Data Augmentation with Diffusion for Open-Set Semi-Supervised Learning

Seonghyun Ban[1*], **Heesan Kong**[1*], **Kee-Eung Kim**[1, 2†]
[1]Kim Jaechul Graduate School of AI, KAIST
[2]School of Computing, KAIST
shban@ai.kaist.ac.kr, hskong@ai.kaist.ac.kr, kekim@kaist.ac.kr

Abstract

Semi-supervised learning (SSL) seeks to utilize unlabeled data to overcome the limited amount of labeled data and improve model performance. However, many SSL methods typically struggle in real-world scenarios, particularly when there is a large number of irrelevant instances in the unlabeled data that do not belong to any class in the labeled data. Previous approaches often downweight instances from irrelevant classes to mitigate the negative impact of class distribution mismatch on model training. However, by discarding irrelevant instances, they may result in the loss of valuable information such as invariance, regularity, and diversity within the data. In this paper, we propose a data-centric generative augmentation approach that leverages a diffusion model to enrich labeled data using both labeled and unlabeled samples. A key challenge is extracting the diversity inherent in the unlabeled data while mitigating the generation of samples irrelevant to the labeled data. To tackle this issue, we combine diffusion model training with a discriminator that identifies and reduces the impact of irrelevant instances. We also demonstrate that such a trained diffusion model can even convert an irrelevant instance into a relevant one, yielding highly effective synthetic data for training. Through a comprehensive suite of experiments, we show that our data augmentation approach significantly enhances the performance of SSL methods, especially in the presence of class distribution mismatch.

1 Introduction

Deep neural networks (DNNs), trained using a large amount of labeled datasets, have shown to achieve remarkable performance in a variety of supervised learning tasks, such as image classification (LeCun et al., 2015; Krizhevsky et al., 2017) and object detection (Everingham et al., 2010; Lin et al., 2014). Nonetheless, the intensive labor of annotating vast datasets often renders the construction of sufficiently large labeled datasets prohibitively expensive for numerous applications (Oliver et al., 2018). To address this, semi-supervised learning (SSL) (Chapelle et al., 2009) has emerged as a viable approach, which aims to leverage abundant unlabeled data to overcome the limited availability of labeled data.

Recent progress in SSL has made many noteworthy advancements, including techniques such as pseudo-labeling (Lee, 2013; Pham et al., 2021), consistency regularization (Sajjadi et al., 2016; Tarvainen & Valpola, 2017; Sohn et al., 2020), and entropy minimization (Grandvalet & Bengio, 2004; Miyato et al., 2018; Berthelot et al., 2019). However, a common limitation of these methods is their reliance on the critical assumption that both unlabeled and labeled data instances are drawn

[*]Equal Contribution.
[†]Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

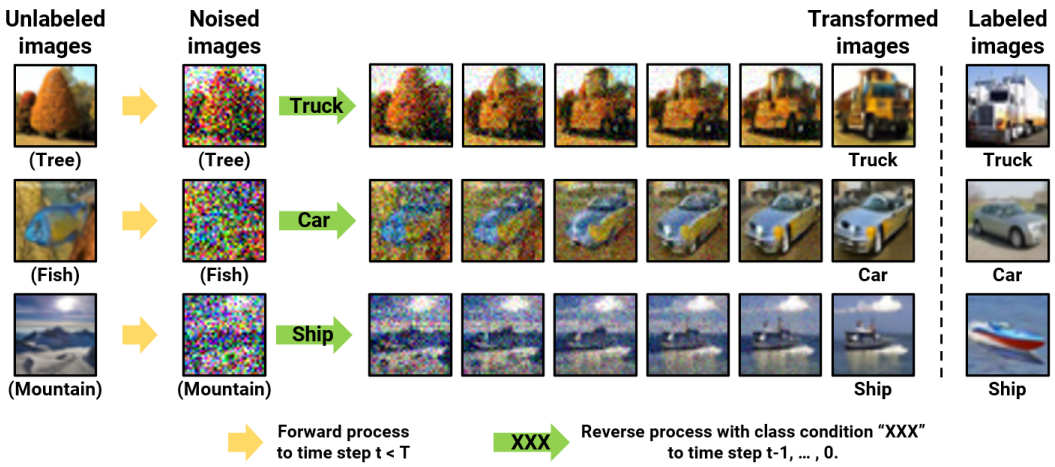

Figure 1: Transforming unlabeled data using a diffusion model. Initially, the unlabeled data includes classes like trees, fish, and mountains, which are irrelevant to the labeled data's classes such as trucks, cars, and ships. The reverse process with class conditioning resolves this mismatch while preserving the diversity of the original unlabeled samples. More examples can be found in Appendix H.

from an identical distribution. This often leads to significant performance degradation when there is a class distribution mismatch, as noted in Oliver et al. (2018). Given that real-world settings often deviate from this assumption (Guo et al., 2020), it becomes crucial to address the class distribution mismatch for successfully applying SSL in realistic scenarios.

A prevalent approach to mitigate the class distribution mismatch in SSL involves selectively utilizing unlabeled data through a weighting function which serves to reduce the influence of irrelevant unlabeled samples (Chen et al., 2020; Guo et al., 2020). While these filtering methods are intuitively appealing and have shown their effectiveness in mitigating negative effect of the class distribution mismatch, they may result in the loss of valuable information such as invariance, regularity, and diversity within the data: even though the labeled data contains only white trucks, could we generate images of red trucks using the red bike images in the unlabeled dataset?

In this paper, we propose a generative data augmentation approach that leverages a diffusion model to enrich labeled data using both labeled and unlabeled samples. A key challenge lies in utilizing the diversity of the unlabeled data to compensate the limited amount of labeled data, while minimizing the generation of samples that are irrelevant to the classes in the labeled dataset. To address this, we integrate diffusion model training with a discriminator that evaluates the relevance of each unlabeled instance. The discriminator's resulting score is used to assign weights to unlabeled instances, allowing those with higher relevance to contribute more significantly to the training of the diffusion model.

In addition, drawing inspiration from the approach in Meng et al. (2022), we add noise to each unlabeled sample and utilize them as guide images during the data generation process. As depicted in Figure 1, we found that incorporating class conditions into this generation process can transform possibly irrelevant unlabeled samples into labeled samples while preserving key characteristics of the original unlabeled samples (e.g., outline, color arrangement, shape, etc.).

Our extensive experimental results, utilizing CIFAR-10, CIFAR-100 (Krizhevsky & Hinton, 2009), ImageNet-30 (Deng et al., 2009), and ImageNet-100 (Cao et al., 2022) datasets with six baseline methods, demonstrate that our approach further improves the performance of recent SSL methods, especially under the class distribution mismatch.

2 Related Works

2.1 Standard Semi-Supervised Learning

Semi-Supervised Learning (SSL) aims to leverage both labeled and unlabeled data to mitigate issues related to the scarcity and high annotation cost of labeled data. Since many SSL methods operate under the assumption that labeled and unlabeled data are sampled from an identical distribution,

we refer to this as the standard SSL setting. Among the various approaches to the standard SSL setting, we briefly review two of the most representative methods: pseudo-labeling and consistency regularization.

Pseudo-labeling (Scudder, 1965; McLachlan, 1975; Lee, 2013) is a technique where the model-predicted labels of the unlabeled data are treated as if they were true labels. Essentially, this method simply converts unlabeled data into labeled data. On the other hand, consistency regularization (Bachman et al., 2014) has become a crucial component in recent SSL regimes. This approach applies data augmentation on the unlabeled data to regularize the model to yield similar outputs for augmented views of the same instance (Sajjadi et al., 2016; Laine & Aila, 2017; Sohn et al., 2020).

However, these methods often fail when there is a mismatch between the distributions of labeled and unlabeled data. In some cases, their performance may be even worse than simply discarding all the unlabeled data (Oliver et al., 2018). The primary source of the problem is the presence of unlabeled instances that do not belong to any of the classes present in the labeled data, which we refer to as out-of-distribution (OOD) instances. They may exacerbate confirmation bias (Arazo et al., 2020) in pseudo-labeling approaches or intensify the overconfidence problem in consistency regularization approaches (Chen et al., 2020).

2.2 Open-set Semi-Supervised Learning

The open-set SSL setting refers to the practical yet challenging scenario where the class distributions of labeled and unlabeled dataset differ significantly. A prevalent solution to the class distribution mismatch is to filter out OOD instances from the unlabeled dataset. To achieve this, it is necessary to accurately identify them, despite the absence of label information. Uncertainty-Aware Self-Distillation (UASD), proposed by Chen et al. (2020), formulates temporally ensembled networks and utilizes ensemble prediction to quantify predictive uncertainty of labels to identify OOD instances. DS3L (Guo et al., 2020) adopts a meta-learning approach to selectively use unlabeled data that enhances generalization performance. OpenMatch (Saito et al., 2021) leverages one-vs-all classifiers as the OOD detector to filter out OOD instances. Safe-Student (He et al., 2022) employs teacher-student mechanism and introduces energy-discrepancy, a new scoring function for detecting OOD instances. The above methods follow the detect-and-filter paradigm, which is the dominant approach of open-set SSL, assuming that OOD instances are fundamentally harmful.

In contrast to these detect-and-filter approaches, several studies share similar goals to ours, aiming to harness the potential of OOD unlabeled data rather than simply discarding them. T2T (Huang et al., 2021) incorporates a warm-up training step using OOD instances to perform a self-supervised pretext task for learning effective discriminative features. TOOR (Huang et al., 2022) introduces a weighting mechanism to evaluate the transferability of each OOD instance based on domain similarity and class tendency, and uses adversarial domain adaptation to align the feature distributions of transferable OOD instances and in-distribution (ID) instances. Fix-A-Step (Huang et al., 2023) leverages OOD instances to obtain useful data augmentation to promote diversity of training data, and integrate this idea into MixMatch (Berthelot et al., 2019). IOMatch (Li et al., 2023) takes into consideration that the OOD detector may be unreliable, particularly when labeled data are scarce. It instead employs a multi-binary classifier to produce unified open-set pseudo-labels for labeled and unlabeled data, including OOD instances.

Our approach significantly deviates from these open-set SSL methods. It is primarily centered on developing an effective data augmentation strategy from both labeled and unlabeled data, which can be used to transform an unlabeled OOD instance into a labeled ID instance.

2.3 Recent Diffusion-based Augmentation Approaches

In this section, we provide a detailed comparison of DWD with diffusion-based methods, namely DPT (You et al., 2023) and DA-Fusion[1] (Trabucco et al., 2024).

DPT is a simple yet effective method that integrates diffusion models into SSL. To generate semantically accurate images, they trained a semi-supervised classifier on partially labeled real images and used it to assign pseudo-labels for all the data. Subsequently, they trained a conditional diffusion

[1]Although DA-Fusion is not a SSL algorithm, we include it as a supervised learning baseline due to its methodological similarity in utilizing image-to-image generation process.

model using the pseudo-labels to synthesize images and then re-trained the semi-supervised classifier using these generated images. However, they still rely on the fundamental assumption behind the standard SSL setting. When faced with mismatch in class distributions, they suffer from the confirmation bias in pseudo labels of OOD unlabeled images and become ineffective. In contrast, our approach addresses the class distribution mismatch by integrating the discriminator into the training of the diffusion model.

DA-Fusion is a data augmentation method that utilizes a large pretrained text-to-image diffusion model (i.e., Stable Diffusion, Rombach et al., 2022). Similar to ours, it also initiates the reverse process with partially noised real images rather than generating images from scratch. However, the purpose of the generation process is distinctly different from that of our approach. While DA-Fusion aims to augment given labeled samples with subtle visual details already contained in the pretrained diffusion model, our method properly trains a diffusion model to capture both the labeled data distribution and the diversity of unlabeled samples from the given datasets, and transforms irrelevant unlabeled samples into labeled ones.

3 Preliminaries

Diffusion models (Sohl-Dickstein et al., 2015) incrementally add Gaussian noises to the data during its forward process and progressively removes this noise during the reverse process to reconstruct the original data. Given a data point $\mathbf{x}_0$, the forward process is defined as a Markov chain that produces a sequence of noisy samples $\mathbf{x}_1, ..., \mathbf{x}_T$ according to a variance schedule $\beta_1, ..., \beta_T$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \qquad q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{1}$$

The forward process exhibits a notable property in that $\mathbf{x}_t$ at any arbitrary time step $t \in \{1, ..., T\}$ can be obtained in closed form:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon} \tag{2}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As the reverse process can be expressed using the same functional form when β_t is sufficiently small (Feller, 1949; Sohl-Dickstein et al., 2015), the reverse process is also defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T, \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)), \qquad p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3}$$

Denoising Diffusion Probabilistic Models (DDPMs) Ho et al. (2020) propose to use a simplified Gaussian distribution parameterization of the reverse process, which sets $\boldsymbol{\Sigma}_\theta$ to time-dependent constants and reparameterizes the mean function approximator $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ with noise predictor $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ which predicts $\boldsymbol{\epsilon}$ in (2). The reverse process is then learned by the following objective:

$$L_{ddpm} = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2] \tag{4}$$

For the conditional DDPM, the primary modification is the incorporation of conditions $\mathbf{c}$ (such as classes or texts) as an additional input, expressed as $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}, t)$. This adaptation allows the diffusion model to take into account specific conditions or attributes during the reverse process, thereby enhancing its applicability to more targeted scenarios.

Positive-Unlabeled (PU) Learning (Liu et al., 2002; Li & Liu, 2003; Du Plessis et al., 2015; Kiryo et al., 2017) is a binary classification task in a situation where negative labels are missing. It aims to train models using positive-labeled and unlabeled data to perform binary classification. The main idea involves indirectly estimating the model loss on negative samples using the class prior. Given that unlabeled data are drawn from $p^u(\mathbf{x}) = \mu\, p^+(\mathbf{x}) + (1-\mu)\, p^-(\mathbf{x})$, where $\mu = p(Y=1)$ is the class prior and $p^+(\mathbf{x}) = p(\mathbf{x}\,|\,Y=1)$ and $p^-(\mathbf{x}) = p(\mathbf{x}\,|\,Y=-1)$ are positive and negative class-conditional densities, the loss can be reformulated as:

$$\begin{aligned}\mathbb{E}\left[\ell(g(\mathbf{x}), Y)\right] &= \mu\mathbb{E}_{p^+}[\ell(g(\mathbf{x}), 1)] + (1-\mu)\mathbb{E}_{p^-}[\ell(g(\mathbf{x}), -1)] \\ &= \mu\mathbb{E}_{p^+}[\ell(g(\mathbf{x}), 1)] - \mu\mathbb{E}_{p^+}[\ell(g(\mathbf{x}), -1)] + \mathbb{E}_{p^u}[\ell(g(\mathbf{x}), -1)]\end{aligned} \tag{5}$$

Here, ℓ denotes a loss function and g represents the model.

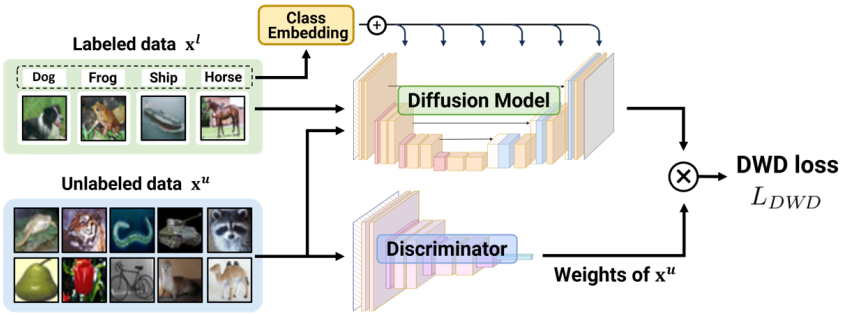

Figure 2: Schematic diagram of Discriminator-Weighted Diffusion (DWD). The conditional diffusion model is trained using both labeled and unlabeled data. The unlabeled data is utilized for unconditional training without class conditions. The pre-trained discriminator assigns weights to each unlabeled data sample to mitigate the potential negative impact of OOD samples.

4 Methodology

In this section, we introduce our data augmentation method, which leverages a diffusion model to generate synthetic data to address the scarcity of labeled data. As previously discussed, we start with training the diffusion model on labeled and unlabeled data while mitigating the class distribution mismatch. Subsequently, we employ the reverse diffusion process to transform unlabeled samples into synthesized labeled samples.

4.1 Training Diffusion Model

A simple training scheme for SSL data The diffusion model trained only on the labeled data will inevitably overfit and merely generate replications due to their limited amount. It is thus essential to train the diffusion model on both labeled and unlabeled data, while taking advantage of the label information. We adopt a class-conditional diffusion model, shown in Figure 2, where labeled data are used for conditional training while unlabeled data are used for unconditional training, trained with the loss

$$L_{\text{semi}} = \mathbb{E}_{(\mathbf{x}_0,\mathbf{c})\sim\mathcal{D}_l,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,\mathbf{c},t)-\boldsymbol{\epsilon}||_2^2] + \alpha\cdot\mathbb{E}_{\mathbf{x}_0\sim\mathcal{D}_u,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)-\boldsymbol{\epsilon}||_2^2], \tag{6}$$

where α serves as a hyper-parameter that controls balance between labeled and unlabeled data, $\mathcal{D}_l$ and $\mathcal{D}_u$ represent labeled and unlabeled datasets. Intuitively, this straightforward objective aims to train the diffusion model to reconstruct not only the limited labeled data but also the abundant unlabeled data, thereby regularizing the model against overfitting to the labeled data. It is noteworthy that this approach shares similarities with the consistency regularization technique in the sense that the unlabeled data are utilized for regularization purposes.

Discriminator However, when we train the diffusion model with the loss in (6), OOD unlabeled samples can have a negative impact on capturing important characteristics of the labeled data distribution. These samples should be made contribute less towards the overall training loss. In line with aforementioned filtering methods (Chen et al., 2020; Guo et al., 2020), we leverage a discriminator to weigh the unlabeled data instances. The discriminator is tasked with differentiating between positive samples that are closely aligned with the distribution of labeled data and negative samples that are irrelevant. To train the discriminator, we adopt PU learning with the training loss

$$L_{\text{disc}} = \mu\cdot\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_l}[-\log\mathbf{d}_\phi(\mathbf{x}) + \log\left(1-\mathbf{d}_\phi(\mathbf{x})\right)] + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_u}[-\log\left(1-\mathbf{d}_\phi(\mathbf{x})\right)], \tag{7}$$

where $\mathbf{d}_\phi$ denotes the discriminator parameterized by ϕ, and μ represents the ratio of positive samples among unlabeled samples, i.e. belonging to one of the classes in the labeled dataset. This ratio can either be estimated from the dataset (Menon et al., 2015; Jain et al., 2016; Christoffel et al., 2016) or treated as a hyperparameter.

We then use the discriminator to assign weights on the unlabeled instances so that they align with the distribution of labeled data. A simple algebraic manipulation tells us that the following weight formula yields an unbiased loss estimation via importance sampling:

Algorithm 1 Discriminator-Weighted Diffusion (DWD) - Training

Input Labeled dataset $\mathcal{D}_l$ and unlabeled dataset $\mathcal{D}_u$.
Parameter Learning rate η_θ, η_ϕ, hyper-parameter μ, α, and total number of iterations K_θ, K_ϕ.
Either pretrain the diffusion model $\boldsymbol{\epsilon}_\theta$ on $\mathcal{D}_u$ or acquire an off-the-shelf pretrained diffusion model.
# Train the discriminator $\mathbf{d}_\phi$ using objective (7):
for $n = 0, 1, 2, ..., K_\phi$ **do**
  Sample a batch of data $\mathbf{x}^l \sim \mathcal{D}_l$ and $\mathbf{x}^u \sim \mathcal{D}_u$.
  $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \left[\mu \cdot \mathbb{E}_{\mathbf{x}^l}[-\log \mathbf{d}_\phi(\mathbf{x}^l) + \log\left(1 - \mathbf{d}_\phi(\mathbf{x}^l)\right)] + \mathbb{E}_{\mathbf{x}^u}[-\log\left(1 - \mathbf{d}_\phi(\mathbf{x}^u)\right)]\right]$
end for
# Finetune the diffusion model $\boldsymbol{\epsilon}_\theta$ using the discriminator $\mathbf{d}_\phi$ and weighting function w from (8):
for $n = 0, 1, 2, ..., K_\theta$ **do**
  Sample a batch of data $(\mathbf{x}^l, \mathbf{c}) \sim \mathcal{D}_l$ and $\mathbf{x}^u \sim \mathcal{D}_u$.
  $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \left[\mathbb{E}_{\mathbf{x}^l,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^l, \mathbf{c}, t) - \boldsymbol{\epsilon}||_2^2] + \alpha \cdot \mathbb{E}_{\mathbf{x}^u,t,\boldsymbol{\epsilon}}[w(\mathbf{x}^u) \cdot ||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^u, t) - \boldsymbol{\epsilon}||_2^2]\right]$
end for
Output Learned diffusion model $\boldsymbol{\epsilon}_\theta$ and learned discriminator $\mathbf{d}_\phi$.

Proposition 4.1. *Given an optimal discriminator $\mathbf{d}^*$, using the following importance weight on unlabeled data yields an unbiased estimation of the loss function with respect to the labeled data distribution:*

$$w(\mathbf{x}) = \frac{\mathbf{d}(\mathbf{x})}{\mu\,\mathbf{d}(\mathbf{x}) + (1-\mu)(1-\mathbf{d}(\mathbf{x}))} \tag{8}$$

For a detailed proof, please refer to Appendix A. The final training loss for the diffusion model then becomes:

$$L_{\text{DWD}} = \mathbb{E}_{(\mathbf{x}_0,\mathbf{c})\sim\mathcal{D}_l,t,\boldsymbol{\epsilon}}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}, t) - \boldsymbol{\epsilon}||_2^2] + \alpha \cdot \mathbb{E}_{\mathbf{x}_0\sim\mathcal{D}_u,t,\boldsymbol{\epsilon}}[w(\mathbf{x}_0)||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2] \tag{9}$$

The overall training scheme, referred to as Discriminator-Weighted Diffusion (DWD), is outlined in Algorithm 1.

4.2 Seeding Data Generation with Unlabeled Instances

After training the diffusion model, a straightforward approach to generating synthetic data would be to start the reverse diffusion process from Gaussian noise. In our methodology, however, we begin the reverse process using a partially noised image of an unlabeled instance. As we will show in Section 5.3, we found that this approach leads to additional performance gains. It successfully transforms an OOD instance into an in-distribution sample while preserving some important characteristics in the original sample (see Figure 1). Thus we can exploit the diversity present in the unlabeled data when generating synthetic in-distribution data. The procedure is detailed in Algorithm 2 in the Appendix B.

We can think of two usage scenarios for the samples generated from the diffusion model: we could take the samples as pseudo-labeled synthetic data to enrich the labeled data and employ a fully supervised learning method, or take the samples as a transformed unlabeled data by discarding the class conditions and employ an SSL method. In the latter case, we expect that the SSL method will perform better, since the class distribution mismatch has been mitigated.

5 Experiments

To assess the effectiveness of DWD, we conduct experiments across a broad set of tasks in two settings. (1) **DWD-SL**: a fully supervised learning setting where the unlabeled dataset is converted to a pseudo-labeled dataset by replacing the instances with their transformations along with their class conditions as target labels; and (2) **DWD-UT**: an SSL setting where the unlabeled dataset is replaced by the transformed unlabeled samples and employ the baseline SSL method.

Tasks The SixAnimal task Oliver et al. (2018) uses CIFAR-10 dataset to classify six animal classes (bird, cat, deer, dog, frog, and horse). Following the setup in Huang et al. (2023), we sampled 400 images per class for the labeled dataset and included up to 4100 images per class in the unlabeled dataset. To investigate the impact of class distribution mismatch, we varied the mismatch percentage

Table 1: Performance comparison on four tasks. We report the mean accuracy averaged over three seeds, along with standard error. Top scores for each task are highlighted.

Task Name	MixMatch	FixMatch	MPL	OpenMatch	Fix-A-Step	IOMatch	DWD-SL
SixAnimal (ζ = 75%)	80.77±0.11	82.50±0.16	65.62±0.47	80.34±0.21	85.34±0.17	83.05±0.16	**85.86±0.28**
CIFAR-10/100	71.02±0.32	78.91±0.15	70.95±0.34	70.15±0.30	74.60±0.31	77.66±0.22	**80.05±0.14**
ImageNet-30	68.67±0.37	70.07±0.26	72.65±0.70	72.78±0.48	79.67±0.81	79.23±0.29	**82.20±0.38**
ImageNet-100	69.30±0.41	65.11±0.32	68.43±0.33	65.42±0.36	65.80±0.49	66.85±0.19	**82.81±0.31**

Table 2: Performance of standard SSL methods before and after applying DWD-UT. Highlighted scores show significant increases without overlapping intervals.

Task Name	MixMatch	Mixmatch +DWD-UT	FixMatch	FixMatch +DWD-UT	MPL	MPL +DWD-UT
SixAnimal (ζ = 75%)	80.77±0.11	**84.72±0.22**	82.50±0.16	**87.17±0.19**	65.62±0.47	**83.88±0.18**
Cifar-10/100	71.02±0.32	**80.47±0.49**	78.91±0.15	**83.80±0.25**	70.95±0.34	**80.24±0.56**
ImageNet-30	68.67±0.37	**85.20±0.10**	70.07±0.26	**81.87±0.61**	72.65±0.70	**90.20±0.23**
ImageNet-100	69.30±0.41	**81.62±0.36**	65.11±0.32	**80.38±0.34**	68.43±0.33	**75.66±0.26**

Table 3: Performance of open-set SSL methods before and after applying DWD-UT.

Task Name	OpenMatch	OpenMatch +DWD-UT	Fix-A-Step	Fix-A-Step +DWD-UT	IOMatch	IOMatch +DWD-UT
SixAnimal (ζ = 75%)	80.34±0.21	**85.71±0.33**	85.34±0.17	**86.68±0.23**	83.05±0.16	**87.20±0.13**
Cifar-10/100	70.15±0.30	**80.99±0.03**	74.60±0.31	**79.02±0.75**	77.66±0.22	**83.22±0.16**
ImageNet-30	72.78±0.48	**75.28±0.68**	79.67±0.81	**82.95±0.45**	79.23±0.29	**81.96±0.26**
ImageNet-100	65.42±0.36	**80.02±0.45**	65.80±0.49	**76.23±0.37**	66.85±0.19	**80.19±0.13**

ζ in the composition of the unlabeled dataset. For example, when ζ = 75%, the unlabeled dataset contains three non-animal classes and one animal classes. We refer Appendix C for further details on the composition of the unlabeled dataset.

The CIFAR-10/100 task uses CIFAR-10 as the labeled dataset, and CIFAR-100 as the unlabeled dataset. While the whole CIFAR-100 was taken as the unlabeled dataset, we sampled 100 images per class from CIFAR-10 to simulate the scarce labeled data scenario. Notably, class labels in CIFAR-10 and CIFAR-100 do not exactly overlap, though there are similarities (e.g., "horse" in CIFAR-10 and "cattle" in CIFAR-100). Thus, this task complements the SixAnimal task, which had an exact class overlap between labeled and unlabeled data.

The ImageNet-30 task uses the ImageNet-30 dataset Hendrycks et al. (2019), which is a subset of ImageNet limited to 30 classes. Following Saito et al. (2021), we selected 5% of the data from the first 20 classes (approximately 50 samples per class) based on the alphabetical ordering of class names for the labeled dataset, and used the remaining data as the unlabeled dataset.

The ImageNet-100 task uses ImageNet-100 dataset, which sub-sampled 100 classes from ImageNet, as described in Cao et al. (2022). We divided these classes equally into 50% ID and 50% OOD classes following alphabetical order. From each ID class, we selected a small portion (10%) as labeled data with the remaining data forming the unlabeled dataset. This task assesses the effectiveness on higher-resolution images with a greater diversity of classes. Please refer to Appendix D for extensive results under various sizes of labeled dataset.

Baseline SSL methods Since we use DWD to transform the unlabeled dataset into a dataset devoid of class distribution mismatch, we comprehensively consider as baseline SSL methods those which operate under the standard setting as well as the open-set setting. The baseline SSL methods under the standard setting are MixMatch (Berthelot et al., 2019), FixMatch (Sohn et al., 2020), and Meta Pseudo Labels (MPL) (Pham et al., 2021), and the methods under the open-set setting are OpenMatch (Saito et al., 2021), Fix-A-Step (Huang et al., 2023), and IOMatch (Li et al., 2023).

5.1 DWD-SL: Labeled Dataset Augmentation

Table 1 reports the comparative performance of DWD-SL against various baseline SSL methods. The results clearly demonstrate that DWD-SL substantially surpasses the performance of methods for standard SSL. This suggests that DWD successfully captures both the inherent distribution of the labeled data and the diversity of the unlabeled data, yielding highly effective synthetic labeled data for training.

Notably, DWD-SL even achieves competitive or superior results relative to open-set SSL methods. This advantage stems from DWD-SL's ability to transform the diversity of unlabeled data into labeled samples, rather than using this diversity merely as a form of regularization, as seen in baseline SSL methods. For further implementation details, please refer to Appendix C.

5.2 DWD-UT: Unlabeled Dataset Transformation

Table 2 and Table 3 show the impact of DWD-UT on the performances of the baseline SSL methods. From the results, we can confirm that DWD-UT effectively addresses the performance degradation caused by the class distribution mismatch. Notably, the most significant improvement was observed in the MPL method. Given that MPL is based on pseudo-labeling, this result indicates that DWD-UT effectively mitigates the confirmation bias associated with pseudo-labeling of OOD instances in the unlabeled dataset.

Figure 3 shows how performance degrades over the range of ζ in the SixAnimal task. We can clearly see that using DWD-UT makes SSL methods generally robust to the degree of class distribution mismatch, even though they operate under the assumption that there are no OOD instances in the unlabeled data.

We also remark that DWD-UT further improves the performance of open-set SSL methods. This implies that DWD-UT is orthogonal to the OOD mitigation mechanisms used in open-set SSL methods, offering a distinct contribution in addressing the class distribution mismatch: while most of the open-set SSL methods operate under the detect-and-filter paradigm to focus on excluding the OOD instances due to their negative impact, the diffusion model provides a powerful tool for making up the diversity lost by such exclusion.

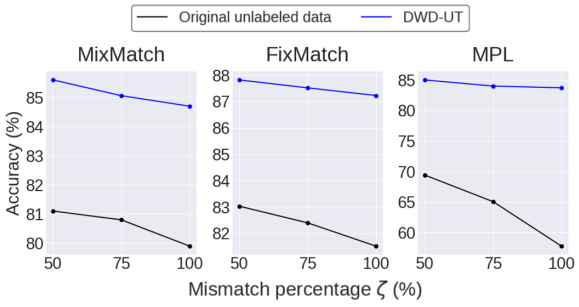

Figure 3: Standard SSL performance with varying ζ.

It is also notable that DWD-UT frequently outperforms DWD-SL, indicating a synergistic effect between DWD-UT and baseline SSL methods. This may be attributed the underlying data selection mechanism in the SSL methods (e.g. thresholding used in pseudo-labeling, weighting function in filtering-based methods), which also contribute to selectively strengthen the impact of synthetic data.

Additionally, to provide direct evidence of DWD-UT's effectiveness in reducing the class distribution mismatch, we compute the minimum distance between each unlabeled data sample and the class centroids of labeled data in the latent space. We refer Appendix E for the results.

5.3 Ablation Studies

We carried out a series of ablation studies on DWD-UT, assessing different training schemes for the diffusion model and varying noise levels for perturbing seed images. For these studies, we employed MPL as the baseline SSL method.

Table 4 shows the results of ablation studies conducted on different training schemes for the diffusion model. Several observations can be drawn from these results. Firstly, the utilization of the discriminator to filter or reduce the weight of OOD instances culminates in a rather marginal performance improvement. Secondly, the incorporation of the diffusion model to generate synthetic data contributes to a substantial performance surge above the baseline, even when fine-tuned solely

Table 4: MPL performance using different training schemes. The notation $\boldsymbol{\epsilon}_\theta[X]$ indicates the inclusion of component X in finetuning the diffusion model. MPL + $\mathbf{d}_\phi$ represents that the discriminator is utilized for filtering unlabeled data.

Training Method	SixAnimal	Cifar-10/100	ImageNet-30	ImageNet-100
MPL	65.62	70.95	72.65	68.43
MPL + $\mathbf{d}_\phi$	67.19	71.73	83.60	70.74
MPL + $\boldsymbol{\epsilon}_\theta[\mathcal{D}_l]$	78.70	75.33	87.44	73.38
MPL + $\boldsymbol{\epsilon}_\theta[\mathcal{D}_l, \mathcal{D}_u]$	80.78	76.79	88.09	74.12
MPL + $\boldsymbol{\epsilon}_\theta[\mathcal{D}_l, \mathcal{D}_u, \mathbf{d}_\phi]$	**83.88**	**80.24**	**90.20**	**75.66**

Table 5: MPL performance on SixAnimal with varying noise levels. DWD-UT is not applied at $t = 0$.

Noise Level	$t = 0$	$t = 200$	$t = 400$	$t = 600$	$t = 800$	$t = 1000$
Accuracy (%)	65.62±0.47	73.65±0.28	82.03±0.18	**83.88±0.18**	**83.83±0.11**	82.06±0.13

Table 6: Performance of standard SSL and generative augmentation methods on ImageNet-30.

Method	**MixMatch**	**FixMatch**	**MPL**	**DPT**	**DA-Fusion**	**DWD-SL**
Accuracy (%)	68.67±0.37	70.07±0.26	72.65±0.70	78.28±0.48	75.26±0.39	**82.20±0.28**

with the labeled data. This suggests that transforming irrelevant unlabeled data is more effective than simply filtering them out. Lastly, the extra step of fine-tuning the diffusion model with the unlabeled dataset and applying discriminative weighting also offers a nontrivial advantage.

Table 5 shows the variations in performance with different noise levels during the data transformation process. The results indicate that introducing a moderate level of noise ($t = 600 \sim 800$) to the unlabeled data during the forward diffusion process, as opposed to initiating from pure Gaussian noise ($t = 1000$), enhances performance. Therefore, it can be inferred from these findings that the efficacy of DWD-UT is contingent upon the balance between the level of noise and the information contained in the unperturbed data. We remark that the optimal noise level can differ among data instances. While we fixed the noise level to $t = 600$ in our experiments for simplicity, determining the noise level individually for each sample presents a potential avenue for future research.

5.4 Comparison with Recent Diffusion-based Augmentation Approaches

We conducted further experiments on the ImageNet-30 task to compare the performance of DWD with those of DPT and DA-Fusion. To ensure fairness in comparison, we equalized the implementation of the model structure, data generation process, and the number of augmented data. As shown in Table 6, both DPT and DA-Fusion have demonstrated effectiveness, yet their performance falls short compared to that of DWD. This is because DPT, assuming no distribution shift, sometime generates wrongly labeled synthetic images due to the confirmation bias in pseudo labeld of OOD unlabeled images, and DA-Fusion only augments given labeled samples with subtle visual details (Please refer to Appendix I for examples). In contrast, DWD synthesizes new labeled samples by transforming diverse unlabeled data, successfully resolving the distribution mismatch.

5.5 Additional Experiments

We also conducted additional experiments to analyze computational costs, investigate the effect of the number of generated data, and assess hyper-parameter sensitivity. For detailed experimental results and analysis, please refer to Appendix F.

6 Conclusion

In this paper, we highlighted the potential of diffusion models for addressing class distribution mismatch in SSL. We introduced Discriminator-Weighted Diffusion (DWD), a semi-supervised training scheme that leverages a discriminator to identify OOD instances within the unlabeled data, facilitating effective training of the diffusion model. Our qualitative and quantitative results demonstrate that DWD captures both the characteristics of labeled data and the diversity of unlabeled data.

Notably, DWD exhibits a unique capability to convert irrelevant samples into relevant ones, making it compatible with other SSL methods and illustrating the orthogonality of our approach. Our extensive experiments show that DWD significantly enhances SSL performance in scenarios with class distribution mismatch. We hope that DWD will inspire future research focused on addressing distribution mismatch from a data-centric perspective.

Acknowledgments and Disclosure of Funding

This work was supported by IITP grant funded by MSIT of Korea (No. RS-2020-II200940, No. RS-2022-II220311, No. RS-2019-II190075, No. RS-2024-00397310, No. RS-2024-00343989, No. RS-2024-00457882), and KAIST-NAVER Hypercreative AI Center.

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE, 2020.

Bachman, P., Alsharif, O., and Precup, D. Learning with Pseudo-Ensembles. Advances in Neural Information Processing Systems, 27, 2014.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. MixMatch: A Holistic Approach to Semi-Supervised Learning. Advances in Neural Information Processing Systems, 32, 2019.

Cao, K., Brbic, M., and Leskovec, J. Open-World Semi-Supervised Learning. International Conference on Learning Representations, 2022.

Chapelle, O., Scholkopf, B., and Zien, Eds., A. Semi-Supervised Learning. IEEE Transactions on Neural Networks, 20(3):542–542, 2009.

Chen, Y., Zhu, X., Li, W., and Gong, S. Semi-Supervised Learning under Class Distribution Mismatch. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 3569–3576, 2020.

Christoffel, M., Niu, G., and Sugiyama, M. Class-prior Estimation for Learning from Positive and Unlabeled Data. In Asian Conference on Machine Learning, pp. 221–236. PMLR, 2016.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. RandAugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 702–703, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009.

Du Plessis, M., Niu, G., and Sugiyama, M. Convex Formulation for Learning from Positive and Unlabeled Data. In International Conference on Machine Learning, pp. 1386–1394. PMLR, 2015.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2):303–338, 2010.

Feller, W. On the Theory of Stochastic Processes, with Particular Reference to Applications, 1949.

Grandvalet, Y. and Bengio, Y. Semi-supervised Learning by Entropy Minimization. Advances in Neural Information Processing Systems, 17, 2004.

Guo, L.-Z., Zhang, Z.-Y., Jiang, Y., Li, Y.-F., and Zhou, Z.-H. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In International Conference on Learning Representations, pp. 3897–3906. PMLR, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

He, R., Han, Z., Lu, X., and Yin, Y. Safe-Student for Safe Deep Semi-Supervised Learning with Unseen-Class Unlabeled Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 14585–14594, 2022.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. Advances in Neural Information Processing Systems, 32, 2019.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

Huang, J., Fang, C., Chen, W., Chai, Z., Wei, X., Wei, P., Lin, L., and Li, G. Trash to Treasure: Harvesting OOD Data with Cross-Modal Matching for Open-Set Semi-Supervised Learning. In Proceedings of the IEEE International Conference on Computer Vision, pp. 8310–8319, 2021.

Huang, Z., Yang, J., and Gong, C. They are Not Completely Useless: Towards Recycling Transferable Unlabeled Data for Class-Mismatched Semi-Supervised Learning. IEEE Transactions on Multimedia, 2022.

Huang, Z., Sidhom, M.-J., Wessler, B., and Hughes, M. C. Fix-A-Step: Semi-supervised Learning from Uncurated Unlabeled Data. In International Conference on Artificial Intelligence and Statistics, pp. 8373–8394. PMLR, 2023.

Jain, S., White, M., and Radivojac, P. Estimating the class prior and posterior from noisy positives and unlabeled data. Advances in Neural Information Processing Systems, 29, 2016.

Kiryo, R., Niu, G., Du Plessis, M. C., and Sugiyama, M. Positive-Unlabeled Learning with Non-Negative Risk Estimator. Advances in Neural Information Processing Systems, 30, 2017.

Krizhevsky, A. and Hinton, G. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, Toronto, Ontario, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 60(6):84–90, 2017.

Laine, S. and Aila, T. Temporal Ensembling for Semi-Supervised Learning. In International Conference on Learning Representations, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. Nature, 521(7553):436–444, 2015.

Lee, D.-H. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In Workshop on challenges in representation learning, ICML, volume 3, pp. 896, 2013.

Li, X. and Liu, B. Learning to Classify Texts Using Positive and Unlabeled Data. In International Joint Conference on Artificial Intelligence, volume 3, pp. 587–592, 2003.

Li, Z., Qi, L., Shi, Y., and Gao, Y. IOMatch: Simplifying Open-Set Semi-Supervised Learning with Joint Inliers and Outliers Utilization. In Proceedings of the IEEE International Conference on Computer Vision, pp. 15870–15879, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common Objects in Context. In Proceedings of the IEEE European Conference on Computer Vision, pp. 740–755, 2014.

Liu, B., Lee, W. S., Yu, P. S., and Li, X. Partially Supervised Classification of Text Documents. In International Conference on Machine Learning, volume 2, pp. 387–394, 2002.

Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In International Conference on Learning Representations, 2019.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. Advances in Neural Information Processing Systems, 35:5775–5787, 2022.

McLachlan, G. J. Iterative Reclassification Procedure for Constructing an Asymptotically Optimal Rule of Allocation in Discriminant Analysis. Journal of the American Statistical Association, 70 (350):365–369, 1975.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In Advances in Neural Information Processing Systems, 2022.

Menon, A., Van Rooyen, B., Ong, C. S., and Williamson, B. Learning from Corrupted Binary Labels via Class-Probability Estimation. In International Conference on Machine Learning, pp. 125–134. PMLR, 2015.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8):1979–1993, 2018.

Niu, G., Du Plessis, M. C., Sakai, T., Ma, Y., and Sugiyama, M. Theoretical Comparisons of Positive-Unlabeled Learning against Positive-Negative Learning. Advances in Neural Information Processing Systems, 29, 2016.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. Advances in Neural Information Processing Systems, 31, 2018.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. Meta Pseudo Labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11557–11568, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10684–10695, 2022.

Saito, K., Kim, D., and Saenko, K. OpenMatch: Open-set Consistency Regularization for Semi-supervised Learning with Outliers. Advances in Neural Information Processing Systems, 34: 25956–25967, 2021.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. Advances in Neural Information Processing Systems, 29, 2016.

Scudder, H. Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory, 11(3):363–371, 1965.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In International Conference on Machine Learning, pp. 2256–2265. PMLR, 2015.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. Advances in Neural Information Processing Systems, 33:596–608, 2020.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in Neural Information Processing Systems, 30, 2017.

Trabucco, B., Doherty, K., Gurinas, M. A., and Salakhutdinov, R. Effective Data Augmentation With Diffusion Models. In International Conference on Learning Representations, 2024.

Ward, G., Hastie, T., Barry, S., Elith, J., and Leathwick, J. R. Presence-only data and the em algorithm. Biometrics, 65(2):554–563, 2009.

You, Z., Zhong, Y., Bao, F., Sun, J., Li, C., and Zhu, J. Diffusion Models and Semi-Supervised Learners Benefit Mutually with Few Labels. In Advances in Neural Information Processing Systems, 2023.

Zagoruyko, S. and Komodakis, N. Wide Residual Networks. arXiv preprint arXiv:1605.07146, 2016.

A Derivation for Proposition 4.1

Problem setting. We consider the two-sample problem setting of PU learning (Ward et al., 2009; Niu et al., 2016). The discriminator $\mathbf{d}(\mathbf{x})$ aims to solve a binary classification problem to classify input data $\mathbf{x}$ into negative or positive class y. Let $X \in \mathbb{R}^d$ and $Y \in \{-1, +1\}$ denote the input and output random variables, $p(\mathbf{x}, y)$ be the joint probability density function of (X, Y), and $p^+(\mathbf{x}) = p(\mathbf{x} \,|\, Y = +1)$, $p^-(\mathbf{x}) = p(\mathbf{x} \,|\, Y = -1)$ be the positive and negative conditional probability density functions respectively. The labeled and unlabeled data are assume to be sampled from $p^+(\mathbf{x})$ and $p(\mathbf{x}) = \mu\, p^+(\mathbf{x}) + (1-\mu)\, p^-(\mathbf{x})$ respectively, where $\mu = p(Y = +1)$ is class prior.

Proof. The goal of the discriminator can be formulated as maximizing following objective

$$\begin{aligned} J &= \mathbb{E}_{\mathbf{x}\sim p^+(\mathbf{x})} \left[\log\left(\mathbf{d}\left(\mathbf{x}\right)\right)\right] + \mathbb{E}_{\mathbf{x}\sim p^-(\mathbf{x})} \left[\log\left(1 - \mathbf{d}\left(\mathbf{x}\right)\right)\right] \\ &= \int p^+(\mathbf{x}) \log\left(\mathbf{d}\left(\mathbf{x}\right)\right) + p^-(\mathbf{x}) \log\left(1 - \mathbf{d}\left(\mathbf{x}\right)\right) \mathrm{d}\mathbf{x} \end{aligned} \tag{10}$$

First order optimality condition gives

$$\frac{p^+(\mathbf{x})}{\mathbf{d}(\mathbf{x})} - \frac{p^-(\mathbf{x})}{1 - \mathbf{d}(\mathbf{x})} = 0 \tag{11}$$

Rearranging (11), we can easily show that the optimal discriminator $\mathbf{d}^*(\mathbf{x})$ satisfies

$$\mathbf{d}^*(\mathbf{x}) = \frac{p^+(\mathbf{x})}{p^+(\mathbf{x}) + p^-(\mathbf{x})} \tag{12}$$

Substituting (12) to (8), we have

$$w(\mathbf{x}) = \frac{\mathbf{d}^*(\mathbf{x})}{\mu\, \mathbf{d}^*(\mathbf{x}) + (1-\mu)(1 - \mathbf{d}^*(\mathbf{x}))} = \frac{p^+(\mathbf{x})}{\mu\, p^+(\mathbf{x}) + (1-\mu)\, p^-(\mathbf{x})} = \frac{p^+(\mathbf{x})}{p(\mathbf{x})} \tag{13}$$

Finally, we can use $w(\mathbf{x})$ as importance weights because

$$\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x})}[w(\mathbf{x}_0)||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2] = \int \frac{p^+(\mathbf{x}_0)}{p(\mathbf{x}_0)} p(\mathbf{x}_0)||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2 \mathrm{d}\mathbf{x} = \mathbb{E}_{\mathbf{x}_0\sim p^+(\mathbf{x})}[||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2] \tag{14}$$

B Data Generation Process

The data generation process is detailed in Algorithm 2. In our experiments, we generate one image per unlabeled image. For the impact of the number of generated data, please refer to Appendix F. We chose to generate images before the classification task, rather than during each batch iteration of training the classification models. Additionally, while we described the sampling process in DDPM style, DPM-Solver[2](Lu et al., 2022) is utilized in implementation for computational efficiency.

Algorithm 2 DWD - Image-Seeded Generation

Input Unlabeled data $\mathbf{x}^u$ and diffusion model $\boldsymbol{\epsilon}_\theta$.
Parameter Time step t.
Sample Gaussian noise: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Forward diffusion process to time step t: $\mathbf{x}_t^u = \sqrt{\bar{\alpha}_t}\mathbf{x}^u + \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon}$
Randomly select class condition $\mathbf{c}$.
# Reverse diffusion process starting at $\mathbf{x}_t^u$:
for $i = t, t-1, ..., 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $i > 1$ else $\mathbf{z} = 0$
  $\mathbf{x}_{i-1}^u = \frac{1}{\sqrt{\alpha_i}}\left(\mathbf{x}_i^u - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_i}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_i^u, \mathbf{c}, i)\right) + \sqrt{\beta_i}\mathbf{z}$
end for
Output Transformed data $\mathbf{x}_0^u$ and class condition $\mathbf{c}$

[2]https://github.com/LuChengTHU/dpm-solver, MIT License

C Implementation Details

Diffusion model and discriminator. For all experiments, we use official implementation of latent diffusion model (Rombach et al., 2022), which is publicly available[3]. Since latent diffusion models perform diffusion processes in the embedded latent space, there is a trade-off between computational cost and generation quality depending on the downsampling factor f Rombach et al. (2022). Therefore, we adjust f based on dataset scale: $f = 2$ for small-scale datasets and $f = 8$ for large-scale datasets. The batch sizes for labeled and unlabeled data, denoted as B_l and B_u respectively, are set to $B_l = 16$, $B_u = 112$ for small-scale datasets, and $B_l = 4$, $B_u = 12$ for large-scale datasets. We follow the Rombach et al. (2022) for other configuration such as learning rate, optimizer, scheduler, etc. After the pre-training phase, we train the models for 200K iterations for Cifar-10, and 1M iterations for ImageNet. For the discriminator, we employ the ResNet-18 He et al. (2016) followed by 2 MLP layers. We train the discriminator using AdamW Loshchilov & Hutter (2019) optimizer with 0.0002 initial learning rate and 0.0001 weight decay. We treat μ as a hyper-parameter, and set it within {0.25, 0.33, 0.5}. Another hyper-parameter α, which control the balance between labeled and unlabeled, we set 5 in all tasks.

Common configuration for DWD-SL and DWD-UT. To ensure fair evaluation, task-specific settings were established for both DWD-SL and DWD-UT. For tasks associated with Cifar-10, the Wide ResNet-28-2 architecture Zagoruyko & Komodakis (2016)) was employed, with training conducted using the AdamW optimizer at an initial learning rate of 0.03 across 256 epochs 1,024 iterations per epoch. In the ImageNet-30 task, we follow the settings from Saito et al.(2021) and Li et al.(2023). Specifically, we employed the ResNet-18 architecture He et al. (2016), and train for 100 epochs with 1,024 iterations per epoch using AdamW 0.1 initial learning rate.

DWD-SL specific configuration. In all DWD-SL tasks, we maintained a 1:1 ratio between labeled and unlabeled samples within each batch. More specifically, we set $B_l = B_u = 64$ for SixAnimal and Cifar-10/100, and $B_l = B_u = 32$ for ImageNet-30. We applied RandAugment Cubuk et al. (2020), widely used in the SSL field to achieve robust results, to both original labeled and DWD-SL data. Additionally, we applied label smoothing to the cross-entropy loss, following the approach used in MPL Pham et al. (2021). The starting time step t for the reverse diffusion process was set to 600.

DWD-UT specific configuration. For SixAnimal and Cifar-10/100 tasks, we used $B_l = 64$ and $B_u = 448$, while for ImageNet-30, we used $B_l = 32$ and $B_u = 32$. Since ImageNet-30 includes out-of-distribution (OOD) classes in the test set while standard SSL methods inherently cannot identify OOD classes, we removed OOD samples from the test set for a fair comparison. Additionally, IOMatch evaluates the performance using balanced accuracy, which classifies all OOD classes as a additional single class. This could also be an unfair comparison. Therefore, we evaluated only on the closed set for all tasks. Except aforementioned, we follow their papers for method-specific hyper-parameters and setting.

Unlabeled data composition in SixAnimal with varying ζ. We configure the SixAnimal task exactly following Chapelle et al. (2009), Huang et al. (2023). The Table 7 shows the composition of labeled / unlabeled set of SixAnimal task according to mismatch percentage (ζ).

Table 7: Configuration of labeled/unlabeled class mismatch scenario in CIFAR-10 SixAnimal task. The bold text of unlabeled set represent the OOD classes.

	Labeled set	Unlabeled set
$\zeta = 25\%$	Bird, Cat, Deer, Dog, Frog, Horse	**Airplane**, Dog, Frog, Horse
$\zeta = 50\%$	Bird, Cat, Deer, Dog, Frog, Horse	**Airplane, Car**, Frog, Horse
$\zeta = 75\%$	Bird, Cat, Deer, Dog, Frog, Horse	**Airplane, Car, Ship**, Horse
$\zeta = 100\%$	Bird, Cat, Deer, Dog, Frog, Horse	**Airplane, Car, Ship, Truck**

[3]https://github.com/CompVis/latent-diffusion, MIT License

D Extended Evaluation on Class Mismatch and Labeled Data Ratios

Different mismatch ratios of ID and OOD classes in the SixAnimal task. As shown in Figure 3 of our paper, we conducted experiments on the SixAnimal task across various ratios of ID and OOD classes (ζ = 50%, 75%, 100%). To extend these findings, we included additional experiments with a lower mismatch ratio, ζ = 25%. Since DWD is designed to address class distribution mismatch, its effectiveness is expected to decrease as the mismatch ratio lowers. However, we observed that DWD was still able to improve the baseline method in the ζ = 25% case, although the performance gain was relatively diminished.

Table 8: Performance evaluation on the SixAnimal task with various ratios of ID and OOD classes (ζ).

Mismatch ratio	MixMatch	MixMatch +DWD-UT	FixMatch	FixMatch +DWD-UT	MPL	MPL +DWD-UT
ζ = 25%	83.78	**85.89 (+2.11)**	84.56	**85.81 (+1.25)**	79.70	**84.76 (+ 5.06)**
ζ = 50%	81.16	**84.83 (+3.67)**	83.23	**87.78 (+4.55)**	69.43	**85.02 (+15.59)**
ζ = 75%	80.77	**84.72 (+3.95)**	82.50	**87.17 (+4.67)**	65.62	**83.88 (+18.26)**
ζ = 100%	79.90	**84.08 (+4.18)**	81.51	**87.03 (+5.52)**	57.77	**83.73 (+25.96)**

Varying the size of labeled data in the ImageNet-100 task. To broaden our evaluation, we conducted additional experiments on the ImageNet-100 dataset, varying the amount of labeled data. Specifically, we used either 10% or 30% of each ID class as labeled data, with the remaining samples forming the unlabeled set. As shown in Table 9, DWD remains effective with different amounts of labeled data, demonstrating strong performance at both 10% and 30% sampling ratios. Notably, the performance gain is more pronounced at the 10% sampling ratio, as the advantages of data augmentation become clearer with smaller datasets. However, DWD's effectiveness may be constrained when labeled data is extremely limited, as the diffusion model may struggle to accurately represent the labeled data distribution. This limitation is also observed in other generative augmentation methods.

Table 9: Performance evaluation on the ImageNet-100 task with varying sampling ratio (γ).

Sampling ratio	MixMatch	FixMatch	MPL	OpenMatch	Fix-A-Step	IOMatch	DWD-SL
γ = 10%	69.30	65.11	68.43	65.42	65.80	66.85	**82.81**
γ = 30%	77.88	75.83	71.67	77.31	73.90	76.02	**84.43**

Sampling ratio	MixMatch	MixMatch +DWD-UT	FixMatch	FixMatch +DWD-UT	MPL	MPL +DWD-UT
γ = 10%	69.30	**81.62 (+12.32)**	65.11	**80.38 (+15.27)**	68.43	**75.66 (+7.23)**
γ = 30%	77.88	**82.26 (+ 4.38)**	75.83	**81.35 (+ 5.52)**	71.67	**77.49 (+5.82)**

Sampling ratio	OpenMatch	OpenMatch +DWD-UT	Fix-A-Step	Fix-A-Step +DWD-UT	IOMatch	IOMatch +DWD-UT
γ = 10%	65.42	**80.02 (+14.60)**	65.80	**76.23 (+10.43)**	66.85	**80.19 (+13.34)**
γ = 30%	77.31	**81.50 (+ 4.19)**	73.90	**78.43 (+ 4.53)**	76.02	**81.52 (+ 5.50)**

E Calculating Distance in Latent Space

We extract features (labeled, original unlabeled, transformed unlabeled) from four datasets using a pre-trained ResNet-50. The features were normalized, and the pair-wise Euclidean distances between each unlabeled sample and the nearest class centroid is visualized using Gaussian Kernel Density Estimation (KDE). As shown in Figure 4, the minimum distances successfully decrease after DWD's data transformation. This result indicates that using DWD to transform unlabeled data does more than just make it look similar to labeled data; it also makes the data semantically similar in the latent space, showing a deeper level of similarity beyond just appearance.

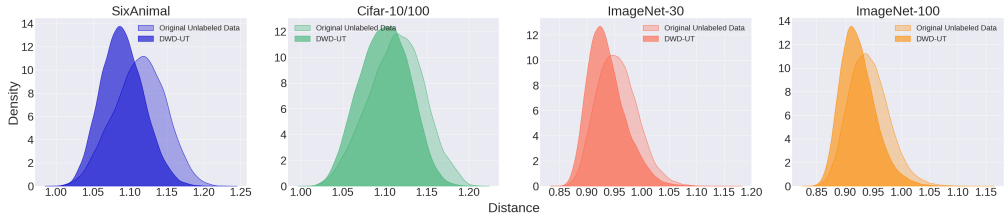

Figure 4: The distribution of the minimum distance between each unlabeled data sample and the class centroids of labeled data.

F Additional Experiments

Computational cost analysis. The main limitation of our work is the additional computation required by incorporating the diffusion model. On the CIFAR-10/100 task, we measured the wall-clock times and memory consumption for each stage of DWD: pretraining, finetuning (including discriminator training), and sampling. We compare the computation costs with standard SSL methods, and the results are reported in Table 10. The additional computation for DWD is not overly burdensome compared to the baselines. It is worth noting that while we include pretraining costs for completeness, these can be omitted when off-the-shelf pre-trained diffusion models are available.

Table 10: Computational Cost Comparison.

	DWD			Baselines		
	Pretraining	Finetuning	Sampling	Mixmatch	Fixmatch	MPL
Elapsed time (Hours)	13.8	9.7	0.1	9.2	7.4	6.7
Memory (GB)	7.1	8.0	7.0	4.9	5.5	7.4

[Machine specification] GPU : NVIDIA GeForce RTX 3090 Ti, CPU : Intel(R) Core(TM) i9-10980XE

Effect of the number of generated data. We conducted additional experiments using DWD-SL on the Cifar-10/100 task, varying the number of generated data denoted as N. The results are summarized in the Table 11. Here, N = 60K corresponds to the original setting in the paper where one synthetic labeled sample per one unlabeled sample is generated. We observed further improvement at N = 120K, with a slight deterioration thereafter. Note that a similar trend was previously reported in Figure 7(c) of Appendix G in You et al. (2023). A reasonable explanation for the performance deterioration is that an excessive value of N could cause the classifier to be dominated by synthetic data, thereby neglecting real data, as suggested by You et al. (2023).

Table 11: Effect of the number of generated data

N	6K	30K	60K	120K	240K
Accuracy (%)	77.64	79.17	80.05	**81.24**	81.07

Hyper-parameters sensitivity. We conducted additional experiments on the SixAnimal task (ζ=75%) using DWD-SL to assess DWD's sensitivity to the hyper-parameters. Again, α serves as weight to control the balance between labeled and unlabeled data (Eq. 9) and μ is treated as positive sample ratio of unlabeled data to train discriminator (Eq. 7) in our training scheme. We observed that a wide range of α and μ values successfully outperform most of the baselines (refer to Table 1 in our paper). Regarding α, an extremely small value may cause the diffusion model training to focus excessively on the labeled data, failing to reflect the diversity of the unlabeled data and potentially leading to overfitting. Conversely, an extremely large value may cause the training to skew towards the unlabeled data, failing to properly capture the labeled data distribution. In our experiments, an α value around 3 achieves an appropriate trade-off. Regarding μ, the optimal value is near the true ratio 1 - ζ, as expected.

Table 12: Performance of DWD with various α.

α	1	**3**	5	10
Accuracy (%)	84.01	**85.86**	83.83	83.51

Table 13: Performance of DWD with various μ.

μ	0.125	**0.25**	0.33	0.55
Accuracy (%)	84.56	**85.86**	85.33	84.72

G Images Corresponding to Discriminator's Output

As shown in the Figure 5, our discriminator successfully identifies the ID/OOD samples. ImageNet-30 consist of 20 in-domain (ID) classes (acorn, airliner, ambulance, american alligator, banjo, barn, bikini, digital clock, dragonfly, dumbbell, forklift, goblet, grand piano, hotdog, hourglass, manhole cover, mosque, nail, parking meter, pillow) and 10 out-of-domain (OOD) classs (revolver, rotary dial telephone, schooner, snowmobile, soccer ball, stingray, strawberry, tank, toaster, volcano)

(a) High scored samples (b) Low scored samples (c) High scored samples (d) Low scored samples

Figure 5: Selected unlabeled samples based on discriminator's output on the SixAnimal (a, b) and ImageNet-30 (c, d).

H Generated Images

Figure 6: Selected Examples of Data Geneartion Process.

I Generated Images from DPT and DA-Fusion

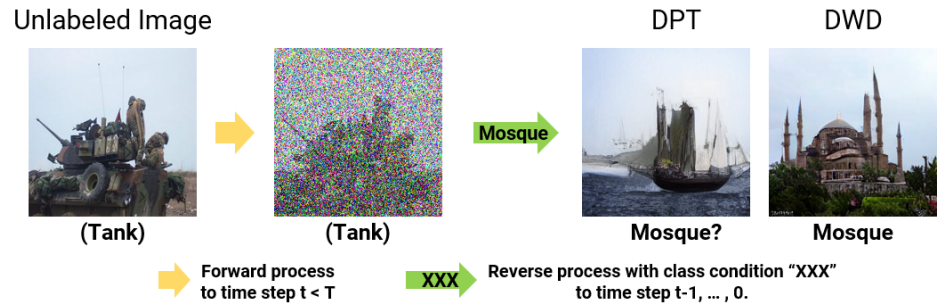

Figure 7: Generated images from DPT and DWD. DPT sometimes generates incorrectly labeled samples (e.g., an image of a schooner, which is an OOD class, labeled as a mosque). Note that while DPT originally samples images from scratch, we applied our data generation algorithm to DPT for comparison.

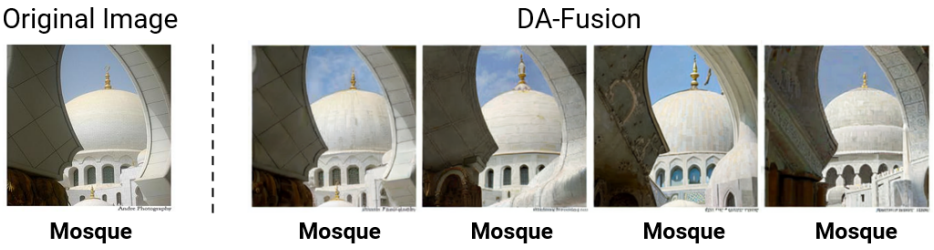

Figure 8: Generated images from DA-Fusion. DA-Fusion only augments given labeled images with subtle visual details.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction reflect the contributions and scope of our work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our main limitation (additional computation) in Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open source repository for reproduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report experimental results with the standard error.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer Appendix F

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewd the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no direct path to any negative application.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the original paper that produced the dataset, and provided URL and license of the existing code we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide open source repository that is well documented.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We did not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We did not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.