructing classes with high similarity is desirable for evaluating the performance of imbalanced data learning. The second factor is the IR between classes. Previous studies constructed an imbalanced data set with a low IR, not higher than 2.5 (100:40) [31]; however, these low IR data are insufficient to verify the methods for an extremely imbalanced case. It is therefore desirable to set a sufficiently large value for the IR.

Considering these two factors, we constructed imbalanced data sets from the CIFAR10 [47]. Based on the first factor, we selected two highly similar classes from the original data set. For example, car (majority class) and truck (minority class) are highly similar to each other. Based on the second factor, we set IR to 20 (100:5) for CIFAR10. We used all samples in the majority class data set and randomly selected 5% of the samples from the minority class data set. A validation data set was constructed with 20% of samples in the selected training data set. For the test data set, we used the original test data set in CIFAR10. For evaluation on real-imbalanced images with high resolution, we conducted experiments on ImageNet [29]. We set up an imbalanced binary class data set where the majority class is "dog" containing 117 species and the minority class is "cat" containing four species, and thus, IR becomes 29.25. To verify the effectiveness of the proposed method on the multiclass classification problem, according to [50], we constructed an imbalanced multiclass data set by extracting 5% of the samples for five (half) classes (0, 2, 4, 6, and 8) in CIFAR10.

B. Evaluation Metrics

Several previous studies [20], [22], [31] have used accuracy as an evaluation metric. However, for extremely imbalanced data, the accuracy metric cannot precisely evaluate the minority class classification because high accuracy can be achieved with a simple zero-rule classifier, which determines all samples as the majority class. To avoid this problem, we adopted the metrics in the following table:

Metric	Definition
AUROC [51]	The area under receiver operating characteristic curve.
AUPR [52]	The area under precision-recall curve.
G-score [53]	The root of the product of class-wise sensitivity.
B-ACC [54]	The average of class-wise accuracy.

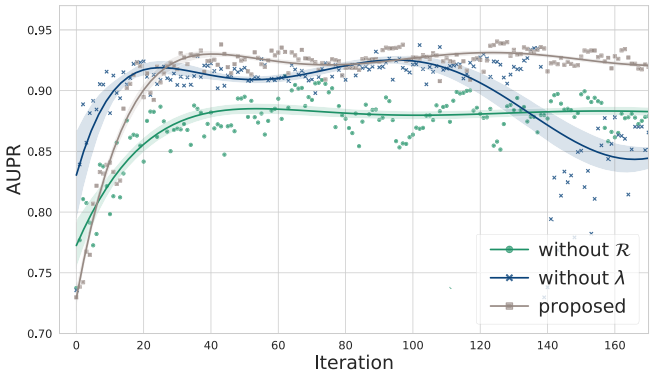

Fig. 7. Effect of λ along with $\mathcal{R}(G, C)$. The proposed method (with λ decay) shows more stable and higher performance than the other two cases.

G-score (geometric mean score) is a metric that measures the balance between classification performances on both the majority and minority classes. B-ACC (balanced accuracy) is a metric for evaluating learning processes in two-class imbalanced domains. While G-score and B-ACC are specific to particular decision thresholds, AUROC and AUPR are not specific to decision thresholds. Hence, AUROC and AUPR are more valid than the accuracy metric for performance evaluations of the trained model. AUROC is more common than AUPR, but AUPR is more sensitive than AUROC for highly imbalanced data sets [52]. For imbalanced data, the AUPR value is very low. Furthermore, as the analytical power of a classifier for the minority class increases, the AUPR value remarkably increases. Thus, AUPR is the most appropriate metric for imbalanced data classification. Furthermore, using Delong *et al.*'s method [55], we test the statistically significant difference in AUROC values against other methods.

C. Self-Analysis

Using the imbalanced data set constructed by two similar classes of the car (majority class) and truck (minority class) CIFAR-10, we deeply self-analyze our method in various aspects detailed in this section.

1) Effect of λ Along With $\mathcal{R}(G, C)$: To analyze the influence of λ and its decay scheme along with $\mathcal{R}(G, C)$ in (4), we evaluated the convergence of the optimization process for each of the three settings: without $\mathcal{R}(G, C)$, without λ decay scheme, and with λ decay scheme. Fig. 7 shows the results of the three cases using CIFAR10. In the case without

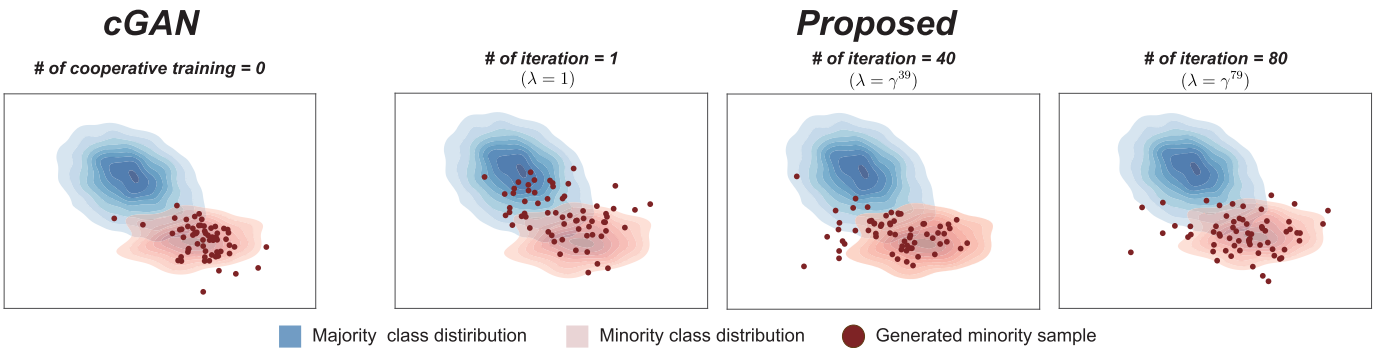

Fig. 8. Distribution of each class and generated minority samples in feature space. Without cooperative training, generated samples are located within the training data distribution. However, with cooperative training along with $\mathcal{R}(G, C)$, generated samples tend to be located on the borderline. As λ decays, generated samples return to the distribution with broader coverage.

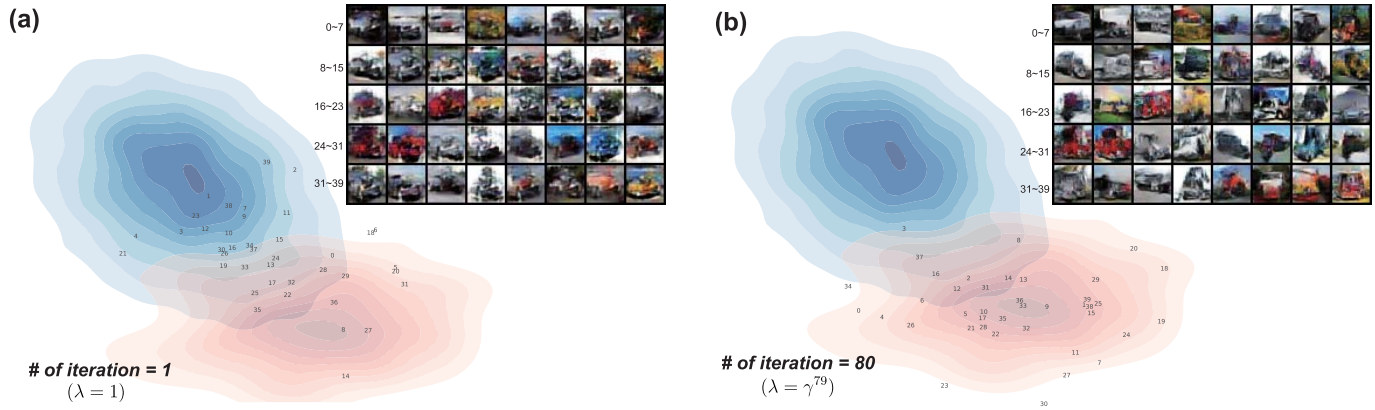

Fig. 9. Feature space mappings and images of generated minority samples (truck) against majority samples (car) in (a) early-stage iteration and (b) late-stage iteration.

$\mathcal{R}(G, C)$ (green line), using the utility function in (3), the performance was not much improved due to the premature convergence explained in Section III-A3. In the case without a λ decay scheme (blue line), performance degraded after approximately 100 iterations, due to overexpansion of the minority region. In contrast, the case with λ decay (proposed, brown line), using the utility function in (4), the high and stable performance was achieved as expected. The degree of decay for $\lambda = \gamma^{i}$ is determined by the value of γ, which is observed to be dependent on the data set. We determined γ empirically as 0.9, 0.1, and 0.5 for CIFAR10, Dementia, and CelebA, respectively.

2) Validity of Samples Generated Throughout Cooperative Training: Fig. 8 shows a map of the samples generated by the proposed GAN in the feature space. The blue and red contours represent the majority and minority class distributions, respectively, for the given training data. The dark red dots represent the 64 samples generated by G. Features in the intermediate layer of C were extracted for all samples and were visualized in 2-D space using the parametric t-distributed stochastic neighbor embedding scheme [56]. For fair visualization, we used a fixed $\mathbf{z}$ to generate samples at each iteration.

In Fig. 8, the leftmost panel shows the samples generated by cGAN learning, which was only trained in the initial phase, without cooperative training. Most of the samples are mapped in a small region within the training data distribution. The remaining panels show a map of the samples generated through repeated cooperative training. Although the samples were generated using the same $\mathbf{z}$ values, they are mapped in different positions of the feature space in every iteration. Especially, in the first cooperative training, as the value of λ is 1, most of the generated minority samples cross the decision boundary between two classes. We can see that as the λ value decays, the tendency of generating samples cross the decision boundary decreases.

Fig. 9 shows the locations of the generated minority samples in feature space. The top-right images in Fig. 9(a) and (b) are the generated sample images. The numbers left to the generated images are the indexes that correspond to the numbers written in feature space. Fig. 9(a) shows the generated sample locations after the first cooperative interaction training. As discussed in Section IV-E, due to $\lambda = 1$, the generated minority samples are located around the borderline of two classes. Fig. 9(b) shows the generated sample locations after the 80th cooperative interaction training. As λ converges to 0, the generated samples are located within the original distribution rather than the borderline. Even though the images with the same index in Fig. 9(a) and (b) are generated with the

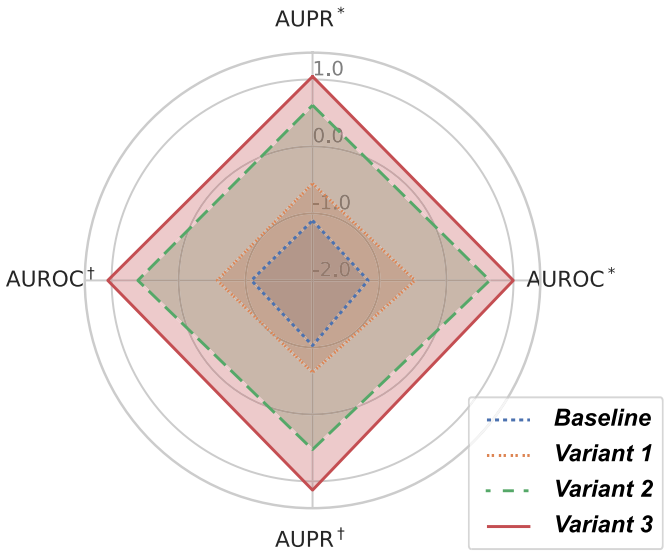

Fig. 10. Radar chart for ablation comparison of classifier performance on CIFAR10. Scores are from the validation* and test† sets. For better visualization, each score is normalized with mean and variance of four variants because AUPR and AUROC have different ranges from each other.

same value of z, appearances of the two images with the same index are different from each other. Many of the generated images in Fig. 9(a) appear to be a car (low and round). This figure illustrates that G trained in the initial cooperative interaction phase can generate the ambiguous minority samples that look like majority samples. These ambiguous minority samples are beneficial to the expansion of the minority region. However, as λ converges to zero, the generated images become similar to truck image (high and box-style), as shown in Fig. 9(b).

As most data-level sampling methods provide samples only in the inner region of the training data distribution, they risk overfitting [57]. In contrast, we can observe that several samples generated by our method are positioned over the decision boundary between two classes. This result implies that the proposed method can expand the minority region to improve the generalization performance of C on the minority class. After the regularization term vanishes by reducing λ to almost zero, the generated samples cover a wide region of the minority class, as shown in the fourth map of Fig. 8.

3) Ablation Study: The ablation study was conducted with CIFAR10 by sequentially adding each ablation component because each component could not be implemented without the previous components. The role of the components is validated through an ablation study on CIFAR10 through ablation of one baseline and three variants as listed in following table:

Baseline	w/o $\mathbf{x}_g^{\mathrm{mi}}$	w/o $\mathcal{U}_{c_2}(G,C)$	w/o $\mathcal{R}(G,C)$
Variant 1	w/ $\mathbf{x}_g^{\mathrm{mi}}$	w/o $\mathcal{U}_{c_2}(G,C)$	w/o $\mathcal{R}(G,C)$
Variant 2	w/ $\mathbf{x}_g^{\mathrm{mi}}$	w/ $\mathcal{U}_{c_2}(G,C)$	w/o $\mathcal{R}(G,C)$
Variant 3	w/ $\mathbf{x}_g^{\mathrm{mi}}$	w/ $\mathcal{U}_{c_2}(G,C)$	w/ $\mathcal{R}(G,C)$

Note[1] w/o: without, w/ : with.
Note[2] $\mathbf{x}_g^{\mathrm{mi}}$: generated minority samples.
Note[3] Variant 3 is the proposed one. Baseline uses only C.

Fig. 10 shows the results of the ablation study. First, on Variant 1 (orange line), performance improves slightly compared to learning using only C (blue line). As this variant corresponds to the existing cGAN, the amount of improvement is not significant. On Variant 2 (green line), a significant improvement of performance is achieved in addition to the first ablation (green line). This implies the terms for joint training of G and C along with alternating training contributes to both G and C so that C helps G generate samples beneficial to C, consequently improving C's performance. Finally, when the $\mathcal{R}(G,C)$ term was added as Variant 3, it significantly improved since G generated samples to interactively expand the minority region (red line).

D. Comparative Analysis

To verify the validity of the proposed method, we compared the classification performance based on four metrics to existing techniques using five configurations from four data sets.

1) Compared Methods: For the conventional data-level methods, we adopted 11 methods described in Section II-A. For implementing SMOTE [6], B-SMOTE [12], ADASYN [13], C-Centroids [11], CN-Neighbor [11], and SMOTE-ENN [17], we used imbalanced-learn library [58]. For MWMOTE [7], NRSB-SMOTE [35], SMOTE-IPF [18], and G-SMOTE [15], we used smote-variants library [59]. For RSNO [16], we acquired MATLAB code from the authors. However, because all the conventional data-level methods support CPU computation only, we could not conduct some of experiments on high-dimensional and large number of samples (marked by "–" in Table III). The compared loss-based methods are CRL [36], MPL [37], and focal loss [38]. GAN-based techniques were compared to three other methods. The first method is based on cGAN, which is used in most GAN-based approaches. The structure of cGAN is the same as that used in our work. The second method is BAGAN [31]. The authors of BAGAN released the source code, and the structure and hyperparameters specified in their paper were used. The third GAN-based method is TripleGAN [33], E-TripleGAN [41], and HexaGAN [34] use the concept of TripleGAN for imbalanced data problem.

2) Hyperparameters and Experimental Settings: For a fair comparison, hyperparameters of the classifier for each data set are searched for the classifier only (baseline) case. Then, the same set of classifier's hyper-parameters was used for the others. Besides, the unique hyperparameters of each method, such as γ of focal loss [38], were searched within a specific range following their guidelines and selected with the values that showed the best validation performance. In the case of GAN-based techniques, the same structure of G and D was used, except for BAGAN having its own structure. Further training details about network structures and hyperparameter values are provided in Appendix B.

3) Comparison Results: The comparative results are listed in Tables III–V. Our method outperformed all the compared methods on all the data sets consistently. Most GAN-based methods tend to give consistent improvements against the baseline "classifier-only" on all data sets. Some loss-based

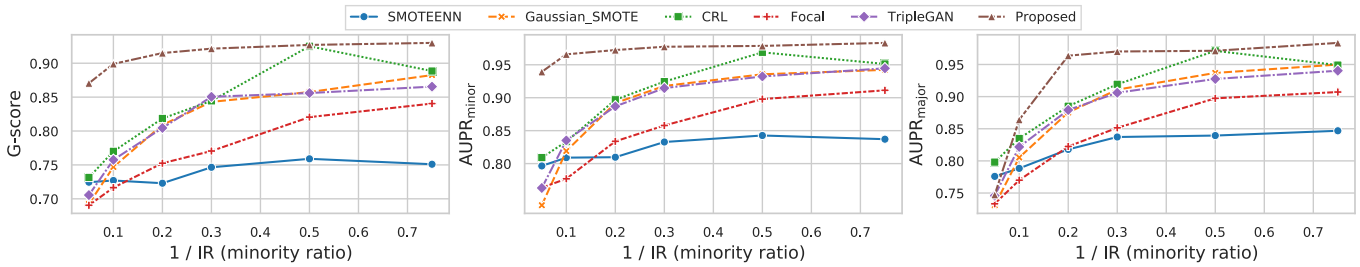

Fig. 11. Performance changes of representative methods depending on IR changes.

TABLE III
TEST SET PERFORMANCE OF BINARY-CLASS DATA SETS

	Methods	CIFAR (Car vs. Truck) B-ACC	AUPR$_{mi}$	AUPR$_{mj}$	AUROC	G-score	ImageNet (Dog vs. Cat) B-ACC	AUPR$_{mi}$	AUPR$_{mj}$	AUROC	G-score	Dementia B-ACC	AUPR$_{mi}$	AUPR$_{mj}$	AUROC	G-score
	Classifier Only	0.7245	0.7880	0.7878	0.7972	0.7245	0.6714	0.4851	0.9888	0.8344	0.6021	0.9200	0.8792	0.9924	0.9668	0.9200
Conventional data-level	SMOTE	0.7165	0.7957	0.7425	0.7901	0.7165	0.5217	0.0717	0.9697	0.6071	0.3084	0.9130	0.8986	0.9923	0.9661	0.9130
	B-SMOTE	0.7215	0.7991	0.7539	0.7956	0.7215	0.5145	0.0658	0.9723	0.6137	0.2847	0.9151	0.9018	0.9916	0.9652	0.9151
	ADASYN	0.7195	0.7909	0.7515	0.7917	0.7195	0.5197	0.0630	0.9680	0.5915	0.3152	0.8980	0.8733	0.9909	0.9579	0.8980
	C-Centroids	0.5955	0.6214	0.5873	0.6293	0.5955	0.5197	0.0630	0.9680	0.5915	0.3152	0.9103	0.8825	0.9933	0.9670	0.9103
	CN-Neighbor	0.6505	0.6906	0.6906	0.7009	0.6505	0.5060	0.1519	0.9769	0.6806	0.2238	0.9247	0.9074	0.9927	0.9701	0.9247
	SMOTEENN	0.7245	0.7965	0.7835	0.7997	0.7245	0.5270	0.0545	0.9617	0.5443	0.3313	0.9128	0.9017	0.9928	0.9674	0.9128
	MWMOTE	0.6865	0.7660	0.7471	0.7659	0.6865	–	–	–	–	–	0.9198	0.8992	0.9924	0.9664	0.9198
	NRSB	0.6945	0.7591	0.7442	0.7610	0.6945	–	–	–	–	–	0.9198	0.9198	0.9926	0.9712	0.9198
	SMOTE-IPF	0.4815	0.4905	0.4776	0.4754	0.4815	–	–	–	–	–	0.9293	0.9230	0.9931	0.9728	0.9293
	G-SMOTE	0.6925	0.7368	0.7314	0.7499	0.6925	–	–	–	–	–	0.9331	0.9303	0.9926	0.9748	0.9331
	RSNO	–	–	–	–	–	–	–	–	–	–	0.9249	0.9178	0.9900	0.9688	0.9249
Loss based	CRL	0.7615	0.8092	**0.7981**	0.8179	0.7615	0.6521	0.4669	0.9870	0.8498	0.5975	0.9247	0.9163	0.9902	0.9732	0.9247
	focal	0.7405	0.7934	0.7833	0.8084	0.7405	0.6764	0.4991	0.9904	0.8456	0.6107	0.9128	0.8842	0.9915	0.9604	0.9128
	MPL	0.7310	0.7973	0.7886	0.8098	0.7310	0.6751	0.4975	0.9918	0.8711	0.6081	0.9276	0.9188	0.9928	0.9730	0.9276
GAN based	cGAN	0.7390	0.8041	0.7559	0.8045	0.7390	0.7142	0.6121	0.9928	0.8866	0.6655	0.9192	0.9296	0.9937	0.9749	0.9192
	BAGAN	0.7035	0.7600	0.7557	0.7687	0.7035	0.7012	0.6001	0.9902	0.877	0.6551	0.9183	0.9116	0.9922	0.9706	0.9183
	Triple	0.7455	0.7930	0.7480	0.8162	0.7455	0.7293	0.6263	**0.9958**	**0.9247**	0.6874	0.9179	0.9270	**0.9938**	0.9746	0.9179
	E-TripleGAN	0.7455	0.7943	0.7531	0.8170	0.7455	0.6456	0.1648	0.9782	0.6961	0.6371	0.9175	0.9265	0.9927	0.9742	0.9171
	Proposed	**0.8705**	**0.9392**	0.7643	**0.8907**	**0.8705**	**0.7649**	**0.6761**	**0.9962**	**0.9322**	**0.7354**	**0.9348**	**0.9393**	**0.9940**	**0.9766**	**0.9348**

For the cases of –, their codes could not be conducted on the cases of high dimension and larger number of samples because they are available only at CPU and require extremely heavy computation. In Delong *et. al.* AUROC test, the proposed method shows statistically significant improvement against all the methods except the underlined cases on Dementia dataset.

TABLE IV
TEST SET PERFORMANCE OF MULTICLASS CIFAR10

Methods		CIFAR10 (All classes) B-ACC	AUPR$_{mi}$	AUPR$_{mj}$	AUROC	G-score
Classifier Only		0.5770	0.5844	0.7872	0.9267	0.7415
Conventional data-level	SMOTE	0.5508	0.5550	0.7745	0.9196	0.7234
	B-SMOTE	0.5460	0.5671	0.7697	0.9208	0.7200
	ADASYN	0.5575	0.5555	0.7587	0.9155	0.7281
	C-Centroids	0.1205	0.2132	0.2224	0.7114	0.3297
	CN-Neighbor	0.5101	0.5043	0.6635	0.8925	0.6945
	SMOTE-ENN	0.3867	0.3961	0.4060	0.8225	0.6003
	MWMOTE	0.5591	0.5677	0.7873	0.9223	0.7292
	NRSB-SMOTE	0.5736	0.5954	0.7888	0.9275	0.7392
	SMOTE-IPF	0.1540	0.1611	0.1634	0.5941	0.3735
	G-SMOTE	0.5983	0.6000	0.7939	0.9285	0.7560
Loss-based	CRL	0.5529	0.4612	0.7638	0.8648	0.7249
	Focal	0.5735	0.5847	0.7184	0.9202	0.7391
	MPL	0.5474	0.5767	0.7484	0.9192	0.7210
GAN-based	cGAN	0.5145	0.4703	0.7085	0.8977	0.6977
	BAGAN	0.5793	0.5756	0.7725	0.9019	0.7480
	TripleGAN	0.6008	0.6063	0.7993	0.9149	0.7577
	E-TripleGAN	0.5110	0.5098	0.7100	0.8878	0.6952
	Proposed	**0.6268**	**0.6120**	**0.8133**	**0.9329**	**0.7751**

TABLE V
TEST SET PERFORMANCE OF MULTILABEL CELEBA

Methods		CelebA B-ACC	AUPR$_{mi}$	AUPR$_{mj}$	AUROC	G-score
Classifier Only		0.8440	0.7173	0.9673	0.9168	0.8440
Loss-based	CRL	0.8495	0.6975	0.9635	0.9200	0.8495
	Focal	0.8411	0.6869	0.9652	0.9079	0.8411
	MPL	0.8326	0.6723	0.9642	0.8630	0.8410
GAN-based	cGAN	0.8528	0.7178	0.9650	0.9114	0.8528
	TripleGAN	0.8479	0.7062	0.9670	0.9133	0.8479
	E-TripleGAN	0.8481	0.7012	0.9665	0.9111	0.8576
	Proposed	**0.8583**	**0.7295**	**0.9706**	**0.9222**	**0.8583**

In Delong *et. al.* AUROC test, the proposed method shows statistically significant improvement against all the methods except CRL giving the underlined score.

(cost-sensitive) methods give improvements on most data sets except the multiclass data set (CIFAR10). However, most of the data-level methods do not give improvements on high-dimensional image data and show improvements only on low-dimensional table data (Dementia). As shown in the underlined scores of Table III, in Delong *et al.* AUROC test, the proposed method shows statistically significant improvement against all the methods except the underlined cases on the Dementia data set.

In addition, we investigated the trend of performance change with IR variation for the representative methods of data-level, loss-based, and GAN-based approach. Every data set except CIFAR10 has fixed IR. Thus, we used CIFAR10 to see the change in performance (G-score, AUPR$_{mi}$, and AUPR$_{mj}$) according to IR change, which is shown in Fig. 11. In G-score, the proposed method outperforms the other methods consistently in the most range of IR.

When IR = 20, AUPR of majority class of our method is slightly degraded instead of remarkably improving AUPR of minority class as shown in the right two graphs in Fig. 11.

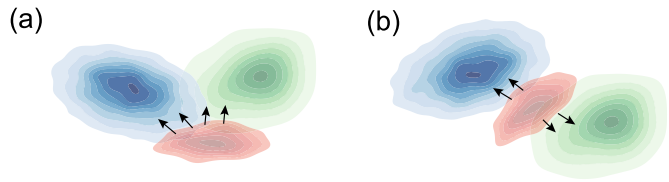

Fig. 12. Two possible cases, (a) easy and (b) hard, for a multiclass problem.

In the range of IR less than 10, our method shows outperforming AUPR for both majority and minority classes.

4) Discussion on Performance Improvements: Although our method consistently outperforms comparative methods, the amount of improvement depends on the data set. Here, we discuss possible causes of the improvement differences.

1) *High Baseline:* The dementia data have small IR and low dimensions. Small IR mitigates the difficulty of imbalanced data problem, and low dimension alleviates the curse of dimensionality. Thus, the baseline accuracy is already high and so the room for improvement is limited.
2) *Complex Decision Boundary in Multiclass Problem:* As the number of classes increases in the multiclass problem, the decision boundary generally becomes more complex. As shown in Fig. 12(a), it is relatively easy to decide the direction of expansion. However, for the case shown in Fig. 12(b), because the expansion directions are opposite to each other, a complex decision boundary in the multiclass problem might provoke ineffective expansion.
3) *Unrealistic Sample Generation in Multilabel Problem:* Since the multilabel vectors of generated samples are sampled randomly, it has a possibility to sample an unrealistic combination of labels such as women-with-mustache. Because this unrealistic image does not exist in the test data, it may not contribute to or even be harmful to the performance. However, this phenomenon inevitably occurs when balancing for the multilabel case.

V. Conclusion

To overcome the difficulty of imbalanced data learning, we proposed a novel methodology based on a three-player game and decision boundary regularization. First, we designed a three-player structure to improve imbalanced data learning performance and analyzed the equilibrium point of the proposed utility function. Second, we introduced a decision boundary regularization to expand the minority region determined by the trained classifier with samples generated by the generator in our three-player structure. Third, we proposed an alternating training scheme to effectively train the three-player structure, in cooperation with the decision boundary regularization. The experiment illustrated that the proposed method outperforms the existing methods by yielding abundant samples to expand the minority decision region, which is beneficial in addressing imbalanced data learning problems.

Although the proposed method showed promising results, certain issues remain. In this study, with a relatively simple form of cGAN (DCGAN), the proposed method achieved a considerable performance improvement in imbalanced data classification. As further work, the use of more precise generators and discriminators such as Wasserstein GAN [60] is expected to yield higher and stable performance for the imbalanced data learning. In addition, for the λ decay schedule, an exponential decay rule was used where the decaying degree was empirically determined depending on the data sets. For further improvement, an elaborate design or adaptive scheme for λ decay should be adopted, which would consider the IR and complexity of the target data set. As open problems, interpretability and authenticity are critical topics to be pursued in a machine learning field. Recently, several types of research have been proposed to interpret GANs [61]–[64]. By applying them to our method, we can take one step closer to solving the open problems.

Appendix A
Proof of Theorem 1

The proof of the following Lemma 1 is equivalent to the proof[1] the original GAN [30], and thereby, we briefly summarize the original proof by rewriting it. For the details, refer to the reference in a footnote. Here, we add Theorem 1 for the proof of the three-player game proposed in this article.

Lemma 1: For any fixed G in $\mathcal{U}_g(D, G)$, the optimal discriminator D is given by

$$D^*(\mathbf{x}, y) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x}, y) + p_g(\mathbf{x}, y)}. \tag{11}$$

Proof: Given G, $\mathcal{U}_g(D, G)$ can be rewritten as

$$\mathcal{U}_g(D, G) = \int\int p(\mathbf{x}, y) \log D(\mathbf{x}, y) dy d\mathbf{x} + \int\int p_g(\mathbf{x}, y) \log(1 - D(\mathbf{x}, y)) dy d\mathbf{x}. \tag{12}$$

This function achieves the maximum at $((p(\mathbf{x}, y))/(p(\mathbf{x}, y) + p_g(\mathbf{x}, y)))$. □

Theorem 1: For given D^*, the equilibrium of $\mathcal{U}(C, D, G)$ is achieved if and only if

$$p(\mathbf{x}, y) = p_g(\mathbf{x}, y) = p_c(\mathbf{x}, y) = p_c(G(\mathbf{z}|y), y). \tag{13}$$

Proof: Given D^*, we can reformulate the minimax game with value function $\mathcal{U}_g(D, G)$ as

$$\mathcal{U}_g(D, G) = \int\int p(\mathbf{x}, y) \log \frac{p(\mathbf{x}, y)}{p(\mathbf{x}, y) + p_g(\mathbf{x}, y)} dy d\mathbf{x} + \int\int p_g(\mathbf{x}, y) \log \frac{p_g(\mathbf{x}, y)}{p(\mathbf{x}, y) + p_g(\mathbf{x}, y)} dy d\mathbf{x}. \tag{14}$$

Following the proof in GAN, $\mathcal{U}_g(D, G)$ can be rewritten as

$$\mathcal{U}_g(D, G) = -\log 4 + 2\mathrm{JSD}(p(\mathbf{x}, y) || p_g(\mathbf{x}, y)) \tag{15}$$

[1] https://srome.github.io/An-Annotated-Proof-of-Generative-Adversarial-Networks-with-Implementation-Notes/

where JSD is the Jensen–Shannon divergence. In addition, according to the definition of Kullback–Leibler (KL) divergence, $\mathcal{U}_c(C, G)$ can be rewritten as

$$\begin{aligned}\mathcal{U}_c(C, G) &= \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[-\log p_c(y|\mathbf{x})] \\ &\quad + \mathbb{E}_{(\mathbf{x},y)\sim p_g(\mathbf{x},y)}[-\log p_c(y|G(\mathbf{z}|y))] \\ &= D_{\mathrm{KL}}(p(\mathbf{x}, y)||p_c(\mathbf{x}, y)) + H_p(y|\mathbf{x}) \\ &\quad + D_{\mathrm{KL}}(p_g(\mathbf{x}, y)||p_c(G(\mathbf{z}|y), y)) \\ &\quad + H_{p_g}(y|G(\mathbf{z}|y)). \end{aligned} \tag{16}$$

From (15) and (16), $\mathcal{U}(C, D, G)$ becomes

$$\begin{aligned}\mathcal{U}(C, D, G) &= \mathcal{U}_g(D, G) + \mathcal{U}_c(C, G) \\ &= 2\mathrm{JSD}(p(\mathbf{x}, y)||p_g(\mathbf{x}, y)) \\ &\quad + D_{\mathrm{KL}}(p(\mathbf{x}, y)||p_c(\mathbf{x}, y)) \\ &\quad + D_{\mathrm{KL}}(p_g(\mathbf{x}, y)||p_c(G(\mathbf{z}|y), y)) \\ &\quad +, (H_p(y|\mathbf{x}) + H_{p_g}(y|G(\mathbf{z}|y)) - \log 4). \end{aligned} \tag{17}$$

Since $\mathrm{JSD}(\cdot)$ and $D_{\mathrm{KL}}(\cdot)$ are nonnegative, their minimum values become zero if and only if $p(\mathbf{x}, y) = p_g(\mathbf{x}, y)$, $p(\mathbf{x}, y) = p_c(\mathbf{x}, y)$, and $p_g(\mathbf{x}, y) = p_c(G(\mathbf{z}|y), y)$. Hence, the equilibrium of $\mathcal{U}(C, D, G)$ becomes $p(\mathbf{x}, y) = p_g(\mathbf{x}, y) = p_c(\mathbf{x}, y) = p_c(G(\mathbf{z}|y), y)$. □

APPENDIX B
IMPLEMENTATIONS AND HYPERPARAMETERS

For the classifier on CIFAR10 and CelebA data, we used the architecture of Resnet18 [39]. For Dementia data set, we used the same classifier as in the original paper that first used the Dementia data set [48], where the classifier is composed of 2-D convolutional neural networks with skip connection and Hilbert curve transform. Both the classifiers used the softmax activation in the output layer. In the case of GAN-based techniques, the same structure of DCGAN [65] was used for every comparison. However, as BAGAN adopts their own sophisticated architecture, we used their own architecture for BAGAN.

Besides, the unique hyperparameters of each method were searched within a specific range following their guidelines and selected with the values that showed the best validation performance. The list of hyperparameters and their ranges is as follows.

1) The number of nearest neighbor, k, of conventional data-level methods: [5, 10].
2) The ratio of CRL loss: [0.1, 0.8].
3) The γ of Focal loss: [2, 4].
4) The number of triplet, k, of CRL loss: [10, 40].
5) The learning rate of BAGAN: [0.00005, 0.0005].
6) The α of tripleGAN: [0.5, 1].

The common training details for every case are as follows.

1) Batch Size: 128.
2) Optimizer: ADAM (with β_1: 0.5, β_2: 0.999, and learning rate: 0.0002).
3) Weight initialization: Xavier normalization with mean = 0, std = 0.02.
4) The number of patient for early stop: 30.

REFERENCES

[1] R. B. Rao, S. Krishnan, and R. S. Niculescu, "Data mining for improved cardiac care," *ACM SIGKDD Explor. Newslett.*, vol. 8, no. 1, pp. 3–10, Jun. 2006.

[2] X.-M. Zhao, X. Li, L. Chen, and K. Aihara, "Protein classification with imbalanced data," *Proteins, Struct., Function, Bioinf.*, vol. 70, no. 4, pp. 1125–1132, Dec. 2007.

[3] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intell. Syst.*, vol. 14, no. 6, pp. 67–74, Nov. 1999.

[4] C. Phua, D. Alahakoon, and V. Lee, "Minority report in fraud detection: Classification of skewed data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 50–59, Jun. 2004.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[7] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[8] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 1999, pp. 155–164.

[9] K. M. Ting, "An instance-weighting method to induce cost-sensitive trees," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 3, pp. 659–665, May 2002.

[10] A. Fernández, V. López, M. Galar, M. J. del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowl.-Based Syst.*, vol. 42, pp. 97–110, Apr. 2013.

[11] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, Apr. 2009.

[12] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Germany: Springer, 2005, pp. 878–887.

[13] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.

[14] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on K-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018, doi: 10.1016/j.ins.2018.06.056.

[15] H. Lee, J. Kim, and S. Kim, "Gaussian-based SMOTE algorithm for solving skewed class distributions," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 17, no. 4, pp. 229–234, Dec. 2017.

[16] X. Tao *et al.*, "Real-value negative selection over-sampling for imbalanced data set learning," *Expert Syst. Appl.*, vol. 129, pp. 118–134, Sep. 2019.

[17] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.

[18] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci.*, vol. 291, pp. 184–203, Jan. 2015.

[19] J. Hu, X. He, D.-J. Yu, X.-B. Yang, J.-Y. Yang, and H.-B. Shen, "A new supervised over-sampling algorithm with application to protein-nucleotide binding residue prediction," *PLoS One*, vol. 9, no. 9, pp. 1–10, Sep. 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0107676

[20] C. Wang, Z. Yu, H. Zheng, N. Wang, and B. Zheng, "CGAN-plankton: Towards large-scale imbalanced class generation and fine-grained classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 855–859.

[21] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, Jan. 2018.

[22] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Proc. Adv. Knowl. Discovery Data Mining*, D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, and L. Rashidi, Eds. Cham, Switzerland: Springer, 2018, pp. 349–360.

[23] W. Gao, L. Wang, R. Jin, S. Zhu, and Z.-H. Zhou, "One-pass AUC optimization," *Artif. Intell.*, vol. 236, pp. 1–29, Jul. 2016.

[24] J. Hu, H. Yang, M. R. Lyu, I. King, and A. M.-C. So, "Online nonlinear AUC maximization for imbalanced data sets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 882–895, Apr. 2018.

[25] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "A novel ensemble method for imbalanced data learning: Bagging of extrapolation-SMOTE SVM," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–11, Jan. 2017.

[26] M. Khalilia, S. Chakraborty, and M. Popescu, "Predicting disease risks from highly imbalanced data using random forest," *BMC Med. Informat. Decis. Making*, vol. 11, no. 1, Jul. 2011.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.

[28] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. Comput. Vis. (ECCV)*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 404–417.

[29] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[30] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[32] Y. Zhang, "Deep generative model for multi-class imbalanced learning," M.S. thesis, Dept. Elect. Eng., Univ. Rhode Island, Kingston, RI, USA, 2018.

[33] L. I. Chongxuan, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, *et al.* Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 4088–4098.

[34] U. Hwang, D. Jung, and S. Yoon, "HexaGAN: Generative adversarial nets for real world classification," in *Proc. 36th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA, Jun. 2019, pp. 2921–2930.

[35] F. Hu and H. Li, "A novel boundary oversampling algorithm based on neighborhood rough set model: NRSBoundary-SMOTE," *Math. Problems Eng.*, vol. 11, pp. 1–10, Jan. 2013.

[36] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1851–1860.

[37] S. R. Bulo, G. Neuhold, and P. Kontschieder, "Loss max-pooling for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7082–7091.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

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

[40] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 2642–2651. [Online]. Available: https://arxiv.org/abs/1610.09585

[41] S. Wu, G. Deng, J. Li, R. Li, Z. Yu, and H.-S. Wong, "Enhancing TripleGAN for semi-supervised conditional instance synthesis and classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10091–10100.

[42] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge distillation with adversarial samples supporting decision boundary," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Jul. 2019, pp. 3771–3778.

[43] K. Sun, Z. Zhu, and Z. Lin, "Enhancing the robustness of deep neural networks by boundary conditional GAN," 2019, *arXiv:1902.11029*. [Online]. Available: http://arxiv.org/abs/1902.11029

[44] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," 2017, *arXiv:1711.09325*. [Online]. Available: http://arxiv.org/abs/1711.09325

[45] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[46] G. Raskutti, M. J. Wainwright, and B. Yu, "Early stopping for non-parametric regression: An optimal data-dependent stopping rule," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, Berlin, Germany: Springer-Verlag, Sep. 2011, pp. 1318–1325.

[47] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[48] H.-S. Choi *et al.*, "Deep learning based low-cost high-accuracy diagnostic framework for dementia using comprehensive neuropsychological assessment profiles," *BMC Geriatrics*, vol. 18, no. 1, p. 234, Oct. 2018, doi: 10.1186/s12877-018-0915-z.

[49] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[50] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[51] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[52] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.

[53] M. Kubat, R. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Mach. Learn.*, vol. 30, nos. 2–3, pp. 195–215, Dec. 1998.

[54] V. García, R. A. Mollineda, and J. S. Sánchez, "Index of balanced accuracy: A performance measure for skewed class distributions," in *Proc. 4th Iberian Conf. Pattern Recognit. Image Anal.* Berlin, Germany: Springer-Verlag, Jun. 2009, p. 441–448.

[55] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, 1988.

[56] L. van der Maaten, "Learning a parametric embedding by preserving local structure," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 5, D. van Dyk and M. Welling, Eds. Clearwater Beach, FL, USA: Hilton Clearwater Beach Resort, Apr. 2009, pp. 384–391.

[57] N. V. Chawla, *Data Mining for Imbalanced Datasets: An Overview*. Boston, MA, USA: Springer, 2005, pp. 853–867, [Online]. Available: https://doi.org/10.1007/0-387-25465-X_40

[58] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.

[59] G. Kovács, "Smote-variants: A Python implementation of 85 minority oversampling techniques," *Neurocomputing*, vol. 366, pp. 352–354, Nov. 2019.

[60] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70. Aug. 2017, pp. 214–223, [Online]. Available: http://proceedings.mlr.press/v70/arjovsky17a.html

[61] D. Bau *et al.*, "Gan dissection: Visualizing and understanding generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–5.

[62] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9243–9252.

[63] A. Jahanian, L. Chai, and P. Isola, "On the'steerability' of generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 214–223.

[64] D. Jung, J. Lee, J. Yi, and S. Yoon, "Icaps: An interpretable classifier via disentangled capsule networks," in *Proc. Comput. Vis. (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 314–330.

[65] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: https://arxiv.org/abs/1511.06434

Hyun-Soo Choi received the B.S. degree in computer and communication engineering (first major) and in brain and cognitive science (second major) from Korea University, Seoul, South Korea, in 2013, and the integrated M.S./Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, in 2019.

Since February 2020, he has been a Senior Researcher with Vision AI Labs, SK Telecom. Since March 2021, he has been working at the Department of Computer Science and Engineering, Kangwon National University, South Korea.

Dahuin Jung received the B.S. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 2016. She is currently pursuing the integrated M.S./Ph.D. degree in electrical and computer engineering with Seoul National University, Seoul, South Korea.

Her research interests include deep learning, representation learning, and explainable artificial intelligence (AI).

Siwon Kim received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2018, where she is currently pursuing the integrated M.S./Ph.D. degree in electrical and computer engineering.

Her research interests include artificial intelligence, deep learning, and biomedical applications.

Sungroh Yoon (Senior Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2002 and 2006, respectively.

He was a Visiting Scholar with the Department of Neurology and Neurological Sciences, Stanford University, from 2016 to 2017. He held research positions at Stanford University and Synopsys, Inc., Mountain View, CA, USA. From 2006 to 2007, he was with Intel Corporation, Santa Clara, CA, USA. He was an Assistant Professor with the School of Electrical Engineering, Korea University, Seoul, from 2007 to 2012. He is currently a Professor with the Department of Electrical and Computer Engineering, Seoul National University. His current research interests include machine learning and artificial intelligence.

Dr. Yoon was a recipient of the SNU Education Award, in 2018, the IBM Faculty Award, in 2018, the Korean Government Researcher of the Month Award in 2018, the BRIC Best Research of the Year in 2018, the IMIA Best Paper Award in 2017, the Microsoft Collaborative Research Grant in 2017 and 2020, the SBS Foundation Award in 2016, the IEEE Young IT Engineer Award in 2013, and many other prestigious awards. Since February 2020, he has been serving as the Chairperson for the Presidential Committee on the Fourth Industrial Revolution established by the Korean Government.