Discriminative-Generative Positive and Unlabeled Learning

Botai Yuan, Chen Gong, *Senior Member, IEEE*, Dacheng Tao, *Fellow, IEEE*, and Jie Yang, *Senior Member, IEEE*

***Abstract*—Positive and Unlabeled (PU) learning aims to train a suitable classifier simply based on a set of positive data and unlabeled data. Existing PU methods usually follow a discriminative framework and yield limited classification performance, because the lack of explicit negative labels poses a great barrier in training a discriminative PU model. To address the challenge of limited supervisory information faced by discriminative PU methods, this paper introduces generative operation to PU learning in addition to the conventional discriminative operation, and proposes a novel algorithm dubbed "Discriminative-Generative Positive and Unlabeled Learning" (DGPU). Specifically, our proposed DGPU consists of a data generation stage and a discriminative annotation stage, which can benefit from each other in an iterative manner. In data generation stage, we employ a tailored diffusion model to generate high-quality negative examples and positive examples to efficiently enrich the supervisory information. In discriminative annotation stage, the classifier is further refined on the initial and generated training data. To the best of our knowledge, this study represents the first attempt to integrate diffusion models into PU learning to make generative model and discriminative model benefit from each other in a collaborative way. Thanks to this, our proposed DGPU significantly outperforms existing PU methods across a wide range of synthetic and real-world benchmark datasets. In particular, our DGPU is almost comparable to the fully supervised counterpart, and improves the test accuracy of existing state-of-the-art methods by 3.89% and 2.56% on CIFAR-10 and CelebA datasets, respectively.**

***Index Terms*—Positive and unlabeled learning, diffusion models, semi-supervised learning.**

I. Introduction

POSITIVE and Unlabeled (PU) learning focuses on binary classification task, of which the target is to train a binary classifier based on a set of labeled positive data and unlabeled data [1]. This learning scenario naturally arises in many real-world applications such as disease diagnosis [2], [3], hyperspectral image classification [4], anomaly detection [5], etc.

Given the intensive practical demands as listed above, PU learning has attracted a great deal of research attention in recent years. Numerous PU algorithms have been developed over the past decades [1], [6], [7], [8]. Early works usually follow a two-step strategy [9], [10], which first identifies reliable negatives from unlabeled data, and then uses (semi-)supervised learning to train a binary classifier with the reliable negative examples and labeled positive examples. Another prevalent research line is to formulate PU learning as a cost-sensitive learning problem [1], [8], [11], which directly treats unlabeled data as negative data with modified importance weights and has achieved state-of-the-art performance. Both the two-step methods and cost-sensitive methods follow discriminative framework, which aims to learn an optimal decision boundary for separating the positive and negative classes. However, in PU learning context, the training set does not contain explicit negative examples, which presents significant challenges for these discriminative methods to accurately acquire the decision boundary, therefore impairing the performance of the established PU classifier.

Some recent studies have employed generative models, such as Generative Adversarial Networks (GANs) [12], [13] and variational autoencoder (VAE) [14], to address this limitation [15], [16], [17], [18], [19], [20]. For instance, GenPU [18] and CGenPU [15] utilize the framework of adversarial learning to train the generator for negative examples, which are then combined with labeled positive data to train a binary classifier. However, two critical limitations have constrained their effectiveness. First, these generative methods fail to harness the power of discriminative PU methods in the process of training generative models. They typically optimize for likelihood or adversarial objectives, neglecting the rich discriminative signals that could refine the learned data distribution. Second, these methods rely heavily on less expressive generators, such as GANs. Despite their success in various contexts, GANs are often hindered by the issues like mode collapse and training instability, which ultimately degrade the quality and diversity of the generated examples.

On the other hand, recent advances in diffusion models offer a promising alternative. Diffusion models have demonstrated an exceptional ability to capture the intricacies of data distributions, yielding high-quality and diverse examples [21],

Received 12 August 2025; revised 10 December 2025; accepted 1 March 2026. Date of publication 16 March 2026; date of current version 20 March 2026. This work was supported in part by NSF of China under Grant 62376153, Grant 62336003, and Grant 12371510; in part by the National Research Foundation, Singapore; and in part by the CyberSG Research and Development Programme Office (CRPO) through the National Cybersecurity Research and Development Programme (NCRP), RIE2025 NCRP Funding Initiative, under Award CRPO-GC1-NTU-002. The associate editor coordinating the review of this article and approving it for publication was Dr. Jianwen Xie. *(Corresponding authors: Jie Yang; Chen Gong; Dacheng Tao.)*

Botai Yuan, Chen Gong, and Jie Yang are with the Institute of Image Processing and Pattern Recognition, School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yuan_botai@sjtu.edu.cn; chen.gong@sjtu.edu.cn; jieyang@sjtu.edu.cn).

Dacheng Tao is with the College of Computing and Data Science, Nanyang Technological University, Singapore (e-mail: dacheng.tao@gmail.com).

Digital Object Identifier 10.1109/TIP.2026.3672381

[22], [23], which provides substantial motivation for incorporating these models to enrich supervisory information in discriminative methods. As a result, to leverage the generative capability of diffusion models as well as harnessing the power of discriminative PU methods, we propose a novel PU learning algorithm dubbed "**D**iscriminative-**G**enerative **P**ositive and **U**nlabeled Learning" (termed "DGPU"), which specifically utilizes diffusion model to generate high-quality examples for classifier training. To be specific, two important stages (*i.e.*, *data generation* and *discriminative annotation*) in our DGPU alternate, so that they can benefit from each other via an iterative manner. For algorithm initialization, a preliminary classifier is trained on the PU dataset with a discriminative PU algorithm to assign pseudo-labels to unlabeled examples. In data generation stage, a conditional diffusion model is trained on both the pseudo-labeled examples and the initial labeled positive examples to generate an additional set of synthetic positive examples and negative examples. In discriminative annotation stage, the initial classifier is further refined in a semi-supervised manner on the augmented labeled set (including both positive examples and negative examples) and all unlabeled examples, and the pseudo-labeled examples generated by the classifier are updated for training the diffusion model in the next round. To mitigate the negative impact of low-quality synthetic examples, we reweight each synthetic example based on the output of the classifier from last round. As a result, the low-quality synthetic examples would receive less attention during the training process. As such, a well-trained classifier obtained in the discriminative annotation stage could guide the diffusion model in generating precise and high-fidelity images, which will in turn facilitate the further improvement of the classifier.

To the best of our knowledge, this is the first work introducing diffusion models to PU learning to make generative model and discriminative model benefit from each other in a collaborative way. Consequently, our DGPU yields superior performance to existing typical PU learning approaches on various PU benchmark datasets. For example, our method achieves error reductions of 3.89% on CIFAR-10 [24] and 2.56% on CelebA [25] when compared with the existing top-level PU methods. Remarkably, the performance of our DGPU for PU learning is even comparable to its fully-supervised counterpart on many datasets.

The rest of the paper is structured as follows. In Section II, we briefly review some related works to our model, consisting of PU learning and diffusion models. In Section III, we give an overview of framework and then provide the details of the proposed method. Then, we present the experimental results in Section IV. Finally, we conclude our paper in Section V.

II. Related Work

In this section, we review the prior works relevant to this paper, including PU learning and diffusion models.

A. Positive and Unlabeled Learning

PU learning is proposed for the setting where only positive and unlabeled data are accessible for training, and the unlabeled data may include both positive and negative examples [26] but the real labels are unknown before training. PU learning [27] can be roughly attributed to two-step methods [9], [10], and cost-sensitive methods [2], [8]. The two-step approaches first identify reliable negative examples from the unlabeled set, and then build a classifier on the positive set, reliable negatives and the remaining unlabeled set. Such methods mainly differ in the ways to pick up reliable negative examples. For example, PUbN [28] pretrained a model to recognize some reliable negative examples, and then combined positive risk, negative risk, and unlabeled risk to train the final classifier. HolisticPU [29] assigned pseudo-labels to unlabeled examples by identifying the unique predictive trend of each example.

Such sample selection methods might be vulnerable to the mis-identified negative examples, and the current state-of-the-art result is usually achieved by cost-sensitive methods. The cost-sensitive approaches attempt to construct an unbiased or biased risk estimator by assigning the data points with different importance weights. Since the first unbiased PU (uPU) risk estimator [7] was proposed, numerous works have been done to further enhance the performance [2], [11]. "Loss Decomposition and Centroid Estimation" (LDCE) [30] decomposed the loss function of corrupted negative examples into a label-independent term and a label-dependent term, where only the latter term is influenced by the label noise. Dist-PU [31] pursued the consistency between the distribution of predicted labels and the class prior, which has achieved state-of-the-art performance. While the above PU learning methods assume that whether a positive instance is selected as labeled is irrelevant to its feature representation, namely instance-independent, some instance-dependent approaches were proposed, which assume that the labeling of a positive example depends on its feature. Representative methods include "Learning from Positive and Unlabeled Data with a Selection Bias" (PUSB) [32] and "Labeling Bias Estimation" (LBE) [33]. Recently, "Latent Group-Aware Meta Disambiguation" (LaGAM) [34] incorporated a contrastive learning module to extract the underlying grouping semantics within PU data to produce compact representations.

Existing generative PU learning frameworks, such as GenPU [18] and CGenPU [15], adopt adversarial architectures to synthesize positive and negative examples from unlabeled data. While these approaches successfully model the underlying class-conditional distributions, the interaction between their generator and classifier is either static or one-directional: once the generator is trained, the classifier is optimized on fixed synthetic data without further feedback. As a result, these methods fail to leverage the progressively improving discriminative ability of the classifier to refine the generation process. In contrast, our proposed DGPU introduces a *Discriminative–Generative* collaborative framework that establishes an iterative feedback loop between the classifier and the conditional diffusion model. Within this loop, the classifier provides continuously updated supervision signals to guide the diffusion model, which in turn produces higher-quality and more diverse synthetic examples that facilitate further improvement of the classifier. Moreover, DGPU incorporates the confidence-based pseudo-labeling, the

weighted supervised loss, and the debiased unsupervised loss to stabilize this iterative process, which are absent in GenPU and CGenPU. Together, these designs enable DGPU to achieve superior robustness and performance in PU learning tasks.

B. Diffusion Models

Diffusion models are a representative branch of generative models, which have demonstrated impressive performance in various domains, including image synthesis [35], [36], [37], [38], video generation [39], [40], [41], and audio generation [42], [43]. The origins of diffusion models can be traced back to the pioneering work [44], which introduced the idea of gradually corrupting data with Gaussian noise until it approximates a typical Gaussian distribution, and then learning a reverse process to recover the original data distribution. Ho et al. [21] advanced this concept in Denoising Diffusion Probabilistic Models (DDPMs), which employed a reparameterization trick to simplify the training objective, thereby achieving superior performance in high-resolution image synthesis. Later, Nichol and Dhariwal [45] improved the sampling procedure through better noise scheduling strategies (e.g., cosine noise scheduling), which significantly enhanced sample quality and diversity. Building upon these foundations, Song et al. [46] established a theoretical link between score-based generative models and stochastic differential equations, thereby enhancing both the robustness and efficiency of the sampling process.

Conditional diffusion model has recently attracted intensive attention as it enables to control the generated results according to specific purpose. For example, classifier guidance [47] leveraged the gradient of a classifier to steer the generative process of a trained diffusion model. Classifier-free guidance [48] jointly trained a conditional and an unconditional diffusion model, and combined the resulting conditional and unconditional scores to obtain a trade-off between sample quality and diversity. The work [49] trained the diffusion model conditioned on embeddings by using self-supervised learning techniques.

Despite the significant success of diffusion models in generative tasks, their application to PU learning remains unexplored. Our work, to the best of our knowledge, is the first-ever attempt to introduce diffusion models to PU learning.

III. Methodology

Let $\mathcal{X} \subseteq \mathbb{R}^d$ (d denotes data dimension) and $\mathcal{Y} = \{0, 1\}$ be the input feature space and output label space, respectively, $p(\boldsymbol{x}, y)$ be the underlying joint density of $(\mathcal{X}, \mathcal{Y})$, and $p(\boldsymbol{x})$ be the marginal distribution of $p(\boldsymbol{x}, y)$. Let $\pi_{\mathrm{P}} = p(y = 1)$ be the positive class prior, and $\pi_{\mathrm{N}} = 1 - \pi_{\mathrm{P}}$ be the negative class prior. In PU learning, only a labeled positive set X_{P} and an unlabeled set X_{U} are accessible for training. Here we consider the case-control scenario [27], where X_{P} and X_{U} are given as:

$$X_{\mathrm{P}} = \{(\boldsymbol{x}_i, y_i = 1) | \boldsymbol{x}_i \sim p_{\mathrm{P}}(\boldsymbol{x})\}_{i=1}^{n_{\mathrm{P}}}, \tag{1}$$

$$X_{\mathrm{U}} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \sim p(\boldsymbol{x})\}_{i=1}^{n_{\mathrm{U}}}, \tag{2}$$

where $p_{\mathrm{P}}(\boldsymbol{x})$ denotes the marginal distribution of positive data, and n_{P} and n_{U} are the amounts of labeled positive examples and unlabeled examples, respectively. The aim of PU learning is to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ with the training set $X_{\mathrm{PU}} = X_{\mathrm{P}} \cup X_{\mathrm{U}}$, of which the size is $n = n_{\mathrm{P}} + n_{\mathrm{U}}$.

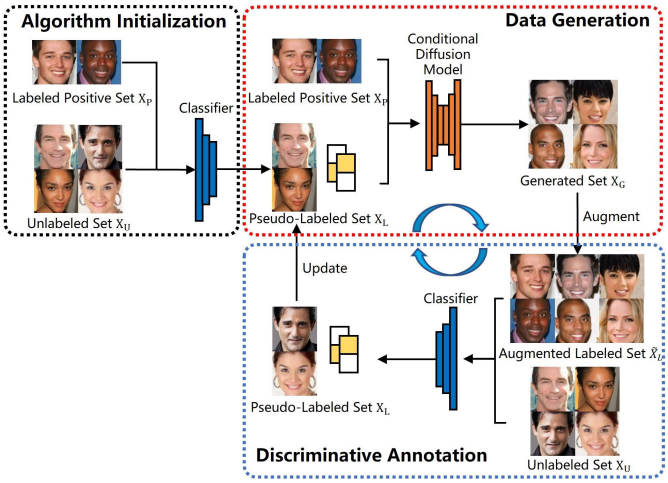


Fig. 1. Pipeline of DGPU. In algorithm initialization, a preliminary classifier is trained on the PU dataset X_{PU} by using a discriminative PU algorithm, so that the examples in X_{L} can be pseudo-labeled. During data generation stage, a conditional diffusion model is trained on a combination of pseudo-labeled set X_{L} and labeled positive set X_{P} to construct the generated set X_{G}, which is subsequently used to augment the labeled set. During discriminative annotation stage, the initial classifier is further refined in a semi-supervised manner on the augmented labeled set $\tilde{X}_{\mathrm{L}}$ and the unlabeled set X_{U}, and the classifier will then be used to update the pseudo-labeled set X_{L}. The data generation stage and discriminative annotation stage work in an iterative manner for collaboration.

Next we will describe our DGPU framework in detail (see Figure 1). Briefly speaking, the training of DGPU iterates between the stages of data generation and discriminative annotation. Initially, a classifier is initialized using a discriminative PU algorithm to assign pseudo-labels to a subset of unlabeled examples. During data generation stage, the pseudo-labeled examples along with the labeled positive examples in X_{P} are utilized to guide a conditional diffusion model in generating high-quality examples to augment the labeled set. During discriminative annotation stage, the initial classifier is further refined using both the augmented labeled examples and unlabeled examples in a semi-supervised manner. This refined classifier is then used to update the pseudo-labeled set, providing a stronger foundation for the next round of data generation. The data generation stage and discriminative annotation stage alternate so that they can benefit from each other in producing a precise PU classifier.

A. Algorithm Initialization

While employing diffusion models to generate high-quality training data provides a promising solution for enriching supervision signals in PU learning, their deployment typically requires sufficient labeled positive and negative examples. This poses a significant challenge in PU settings where labeled positives are scarce and explicit negatives are entirely absent. To address this issue, we adopt a straightforward yet effective strategy: we initialize the classifier using an existing discriminative PU algorithm and employ it to assign pseudo-labels to the unlabeled data.

Specifically, we adopt Dist-PU [31], a state-of-the-art discriminative PU method, to initialize the classifier using both

the labeled positive set X_{P} and the unlabeled set X_{U}. Dist-PU aligns the predicted label distribution with the known class prior, thereby mitigating the negative prediction bias commonly observed in cost-sensitive methods. Let $\mathcal{A}^w(\cdot)$ denote a weak augmentation operator (i.e., random crop and horizontal flip), and compute $\boldsymbol{z}_i = \mathrm{softmax}(f(\mathcal{A}^w(\boldsymbol{x}_i)))$ for each example $\boldsymbol{x}_i$, where z_i^j denotes the j-th entry of $\boldsymbol{z}_i$. The objective function $\mathcal{L}_{\mathrm{align}}$ of Dist-PU is formulated as:

$$\mathcal{L}_{\mathrm{align}} = 2\pi_{\mathrm{P}} \left| \frac{1}{n_{\mathrm{P}}} \sum_{\boldsymbol{x}_i \in X_{\mathrm{P}}} z_i^1 - 1 \right| + \left| \frac{1}{n_{\mathrm{U}}} \sum_{\boldsymbol{x}_i \in X_{\mathrm{U}}} z_i^1 - \pi_{\mathrm{P}} \right|. \tag{3}$$

The classifier obtained during this initialization stage is subsequently used to construct the pseudo-labeled set X_{L}, which provides supervision for training the generative diffusion model. This pseudo-labeled set is iteratively updated in subsequent rounds of discriminative annotation. The detailed construction strategy of X_{L} is described in Section III-C.

B. Data Generation

Based on the constructed pseudo-labeled set, we can train a conditional diffusion model by using a combination of the pseudo-labeled set and the labeled positive set (*i.e.*, $X_{\mathrm{L}} \cup X_{\mathrm{P}}$) to generate a set of synthetic positive examples and negative examples. Diffusion models consist of two processes, namely a forward noising process and a reverse denoising process. In the forward process, we gradually add noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ to data $\boldsymbol{x}_0$ from time $t = 0$ to $t = T$ to obtain a series of noisy latent variables:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\ \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{4}$$

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{5}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ is a pre-defined noise schedule. Given sufficiently large T, the latent x_T is nearly an isotropic Gaussian distribution [21]. The reverse denoising process, parameterized by another Gaussian transition, gradually denoises the latent variables and restores the real data $\boldsymbol{x}_0$ from a Gaussian noise:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)), \tag{6}$$

where θ defines the parameters of the neural network that predicts the mean $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ and the variance $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ of the Gaussian distribution. In denoising diffusion probabilistic models (DDPM) [21], the mean $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ is learned and the variance $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ is set to a constant. The mean $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ can be decomposed as a linear combination of $\boldsymbol{x}_t$ and a noise approximation model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$, namely:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) \right). \tag{7}$$

The training objective of diffusion model is then derived as:

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{\boldsymbol{x}_0, t, \boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|^2], \tag{8}$$

where $\mathbb{E}(\cdot)$ denotes the expectation, and $\|\cdot\|$ corresponds to the Frobenius norm of a tensor.

Classifier-free guidance (CFG) [48] leverages a conditional noise approximation model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y, t)$ and an unconditional noise approximation model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ to enable conditional generation. Formally, CFG restores $\boldsymbol{x}_0$ from a random Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$ by iteratively deploying the following equation:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, y, t) \right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \tag{9}$$

where $\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{x}_t, y, t) = (1+s)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y, t) - s\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$, and s is the guidance strength.

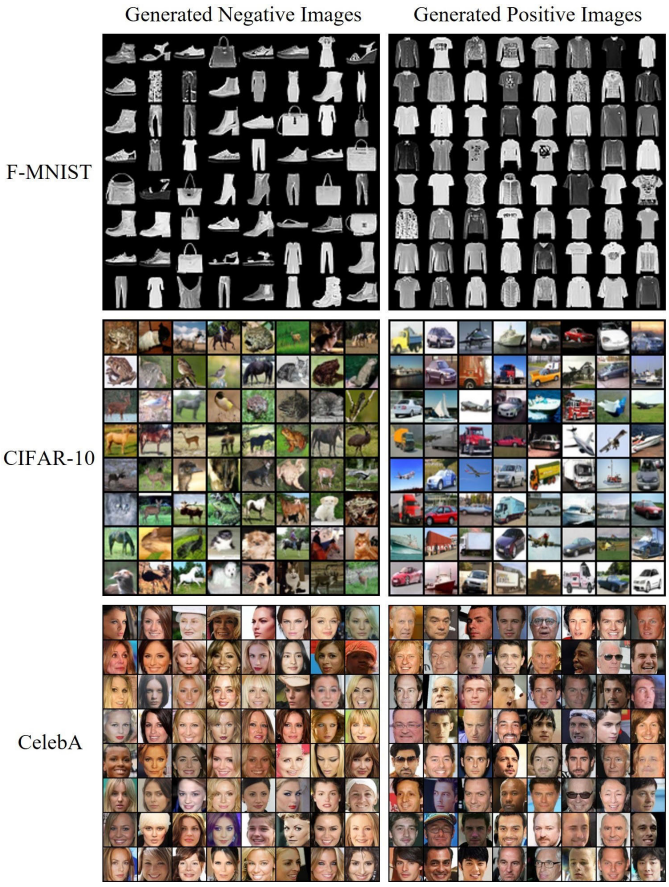


Fig. 2. A visualization of generated images on F-MNIST (top row), CIFAR-10 (middle row), and CelebA (bottom row) datasets with our DGPU.

In our DGPU, the conditional diffusion model can be any well-performing diffusion model in principle. However, considering that the diffusion process can be time-consuming and our DGPU needs to invoke the diffusion model in every iteration, in our work, we adopt EDM [50] for conditional generation, since it can significantly reduce the number of sampling steps during generation and thus can be efficiently implemented. The computational cost analysis can be found in Section IV-I.

1) Generated Set Construction: With the aid of the well-trained conditional diffusion model, we obtain a generated set $X_{\mathrm{G}} = \cup_{y\in\{0,1\}}\{(\hat{\boldsymbol{x}}_{i,y}, y)\}_{i=1}^{K_y}$, where $\hat{\boldsymbol{x}}_{i,y}$ denotes the i-th generated image of class y, and K_y represents the number of generated images for class y. The total number of generated examples is denoted as $n_{\mathrm{G}} = K_0 + K_1$. To ensure that the generated data follow the true class prior π_{P}, we set $K_1 = n_{\mathrm{G}} \times \pi_{\mathrm{P}}$ and $K_0 = n_{\mathrm{G}} \times (1-\pi_{\mathrm{P}})$. A visualization of the generated images is provided in Figure 2, revealing that the generated image examples are very close to the real ones.

C. Discriminative Annotation

Since the above-mentioned data generation stage constructs the generated set composed of high-quality positive and negative data, the original PU learning problem can be reformulated as a Semi-Supervised Learning (SSL) problem, where the initial classifier is further refined on the augmented labeled set $\tilde{X}_{\mathrm{L}} = X_{\mathrm{P}} \cup X_{\mathrm{G}}$ and the unlabeled set X_{U}. Importantly, the classifier is not re-initialized at each round, it is incrementally optimized based on the previously learned parameters instead. However, the incorporation of low-quality images in the generated set would negatively affect the training of the classifier. To mitigate this issue, we leverage the pre-trained classifier from the previous stage. Specifically, we assign a unique weight ω_i to each example $\boldsymbol{x}_i \in \tilde{X}_{\mathrm{L}}$ based on the confidence of the classifier. Let $\hat{f}$ denote the classifier trained in the previous stage, the weight ω_i is determined as $\omega_i = [\max(\mathrm{softmax}(\hat{f}(\mathcal{A}^w(\boldsymbol{x}_i))))]^{1/2}$. Here the square root operation is to introduce a sublinear scaling effect for controlling the degree of emphasis on high-confidence predictions, which has been shown effective in recent studies [51], [52]. The weighted supervised loss $\mathcal{L}_{\mathrm{s}}$ is then given by:

$$\mathcal{L}_{\mathrm{s}} = \frac{1}{|\tilde{X}_{\mathrm{L}}|} \sum\nolimits_{\boldsymbol{x}_i \in \tilde{X}_{\mathrm{L}}} \omega_i \mathcal{H}(y_i, \boldsymbol{z}_i), \tag{10}$$

where $\mathcal{H}(\cdot,\cdot)$ denotes the cross-entropy loss, and $|\cdot|$ corresponds to the size of a set. With the introduction of weight ω_i, the less confident examples, which are more likely to be low-quality, would receive less attention in the training process.

For unlabeled examples, conventional SSL methods [53], [54] typically adopt a combination of consistency regularization and confidence thresholding, where high-confidence predictions on weakly augmented examples are used as supervisory signals for their strongly augmented counterparts. Formally, the unsupervised loss $\mathcal{L}_{\mathrm{u}}$ for unlabeled examples is given by:

$$\mathcal{L}_{\mathrm{u}} = \frac{1}{|X_{\mathrm{U}}|} \sum\nolimits_{\boldsymbol{x}_i \in X_{\mathrm{U}}} \mathbb{1}\big(\max(\boldsymbol{z}_i) > \tau\big) \mathcal{H}(\hat{y}_i, \boldsymbol{Z}_i), \tag{11}$$

where $\hat{y}_i = \arg\max(\boldsymbol{z}_i)$ is the predicted label of the classifier, $\mathbb{1}(\cdot > \tau)$ is the indicator function for confidence thresholding with τ being the threshold. Here, $\boldsymbol{Z}_i = \mathrm{softmax}(f(\mathcal{A}^s(\boldsymbol{x}_i)))$, where $\mathcal{A}^s(\cdot)$ denotes the strong augmentation (*e.g.*, RandAugment [55]).

However, previous work [56] shows that this approach may bias the classifier toward easy-to-learn classes even when the training data are balanced. This bias would lead to the suboptimal decision boundary and degraded overall performance. In our DGPU framework, this problem can be further exacerbated because the classifier is iteratively refined, with bias introduced in earlier rounds persisting into later ones and accumulating its negative effects over iterations. To address this issue, we adopt Debiased Pseudo Labeling (DPL) [57], which adjusts the predicted logits by explicitly incorporating class prior information and correcting the bias introduced by confidence thresholding. Formally, given the original logits $f(\mathcal{A}^w(\boldsymbol{x}_i))$, the debiased logits $\tilde{f}_i$, which are then used to perform threshold filtering, can be computed as:

$$\tilde{f}_i = f(\mathcal{A}^w(\boldsymbol{x}_i)) - \lambda \log \boldsymbol{\eta}, \tag{12}$$

where λ is a tuning parameter that controls debiasing strength, and $\boldsymbol{\eta}$ is a moving-average estimation of the model's predicted distribution. Here we update the estimation of $\boldsymbol{\eta}$ in a moving-average manner:

$$\boldsymbol{\eta} = m\boldsymbol{\eta} + (1-m) \sum_{i=1}^{b} \boldsymbol{z}_i, \tag{13}$$

where $m \in [0, 1)$ is a momentum coefficient, and b corresponds to the batch size.

Moreover, we also incorporate a debiased marginal loss $\mathcal{L}_{\mathrm{DML}}$ to encourage a larger margin between positive and negative class. $\mathcal{L}_{\mathrm{DML}}$ is formulated as:

$$\mathcal{L}_{\mathrm{DML}}(\hat{y}_i, \boldsymbol{Z}_i) = -\log \frac{e^{(Z_i^{\hat{y}_i} + \lambda \log \eta_{\hat{y}_i})}}{\sum_{k=0}^{1} e^{(Z_i^k + \lambda \log \eta_k)}}, \tag{14}$$

where Z_i^k is the k-th entry of $\boldsymbol{Z}_i$, and η_k is the k-th entry of $\boldsymbol{\eta}$. We use $\mathcal{L}_{\mathrm{DML}}(\hat{y}_i, \boldsymbol{Z}_i)$ to replace $\mathcal{H}(\hat{y}_i, \boldsymbol{Z}_i)$ in Eq. (11), and use the debiased logits $\tilde{f}_i$ for threshold filtering, the debiased unsupervised loss can be formulated as:

$$\tilde{\mathcal{L}}_{\mathrm{u}} = \frac{1}{|X_{\mathrm{U}}|} \sum\nolimits_{\boldsymbol{x}_i \in X_{\mathrm{U}}} \mathbb{1}\big(\max(\tilde{\boldsymbol{z}}_i) > \tau\big) \mathcal{L}_{\mathrm{DML}}(\hat{y}_i, \boldsymbol{Z}_i), \tag{15}$$

where $\tilde{\boldsymbol{z}}_i = \mathrm{softmax}(\tilde{f}_i)$. The objective function for discriminative annotation stage $\mathcal{L}_{\mathrm{dis}}$ is then given by:

$$\mathcal{L}_{\mathrm{dis}} = \mathcal{L}_{\mathrm{s}} + \tilde{\mathcal{L}}_{\mathrm{u}}. \tag{16}$$

1) Pseudo-Labeled Set Construction: As the final step of each discriminative annotation round, the updated classifier is used to refine the pseudo-labeled set X_{L}, which provides supervision signals for guiding the finetuning of the conditional diffusion model in the next iteration. For each $\boldsymbol{x}_i \in X_{\mathrm{U}}$, we use o_i to indicate whether it is pseudo-labeled (*i.e.*, $o_i = 1$) or not (*i.e.*, $o_i = 0$). The probability of pseudo-labeling the example $\boldsymbol{x}_i \in X_{\mathrm{U}}$ is proportional to the confidence level of classifier on $\boldsymbol{x}_i$, which is expressed as:

$$p(o_i = 1) \propto \max(\boldsymbol{z}_i). \tag{17}$$

Therefore, the examples with higher confidence levels, which are more reliably classified, will have a greater probability to be pseudo-labeled. This selection strategy enhances the reliability of the pseudo-labeled set by prioritizing high-confidence examples. Moreover, compared with the strategy that setting a fixed threshold for selecting the pseudo-labeled examples, our mechanism ensures that every example has a chance to participate in the training process of the diffusion model, thereby mitigating the risk of generating a biased data distribution and helping prevent mode collapse. The pseudo-labeled set X_{L} is then given as:

$$X_{\mathrm{L}} = \cup_{\hat{y}_i \in \{0,1\}} \{(\boldsymbol{x}_i, \hat{y}_i) | \boldsymbol{x}_i \in X_{\mathrm{U}}, o_i = 1\}_{i=1}^{N}, \tag{18}$$

where $\hat{y}_i = \arg\max(\boldsymbol{z}_i)$ denotes the pseudo-label rendered by the classifier, and N represents the number of pseudo-labeled examples for each class. This refined pseudo-labeled set is subsequently used to guide the finetuning of conditional diffusion model in the next round of data generation.

TABLE I
SUMMARY OF EMPLOYED DATASETS

Dataset	Input Image Size	n_{P}	n_{U}	# Test Data	π_{P}	Positive Class	Backbone
F-MNIST	28×28	500	60,000	10,000	0.4	Top (*i.e.*, Class ID: 0, 2, 4, 6)	6-layer MLP
CIFAR-10	$3 \times 32 \times 32$	1,000	50,000	10,000	0.4	Vehicles (*i.e.*, Class ID: 0, 1, 8, 9)	13-layer CNN
CelebA	$3 \times 64 \times 64$	2,000	40,000	1,000	0.5	Male Faces	ResNet-18 [58]

Algorithm 1 Pseudo-Code of DGPU

Input: Training set X_{PU}, positive class prior π_{P}, classifier f parameterized by θ_f, conditional diffusion model $\boldsymbol{\epsilon}_\theta$, trade-off parameter λ.
Output: The optimal parameters θ_f^* for classifier f.
`// Algorithm initialization`
Initialize the classifier f with Eq. (3).
Construct the pseudo-labeled set X_{L} with Eq. (17) and Eq. (18).
for $iter = 1, 2, \ldots,$ **do**
  `// Data generation stage`
  Train the conditional diffusion model $\boldsymbol{\epsilon}_\theta$ with Eq. (8).
  Construct the generated set X_{G} with Eq. (9).
  `// Discriminative annotation stage`
  Refine the classifier f with Eq. (16).
  Update the pseudo-labeled set X_{L} with Eq. (17) and Eq. (18).
end

D. Interaction Between Data Generation and Discriminative Annotation

The two key components of our framework (*i.e.*, data generation and discriminative annotation) work in a collaborative fashion. In the data generation stage, the goal is to create high-quality synthetic images. However, generating precise and realistic images relies on having an accurate pseudo-labeled dataset. This is where the discriminative annotation stage plays a crucial role: it continuously refines the pseudo-labels to improve their accuracy, ensuring that the dataset used for training the diffusion model is as accurate as possible. With these improved labels, the data generation stage can produce more precise and high-fidelity images. These improved synthetic images, in turn, provide richer and more informative supervision for the discriminative annotation stage, enabling it to further refine its annotations and improve the performance of classifier. These two components benefit from each other in an iterative manner: better annotations lead to better synthetic images, which in turn further enhance the annotation quality. Over successive iterations, both stages improve, leading to a more accurate and robust overall framework. The complete pseudo-code of our algorithm is shown in Algorithm 1.

IV. EXPERIMENT

To demonstrate the effectiveness of our proposed DGPU, in this section, we perform exhaustive experiments on several publicly available benchmark datasets and real-world datasets.

A. Experimental Settings

Datasets. We conduct experiments on three popular benchmark datasets including F-MNIST [59], CIFAR-10 [24], and CelebA [25] datasets. More details of the datasets are provided below as well as in Table I.

- **F-MNIST** dataset [59] is a collection of grayscale images of clothing items and accessories. In our experiments, the images of tops (*i.e.*, "t-shirt", "pullovers", "coats", "shirts") are regarded as positive class and the images of other classes (*i.e.*, "trousers", "dresses", "sandals", "sneakers", "bags", "ankle boots") are regarded as negative class.
- **CIFAR-10** dataset [24] consists of colored images in 10 different classes including "airplanes", "automobiles", "birds", etc. In our experiments, we take the images of vehicles (*i.e.*, "airplanes", "automobiles", "ships", "trucks") as the positive class and treat the images of animals (*i.e.*, "birds", "cats", "deer", "dogs", "frogs", "horses") as the negative class.
- **CelebA** dataset [25] contains over 200,000 colored celebrity images. Following GenPU [18], we build a PU dataset by treating the first 20,000 images of male faces as the positive class, and the first 20,000 images of female faces as the negative class. The last 1,000 faces are used as test set.

Following the previous work [8], [29], [31], we randomly select n_{P} examples from positive examples to construct the labeled positive set X_{P}.

1) Baseline Methods: We choose fifteen state-of-the-art PU learning algorithms for comparison, which are listed as follows:

- **uPU** [7] designs an unbiased risk estimator for PU learning.
- **nnPU** [8] overcomes the overfitting problem by forcing the risk estimator of negative class to be non-negative.
- **RP** [60] ranks the training data by confidence and selects the most confident examples as positive or negative.
- **PUSB** [32] proposes a threshold estimation algorithm to deal with the selection bias during the labeling process.
- **PUbN** [28] first pretrains a model with nnPU algorithm to classify some reliable positive data, negative data, and unlabeled data, and then minimizes a risk approximated by the above three partitions.
- **Self-PU** [2] employs several self-supervised techniques to extend the learning capability of the previous PU model.
- **aPU** [61] deals with the arbitrary positive shift between source and target distributions.
- **VPU** [62] introduces a variational principle to relax the requirement of class prior.
- **P^{3}Mix** [63] proposes a heuristic mixup technique to provide data augmentation and supervision correction for PU learning.
- **Dist-PU** [31] forces the distribution of predicted labels to align with that of ground-truth ones.

- **HolisticPU** [29] assigns pseudo-labels to unlabeled examples by identifying the unique predictive trend of each example.
- **LaGAM** [34] incorporates a hierarchical contrastive learning module to extract the underlying grouping semantics within PU data and produce compact representations.
- **GenPU** [18] employs a complex adversarial framework with two generators and three discriminators to learn positive and negative distributions, then uses the generated data to train a separate classifier.
- **CGenPU** [15] simplifies GenPU to a single-stage model, via using one conditional generator and a discriminator with an auxiliary classifier that serves as the final classifier.
- **VAE-PU** [17] is a generative method specifically designed for the instance-dependent PU setting. It uses a Variational Autoencoder (VAE) [14] combined with a GAN to address selection bias.

We additionally include a fully supervised comparator which assumes complete label observability. This model is implemented using the same network architecture, optimizer, and hyperparameters as DGPU, but is trained on the entirely labeled dataset containing all positive and negative examples with their true labels.

2) Implementation Details: In our method, the backbone networks on every dataset are listed in Table I. We follow the convention [8], [31] to assume π_P as known in our experiments. For a fair comparison, all baseline methods are trained using the same weak augmentation strategy, which includes random crop and horizontal flip. This ensures that the reported performance gains of DGPU originate from the algorithmic design rather than differences in data preprocessing. The coefficient λ for controlling the debiasing strength is set to 0.8. Following conventional SSL methods such as FixMatch [53] and DPL [56], we set the confidence threshold parameter τ as a fixed value of 0.95 throughout all experiments. The number of pseudo-labeled examples for each class N is set to 10% of the number of unlabeled data in each iteration. We use Adam [64] as optimizer, where the learning rate is set to 1×10^{-4}, and the batch size is set to 256 for all datasets. In data generation stage, we use the training configuration of original EDM [50] with a batch size of 512 and a learning rate of 1×10^{-3}. Following previous works [2], [34], [63], we report test accuracy as the evaluation metric. For each dataset, we train each algorithm 3 times and report the mean and the standard deviation of the test accuracy. The parametric configurations of compared methods are the same with their original paper.

B. Comparison With State-of-the-Art Methods

In this section, we evaluate the classification performance of the proposed method by comparing it with the baseline methods mentioned above and also the fully supervised counterpart. The performance of all baseline methods and our DGPU on the adopted three benchmark datasets is summarized in Table II. According to the results, we have the following observations:

- On all three datasets, our proposed DGPU outperforms all competitors significantly. In particular, we improve the best baseline method by 3.89% on CIFAR-10 dataset, and 2.56% on CelebA dataset. The comparison result further validates the effectiveness of our proposed method.
- The standard deviation of the results of DGPU across different trials is relatively small when compared with most of the other methods, indicating that our method is robust to different selections of initial labeled positive examples in X_P.
- Our DGPU significantly outperforms all existing generative PU baseline methods on all three datasets. This performance gain mainly stems from two advances in our design. First, DGPU employs a conditional diffusion model as a stronger generative backbone, which produces more diverse and high-fidelity data than previous GAN- or VAE-based PU frameworks, thereby providing more effective supervision for classifier training. Second, DGPU unifies the generative and discriminative paradigms within a single iterative framework, where the diffusion generator and PU classifier benefit from each other in an iterative manner.
- The performance of our DGPU across all three datasets is comparable to that of its fully supervised counterpart. This result indicates that DGPU is capable of generating high-quality, semantically accurate images and providing precise supervisory signals for classifier training. Notably, DGPU even surpasses the fully supervised comparator on F-MNIST and CIFAR-10 datasets. This phenomenon can be attributed to the strong data augmentation effect introduced by the conditional diffusion model. The conditional diffusion model in DGPU acts as a powerful data augmenter that expands the training distribution beyond the limited real examples. Through iterative generation, the DGPU framework provides a larger training set than the fully supervised comparator, exposing the classifier to richer examples and reducing overfitting to the limited real data. Similar observations have also been reported in recent studies [65], where diffusion-based

TABLE II

COMPARISON RESULTS OF TEST ACCURACY (MEAN±STD) ON F-MNIST, CIFAR-10, AND CELEBA DATASETS. THE HIGHEST SCORES AMONG PU LEARNING METHODS ARE HIGHLIGHTED IN **BOLD**. ✓ DENOTES THAT OUR DGPU IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH CONFIDENCE LEVEL 95%

Dataset	F-MNIST	CIFAR-10	CelebA
uPU [7]	93.88±0.62 ✓	88.41±0.41 ✓	91.80±0.56 ✓
nnPU [8]	94.21±0.51 ✓	88.91±0.43 ✓	92.20±0.67 ✓
RP [60]	92.75±0.99 ✓	88.74±0.15 ✓	90.53±1.11 ✓
PUbN [28]	94.88±0.31 ✓	89.83±0.31 ✓	92.93±0.33 ✓
Self-PU [2]	94.91±0.21 ✓	89.31±0.56 ✓	92.85±0.31 ✓
aPU [61]	94.55±0.32 ✓	89.09±0.44 ✓	92.53±0.55 ✓
VPU [62]	92.21±1.01 ✓	87.89±0.51 ✓	91.75±0.78 ✓
P^3Mix [63]	94.35±0.25 ✓	89.21±0.49 ✓	92.73±0.41 ✓
Dist-PU [31]	95.40±0.31 ✓	91.88±0.32 ✓	93.73±0.25 ✓
HolisticPU [29]	95.54±0.24 ✓	91.95±0.42 ✓	93.65±0.57 ✓
LaGAM [34]	95.61±0.29 ✓	93.56±0.29 ✓	94.77±0.44 ✓
GenPU [18]	88.57±0.51 ✓	85.97±0.41 ✓	91.77±0.31 ✓
VAE-PU [17]	93.91±0.39 ✓	89.25±0.56 ✓	92.28±0.49 ✓
CGenPU [15]	91.04±0.31 ✓	89.56±0.33 ✓	92.77±0.29 ✓
DGPU (ours)	**97.44±0.15**	**97.45±0.22**	**97.33±0.18**
Fully Supervised	97.21±0.08	96.55±0.11	97.49±0.12

TABLE III
ABLATION STUDY OF DGPU ON F-MNIST, CIFAR-10, AND CELEBA DATASETS, WITH ✓ INDICATING THE ENABLING OF THE CORRESPONDING MODULE

Variant	Diffusion Model	Weighted Supervised Loss	Debiased Unsupervised Loss	F-MNIST	CIFAR-10	CelebA
Dist-PU				95.40±0.31	91.88±0.32	93.73±0.25
I	✓			96.75±0.28	95.93±0.33	96.15±0.21
II	✓		✓	97.01±0.24	96.79±0.22	96.93±0.19
III	✓	✓		96.88±0.17	96.56±0.25	96.64±0.22
DGPU	✓	✓	✓	**97.44±0.15**	**97.45±0.22**	**97.33±0.18**

synthetic data improve or even surpass fully supervised baseline methods in classification tasks.

The main reason for the above merits is that our algorithm can generate high-quality and diverse images, which accounts for the performance stability and superiority of the induced PU classifier when compared with other popular PU learning methodologies.

C. Experiments Under Instance-Dependent Setting

To evaluate the robustness and generalization of DGPU under more realistic conditions, we conduct experiments under the instance-dependent PU setting, where the labeling probability depends on instance features rather than being randomly assigned. Following PUSB [32], we first train a model by logistic regression on the data with ground-truth positive and negative labels. Then based on the predicted class probabilities of examples output by logistic regression, we select a subset of the positive training data as labeled positive set, and then combine the remaining positive data with negative data to compose the unlabeled set. Specifically, in every dataset, the positive examples are sampled according to the following strategy:

$$p(o = +1|\boldsymbol{x}, y = +1) \propto (p(y = +1|\boldsymbol{x}))^{10}. \tag{19}$$

In this way, the positive data that are far from the potential ideal decision boundary are more likely to be labeled. This sampling strategy models the situations that the human annotators prefer to label the positive examples that they are almost sure.

For comparison, we include PUSB [32] and VAE-PU [17], two representative methods designed for instance-dependent PU setting. As illustrated in Table V, our proposed DGPU achieves substantial improvements over methods explicitly designed to address selection bias under the challenging instance-dependent setting. This demonstrates that our collaborative framework is not only superior in the standard random-selection setting but also effective at handling the more realistic instance-dependent scenarios.

D. Ablation Studies

In order to analyze the impact of each module in DGPU, we conduct ablation studies on F-MNIST, CIFAR-10, and CelebA datasets, and investigate the model performance. The results are shown in Table III.

TABLE IV
IMAGE GENERATION RESULTS OF FID ON CIFAR-10 AND CELEBA DATASETS. † INDICATES THAT EDM IS TRAINED IN A FULLY SUPERVISED MANNER

Dataset	Method	FID-50K ↓
CIFAR-10	EDM [50]†	1.82
	DGPU (ours)	1.84
CelebA	EDM [50]†	1.87
	DGPU (ours)	1.91

TABLE V
COMPARISON RESULTS OF TEST ACCURACY (MEAN±STD) ON F-MNIST, CIFAR-10, AND CELEBA DATASETS UNDER THE INSTANCE-DEPENDENT PU SETTING. THE HIGHEST SCORES AMONG PU LEARNING METHODS ARE HIGHLIGHTED IN **BOLD**. ✓ DENOTES THAT OUR DGPU IS SIGNIFICANTLY BETTER THAN THE CORRESPONDING METHODS REVEALED BY THE PAIRED T-TEST WITH CONFIDENCE LEVEL 95%

Dataset	F-MNIST	CIFAR-10	CelebA
PUSB [32]	92.72±0.41 ✓	86.68±0.53 ✓	91.12±0.51 ✓
VAE-PU [17]	93.11±0.81 ✓	88.11±0.79 ✓	91.73±0.68 ✓
DGPU (ours)	**96.92±0.24**	**96.74±0.29**	**97.02±0.32**

1) Effect of Diffusion Model: To assess the impact of diffusion model in our DGPU, we compare the baseline Dist-PU (which does not utilize any generated data) with Variant I, where only the diffusion model is enabled. Across all datasets, we observe substantial performance improvements when the diffusion model is introduced. The consistent gains across datasets indicate that the diffusion model does not merely inflate the data volume, but contributes informative examples that enhance the decision boundary learned by the classifier. This validates the effectiveness of diffusion models in PU settings.

2) Effect of Weighted Supervised Loss: To explore the effect of the weighted supervised loss, we replace it with the standard cross-entropy loss during the discriminative annotation stage (Variant II). Without the confidence-based weighting, low-quality synthetic images contribute equally during training, which can mislead the classifier and degrade performance.

3) Effect of Debiased Unsupervised Loss: To evaluate the effect of the debiased unsupervised loss, we replace it with vanilla threshold filtering and consistency regularization. As shown in Table III, removing the debiasing mechanism leads to noticeable performance degradation, highlighting the importance of bias mitigation in discriminative annotation process. Without this debiasing mechanism, the classifier tends to become increasingly biased toward the majority class (either positive or negative). This issue is further exacerbated under

TABLE VI
COMPARISON RESULTS OF DIFFERENT SELECTION OF GENERATIVE MODELS

Dataset	Method	Model	Test Accuracy ↑	FID-50K ↓
CIFAR-10	StyleGAN2-ADA [66]	GAN	96.06 ± 0.35	2.94
	DGPU (ours)	Diffusion	97.45 ± 0.22	1.84
CelebA	StyleGAN2-ADA [66]	GAN	95.47 ± 0.29	3.22
	DGPU (ours)	Diffusion	97.33 ± 0.18	1.91

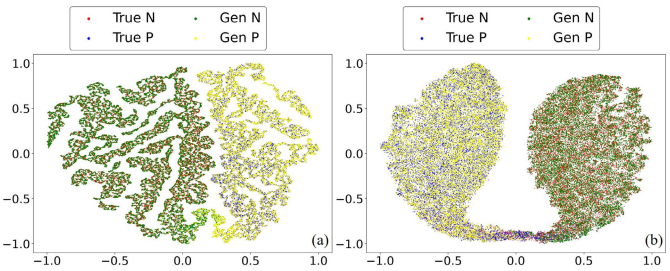


Fig. 3. t-SNE [67] visualization results of real images and generated images on CIFAR-10 and CelebA datasets. The subfigure (a) and (b) correspond to CIFAR-10 and CelebA datasets, respectively.

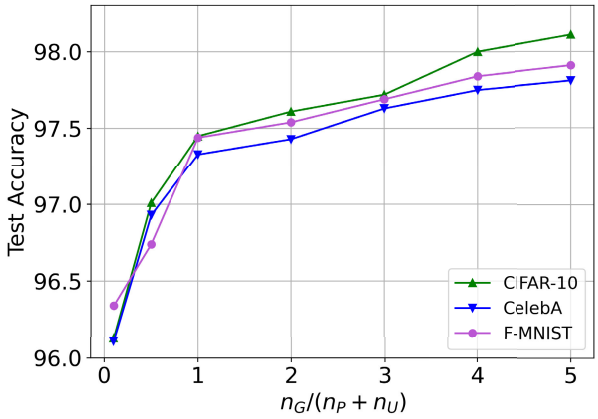


Fig. 4. Sensitivity to the amount of generated images on F-MNIST, CIFAR-10, and CelebA datasets.

our iterative refinement framework, where the pseudo-labeled set is continuously updated based on the current classifier's predictions. As the bias accumulates over iterations, the pseudo-labeled set becomes increasingly skewed, ultimately impairing the training of the diffusion model by providing imbalanced and misleading supervision.

E. Image Generation Results

To demonstrate the effectiveness of our DGPU in generating high-quality images, we give an analysis on the generated images. Qualitatively, we present the generated images of our DGPU on F-MNIST, CIFAR-10, and CelebA datasets in Figure 2. As illustrated in Figure 2, our DGPU can generate high-quality, diverse, and semantically correct images under the setting of PU learning, which is the key to the superior performance of our method. Quantitatively, we employ the Fréchet Inception Distance (FID) [68] to evaluate the quality of generated images. Table IV demonstrates that our DGPU achieves competitive performance using only PU data with EDM [50], which relies on full labels. Moreover, to further demonstrate that the generated images of our DGPU can effectively represent the distribution of real images, we present the t-SNE [67] visualization results of real images and generated images on CIFAR-10 and CelebA datasets. As illustrated in Figure 3, there is a noteworthy alignment between the distribution of generated and real images. This alignment strongly suggests that our DGPU can provide precise supervisory information for classifier training.

F. Effect of the Choice of Generative Models

Our DGPU focuses exclusively on diffusion models for data generation. However, a natural question arises: whether alternative generative models, such as the Generative Adversarial Networks [12] (GANs), could be potentially applicable. Therefore, to elucidate why diffusion models are particularly suitable for PU learning, we substitute the conditional diffusion model in our DGPU with StyleGAN2-ADA [66], and compare it with our original DGPU framework. Table VI demonstrates that our DGPU significantly outperforms the GAN-based method in terms of both classification and generation performance. This superior performance can be attributed to the capability of diffusion models to generate images of higher quality and diversity compared to GANs.

G. Effect of the Amount of Generated Images

In this section, we consider how the performance of classifier depends on the amount of generated data that is used to augment the labeled set. The results are shown in Figure 4, where n_{G} is the number of generated images. We observe that as n_{G} increases, the classification performance improves monotonically. To be specific, the test accuracy rises sharply when $\frac{n_{\mathrm{G}}}{n_{\mathrm{P}}+n_{\mathrm{U}}} < 1$, but the improvement becomes marginal when $\frac{n_{\mathrm{G}}}{n_{\mathrm{P}}+n_{\mathrm{U}}} > 1$, suggesting diminishing returns from excessive data augmentation. Interestingly, even at a low ratio of $\frac{n_{\mathrm{G}}}{n_{\mathrm{P}}+n_{\mathrm{U}}} = 0.1$, the model achieves surprisingly strong performance. This indicates that the generated images are of high quality and can significantly enhance learning performance even in small quantities. Consequently, we set $\frac{n_{\mathrm{G}}}{n_{\mathrm{P}}+n_{\mathrm{U}}} = 1$ in our experiments to achieve a balance between performance and computational efficiency.

H. Effect of the Selection of Pseudo-Labeled Examples

In this section, we consider how the performance of our DGPU is influenced by the selection method for pseudo-labeled examples. Specifically, we empirically evaluate two alternative strategies for constructing the pseudo-labeled set: 1) **No Filtration**: all examples in the unlabeled set X_{U} are assigned pseudo-labels. 2) **Fixed Threshold**: only examples satisfying the confidence criterion $\max(z_i) \geq \delta(\delta = 0.99)$ are

TABLE VII

COMPARISON RESULTS OF TRAINING TIME, TEST ACCURACY, AND FID ON CIFAR-10 AND CELEBA DATASETS

Dataset	Method	V100-hours	Test Accuracy ↑	FID-50K ↓
CIFAR-10	GenPU [18]	301	85.97 ± 0.41	25.94
	CGenPU [15]	335	89.56 ± 0.33	21.35
	LaGAM [34]	15	93.56 ± 0.29	–
	DGPU (ours)	388	97.45 ± 0.22	1.84
CelebA	GenPU [18]	1087	91.77 ± 0.31	23.49
	CGenPU [15]	1152	92.77 ± 0.29	19.81
	LaGAM [34]	24	94.77 ± 0.44	–
	DGPU (ours)	1352	97.33 ± 0.18	1.91

TABLE VIII

TEST ACCURACY OF DIFFERENT PSEUDO-LABELED SET CONSTRUCTION METHODS ON CIFAR-10 AND CELEBA DATASETS

Dataset	Method	Test Accuracy ↑
CIFAR-10	No Filtration	96.56 ± 0.32
	Fixed Threshold	96.87 ± 0.24
	DGPU (ours)	97.45 ± 0.22
CelebA	No Filtration	96.32 ± 0.24
	Fixed Threshold	96.59 ± 0.21
	DGPU (ours)	97.33 ± 0.18

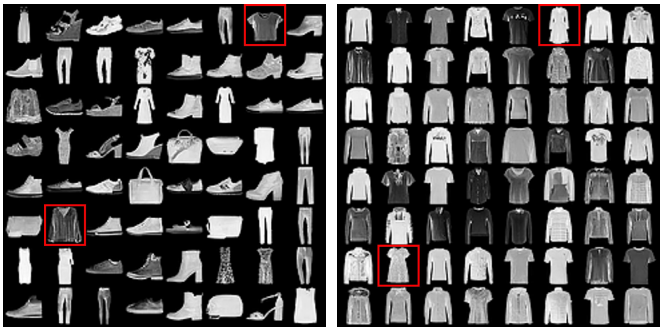
selected for pseudo-labeling. As shown in Table VIII, both strategies result in lower performance compared to our original design. The reason for this performance gap is that, in the no filtration variant, the model fails to filter out unreliable examples, thereby introducing noise into the training process of the diffusion model. In the fixed threshold variant, the diffusion model tends to overfit a subset of the training data, leading to the generation of a biased data distribution.

I. Computational Efficiency Analysis

In this section, we provide a comparative analysis of our DGPU against representative PU learning baseline methods, with a particular focus on computational efficiency, classification accuracy, and generative capability. As shown in Table VII, although DGPU requires more training time than GenPU and CGenPU, it achieves substantially higher classification accuracy and significantly better generation quality (e.g., an FID of 1.84 on CIFAR-10 and 1.91 on CelebA). Given the huge performance gap, a trade-off between performance and computational cost is justified. For completeness, we also include the results of LaGAM, which is a strong discriminative PU method that does not involve any generative component. Since LaGAM is a purely discriminative method without any generative component, its training time is substantially lower than that of DGPU. However, DGPU achieves much better performance on both datasets. This further demonstrates the advantage of incorporating high-quality generated data into the PU learning process.

J. Why Diffusion Models Can Benefit Classification?

In this section, we investigate the underlying mechanisms through which diffusion models enhance classification performance. Our empirical analysis reveals a key phenomenon:



(a) Generated negative images (b) Generated positive images

Fig. 5. A visualization of generated images on F-MNIST [59] under label noise conditions. The training set was intentionally corrupted with 10% symmetric label noise. Red bounding boxes indicate generated examples deviating from the expected class distribution (tops as positive class).

diffusion models exhibit remarkable tolerance to label noise while preserving sample fidelity. As evidenced in Figure 5, the generated images demonstrate a notably lower out-of-class-distribution rate compared to the noise rate present in the training labels. This observation is particularly significant in our DGPU setup, where the diffusion model is trained using pseudo-labels assigned by the classifier. Despite the inherent noise in these pseudo-labels, the diffusion model is able to generate images that are more semantically aligned with the intended classes. This noise-tolerant behavior implies that the generated data serve as a more reliable source of supervision compared to the noisy pseudo-labels themselves. As a result, incorporating these high-quality generated images into the training process leads to improved classifier performance.

V. CONCLUSION

This paper proposed a new PU learning algorithm termed "Discriminative-Generative Positive and Unlabeled Learning" (DGPU) that integrates discriminative model and generative model into a unified framework for conducting PU learning. In DGPU, a generative model (*i.e.*, diffusion model), is critically introduced to enrich supervisory information for training a discriminative model. Moreover, we utilize discriminative approach to generate pseudo-labels for better training the conditional diffusion model, which in turn produces high-precision and high-fidelity images to enhance the performance of the classifier. Consequently, the discriminative model and diffusion model can benefit from each other via an iterative

manner. As a result, our method has shown superior performance to various state-of-the-art PU learning methods on diverse datasets. In the future, we believe that the cooperation of discriminative model and generative model can play an important role in solving other weakly-supervised learning tasks such as partial label learning and label noise learning.

REFERENCES

[1] M. C. Du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1386–1394.

[2] X. Chen et al., "Self-PU: Self boosted and calibrated positive-unlabeled training," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, 2020, pp. 1510–1519.

[3] J. Zhou, B. Gao, K. Wang, J. Pei, P.-A. Heng, and J. Qin, "Landmark-free preoperative-to-intraoperative registration in laparoscopic liver resection," *IEEE Trans. Med. Imag.*, vol. 44, no. 11, pp. 4350–4362, Nov. 2025.

[4] W. Li, Q. Guo, and C. Elkan, "A positive and unlabeled learning algorithm for one-class classification of remote-sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 717–725, Feb. 2011.

[5] J. Zhang, Z. Wang, J. Meng, Y.-P. Tan, and J. Yuan, "Boosting positive and unlabeled learning for anomaly detection with multi-features," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1332–1344, May 2019.

[6] X. Li and B. Liu, "Learning from positive and unlabeled examples with different data distributions," in *Proc. Eur. Conf. Mach. Learn.*, 2005, pp. 218–229.

[7] M. C. D. Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 703–711.

[8] R. Kiryo, G. Niu, M. C. D. Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 117, 2017, pp. 63–70.

[9] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *Proc. Int. Conf. Mach. Learn.*, 2002, pp. 387–394.

[10] H. Yu, J. Han, and K. C. Chang, "Pebl: Web page classification without negative examples," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 70–81, Jan. 2004.

[11] G. Su, W. Chen, and M. Xu, "Positive-unlabeled learning from imbalanced data," in *Proc. Thirtieth Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 2995–3001.

[12] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[13] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[15] A. Papic, I. Kononenko, and Z. Bosnic, "Conditional generative positive and unlabeled learning," *Expert Syst. Appl.*, vol. 224, Aug. 2023, Art. no. 120046.

[16] W. Hu et al., "Predictive adversarial learning from positive and unlabeled data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 7806–7814.

[17] B. Na, H. Kim, K. Song, W. Joo, Y.-Y. Kim, and I.-C. Moon, "Deep generative positive-unlabeled learning under selection bias," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1155–1164.

[18] M. Hou, B. Chaib-Draa, C. Li, and Q. Zhao, "Generative adversarial positive-unlabelled learning," in *Proc. Twenty-Seventh Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2255–2261.

[19] R.-C. Tu et al., "Global and local semantic completion learning for vision-language pre-training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 12, pp. 11065–11079, Dec. 2025.

[20] R.-C. Tu, X.-L. Mao, J.-Y. Liu, Z.-A. Ma, T. Lan, and H. Huang, "Prospective layout-guided multi-modal online hashing," *IEEE Trans. Image Process.*, vol. 34, pp. 5935–5947, 2025.

[21] J. Ho, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2024, pp. 6840–6851.

[22] J. Ma, T. Hu, W. Wang, and J. Sun, "Elucidating the design space of classifier-guided diffusion generation," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–16.

[23] C.-Y. Chan, W.-C. Siu, Y.-H. Chan, and H. Anthony Chan, "AnlightenDiff: Anchoring diffusion probabilistic model on low light image enhancement," *IEEE Trans. Image Process.*, vol. 33, pp. 6324–6339, 2024.

[24] A. Krizhevsky, "*Learning multiple layers of features from tiny images*," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009-10, 2009.

[25] 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, doi: 10.1109/ICCV.2015.425.

[26] F. Denis, "PAC learning from positive statistical queries," in *Proc. Algorithmic Learn. Theory*, 1998, pp. 112–126.

[27] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Mach. Learn.*, vol. 109, no. 4, pp. 719–760, Apr. 2020.

[28] Y.-G. Hsieh, G. Niu, and M. Sugiyama, "Classification from positive, unlabeled and biased negative data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2820–2829.

[29] S. Chen, C. Geng, S.-Y. Li, W. Wan, and W. Xinrui, "Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 67589–67602.

[30] C. Gong, H. Shi, T. Liu, C. Zhang, J. Yang, and D. Tao, "Loss decomposition and centroid estimation for positive and unlabeled learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 918–932, Mar. 2021.

[31] Y. Zhao, Q. Xu, Y. Jiang, P. Wen, and Q. Huang, "Dist-PU: Positive-unlabeled learning from a label distribution perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14441–14450.

[32] M. Kato, T. Teshima, and J. Honda, "Learning from positive and unlabeled data with a selection bias," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[33] C. Gong et al., "Instance-dependent positive and unlabeled learning with labeling bias estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4163–4177, Aug. 2022.

[34] L. Long et al., "Positive-unlabeled learning by latent group-aware meta disambiguation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23138–23147.

[35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.

[36] X. Huang et al., "HumanNorm: Learning normal diffusion model for high-quality and realistic 3D human generation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4568–4577.

[37] C. Xu, J. Yan, M. Yang, and C. Deng, "Rethinking noise sampling in class-imbalanced diffusion models," *IEEE Trans. Image Process.*, vol. 33, pp. 6298–6308, 2024.

[38] J. Yue, L. Fang, S. Xia, Y. Deng, and J. Ma, "Dif-fusion: Toward high color fidelity in infrared and visible image fusion with diffusion models," *IEEE Trans. Image Process.*, vol. 32, pp. 5705–5720, 2023.

[39] A. Jolicoeur-Martineau, C. Pal, and V. Voleti, "MCVD–masked conditional video diffusion for prediction, generation, and interpolation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23371–23385. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/944618542d80a63bbec16dfbd2bd689a-Abstract-Conference.html

[40] A. Blattmann et al., "Align your latents: High-resolution video synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22563–22575, doi: 10.1109/CVPR52729.2023.02161.

[41] S. Zhou, P. Yang, J. Wang, Y. Luo, and C. C. Loy, "Upscale-A-video: Temporal-consistent diffusion model for real-world video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 2535–2545.

[42] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–9.

[43] H. Liu et al., "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 21450–21474. [Online]. Available: https://proceedings.mlr.press/v202/liu23f.html

[44] J. Sohl-Dickstein, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 2256–2265.

[45] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.

[46] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–17.

[47] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8780–8794.

[48] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022, *arXiv:2207.12598*.
[49] A. Graikos et al., "Learned representation-guided diffusion models for large-image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 8532–8542.
[50] T. Aila, M. Aittala, T. Karras, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 26565–26577.
[51] H. Hu, F. Wei, H. Hu, Q. Ye, J. Cui, and L. Wang, "Semi-supervised semantic segmentation via adaptive equalization learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 22106–22118.
[52] H. Chen et al., "SoftMatch: Addressing the quantity-quality tradeoff in semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–16.
[53] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–10.
[54] Y. Wang et al., "FreeMatch: Self-adaptive thresholding for semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–20.
[55] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.
[56] X. Wang, Z. Wu, L. Lian, and S. X. Yu, "Debiased learning from naturally imbalanced pseudo-labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14627–14637.
[57] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–13.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[59] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[60] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," in *Proc. Conf. Uncertainty Artif. Intell.*, 2017, pp. 1–10.
[61] Z. Hammoudeh and D. Lowd, "Learning from positive and unlabeled data with arbitrary positive shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 13088–13099.
[62] H. Chen, F. Liu, Y. Wang, L. Zhao, and H. Wu, "A variational approach for learning from positive and unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14844–14854.
[63] C. Li, X. Li, L. Feng, and J. Ouyang, "Who is your right mixup partner in positive and unlabeled learning," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=NH29920YEmj
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[65] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves ImageNet classification," *Trans. Mach. Learn. Res.*, vol. 2023, pp. 1–16, Oct. 2023.
[66] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.
[67] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[68] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. NIPS*, 2017, pp. 6626–6637.

Botai Yuan received the bachelor's degree from Huazhong University of Science and Technology (HUST) in 2021. He is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University (SJTU) under the supervision of Prof. Jie Yang and Prof. Chen Gong. His research interests include positive and unlabeled learning.

Chen Gong (Senior Member, IEEE) received the dual Ph.D. degree from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the University of Technology Sydney (UTS). He is currently a Full Professor with Shanghai Jiao Tong University. He has published more than 130 technical papers at prominent journals and conferences, such as *Journal of Machine Learning Research*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *ACM Transactions on Intelligent Systems and Technology*, ICML, NeurIPS, ICLR, CVPR, ICCV, AAAI, IJCAI, and ICDM. His research interests mainly include machine learning, data mining, and learning-based vision problems. He won the "Excellent Doctorial Dissertation Award" of Chinese Association for Artificial Intelligence, "Young Elite Scientists Sponsorship Program" of China Association for Science and Technology, "Wu Wen-Jun AI Excellent Youth Scholar Award," and the Scientific Fund for Distinguished Young Scholars of Jiangsu Province. He was also selected as the "Global Top Chinese Young Scholars in AI" released by Baidu and "World's Top 2% Scientists" released by Stanford University. He serves as the Area Chair or a Senior PC Member for several top-tier conferences, such as AAAI, IJCAI, ICML, ICLR, ECML-PKDD, AISTATS, ICDM, and ACM MM. He serves as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Neural Networks*, and *Neural Processing Letters*.

Dacheng Tao (Fellow, IEEE) is currently a Distinguished University Professor with the College of Computing and Data Science, Nanyang Technological University, Singapore. His publications have been cited more than 142 K times and he has an H-index of more than 180 in Google Scholar. He mainly applies statistics and mathematics to artificial intelligence and data science, and his research is detailed in one monograph and more than 200 publications in prestigious journals and proceedings, leading conferences, with best paper awards, best student paper awards, and test-of-time awards. He is a fellow of Australian Academy of Science, AAAS, and ACM. He received the 2015 and 2020 Australian Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award.

Jie Yang (Senior Member, IEEE) received the bachelor's degree in automatic control and the master's degree in pattern recognition and intelligent systems from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the Ph.D. degree from the Department of Computer Science, University of Hamburg, Hamburg, Germany, in 1994. He is currently a Professor and the Director of the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. He is a Principal Investigator of more than 30 national and ministry scientific research projects in image processing, pattern recognition, data mining, and artificial intelligence. He has published six books, and more than 500 articles in national or international academic journals and conferences. Google citation 29 300, H-index of 88. Up to now, he has awarded six research achievement prizes from the Ministry of Education, China, and Shanghai Municipality. He has owned 48 patents. Three Ph.D. dissertations, he supervised were evaluated as "National Best Ph.D. Dissertation" in 2009, 2017, and 2019. He has been the Chairperson and a keynote speaker at more than ten international conferences. He is selected in the list of 2025 World Top 2% Career-long Impact Scientists issued by Stanford University and Elsevier.