# INFORMATIVE DATA REWEIGHTING FOR IMAGE CLASSIFICATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The success of Deep Neural Networks (DNNs) in image classification largely depends on access to large-scale, high-quality labeled datasets, which are often scarce or costly to obtain. To alleviate this challenge, Generative Data Augmentation (GDA) has been introduced to improve model performance by increasing the number of training samples with synthetic data generated by models such as Diffusion Models (DMs). Despite its benefits, GDA-generated synthetic samples often contain noise, which can negatively impact the performance of image classification models when incorporated into training. Prior approaches, including data selection and reweighting techniques, aim to address this issue but often rely on external expert models or clean metadata. In this work, we introduce Informative Data Reweighting (IDR), a principled sample reweighting framework based on the Information Bottleneck (IB) principle, to enhance the performance of DNNs for image classification using GDA. Through extensive experiments, we demonstrate that IDR effectively prioritizes more informative training samples in the augmented training set comprising original real training samples and synthetic training samples, resulting in substantial improvements over existing data selection and reweighting strategies for GDA in image classification. The code for IDR is available at https://anonymous.4open.science/r/IDR-3BE0/.

## 1 INTRODUCTION

Deep Neural Networks (DNNs) have achieved strong performance across computer vision tasks such as image classification, semantic segmentation, and object detection, but their success critically depends on large-scale, high-quality annotated datasets (Feng et al., 2020). However, obtaining such datasets poses substantial challenges for many computer vision tasks (El Jiani et al., 2022; Xiao et al., 2023) due to constraints such as resource limitations and privacy concerns (Esteva et al., 2021; Price & Cohen, 2019; Ali et al., 2023; Ramudu et al., 2023). To mitigate this issue, researchers have explored Generative Data Augmentation (GDA) (Sarıyıldız et al., 2023; Lei et al., 2023; Azizi et al., 2023; Trabucco et al., 2024), leveraging techniques such as Generative Adversarial Networks (GANs) (Zhang et al., 2021; Li et al., 2022) and Diffusion Models (DMs) (He et al., 2023; Tian et al., 2023; Yuan et al., 2022; Bansal & Grover, 2023; Vendrow et al., 2023) to generate synthetic training samples that can be combined with the original training set for data augmentation, leading to an augmented training set.

**Limitations in Existing GDA Methods.** Despite its potential, synthetic samples generated by GDA often contain noise, such as visual artifacts and semantic inconsistencies as illustrated in Figure 2 (a) and (b), which degrade downstream image classification performance (Corvi et al., 2023; Azizi et al., 2023; He et al., 2023). To mitigate this issue, prior work has proposed data selection and sample reweighting strategies that train classifiers on curated or weighted synthetic samples (Chhabra et al., 2024; He et al., 2023), typically relying on meta-networks with clean metadata to assign higher importance to informative samples, while assuming the availability of such clean metadata (Shu et al., 2019; Guo et al., 2022; Jain et al., 2024). A closely related study, CBF (He et al., 2023), filters low-quality synthetic sam-

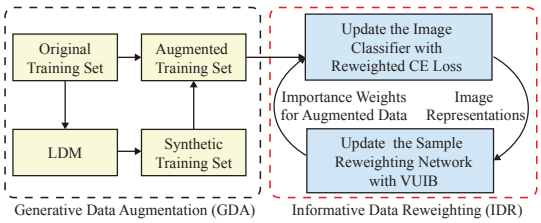



Figure 1: Generative Data Augmentation (GDA) and Informative Data Reweighting (IDR).



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ples using CLIP zero-shot confidence (Radford et al., 2021) in GDA, but its effectiveness relies heavily on CLIP, an expert vision–language model pretrained on large-scale, non-public image–text datasets. GenDataAgent (Li et al., 2025) filters out-of-distribution samples from an expert generative model, Stable Diffusion, using VoG-based logit gradients, while it cannot be employed to select informative in-domain synthetic samples as in GDA (He et al., 2023; Azizi et al., 2023), where the generative model is trained on the same domain. In summary, the current machine learning literature lacks a principled sample reweighting method for GDA, which does not rely on clean metadata or external expert models. We further note that the upper bound for mutual information between the learned features and the input training features in the IB loss in IDS (Wang et al., 2025) and DCS-Transformer (Wang et al., 2024) is vacuous in terms of the learned features, with details in Section E.2 of the appendix. Such drawback of IDS results in its inferior performance compared to our IDR as evidenced in Tables 1, 3, 6, and 7 of Sections 4.1, 4.4, D.3, and D.4.

To address this issue, we propose a novel Informative Data Reweighting (IDR) method for reweighting samples in the augmented training set obtained from GDA, inspired by the Information Bottleneck (IB) principle. As illustrated in Figure 1, IDR starts with GDA to generate a synthetic training set using the LDM trained on the original training set, which is then combined with the original training set to form an augmented training set. IDR aims to assign higher importance weights to more informative training samples in the augmented training set that contribute more to the image classification task. To achieve this goal, IDR trains a sample reweighting network that assigns importance weights to the samples in the augmented training set by reducing the IB loss on them. Let $\tilde{X}$ be the input features, and $\tilde{Z}$ be the learned features by the classification network. Let $\tilde{Y}$ be the ground truth training class labels. The principle of IB is to increase the mutual information between $\tilde{Z}$ and $\tilde{Y}$ while reducing the mutual information between $\tilde{Z}$ and $\tilde{X}$. That is, the IB principle encourages reduction of the IB loss, $I(\tilde{Z}, \tilde{X}) - I(\tilde{Z}, \tilde{Y})$, where $I(\cdot, \cdot)$ denotes mutual information modeling the correlation of its input variables. The reduction of the IB loss is achieved by optimizing a novel variational upper bound for the IB loss, termed VUIB, which can be optimized by standard SGD algorithms. As illustrated in Figure 2 (a) - (b), IDR assigns lower importance weights to samples with visual artifacts and semantic inconsistencies that can degrade the performance of the classification network. In contrast, samples with high fidelity and clear semantic meanings receive higher importance weights as illustrated in Figure 2 (c) - (d). As illustrated in Figure 1, the cross-entropy loss reweighted by the importance weights and the VUIB are iteratively optimized to update the weights in the classification network and the sample reweighting network, inspired by the bi-level optimization method in (Shu et al., 2019).

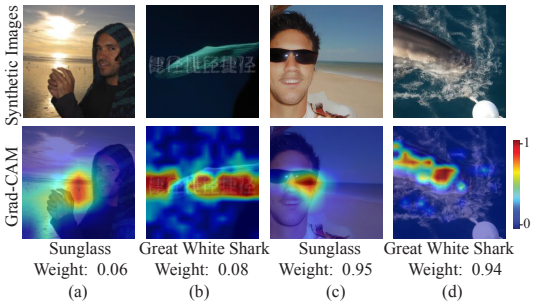



Figure 2: Figures (a) and (b) illustrate synthetic images with low importance weights and Figures (c) and (d) illustrate synthetic images with high importance weights. The Grad-CAM visualizations for the ground truth classes are attached below the images. The class name and importance weights learned by IDR are attached below the Grad-CAM. Figure (a) illustrates **semantic inconsistency** in the image generated for the class sunglass, where Grad-CAM shows that the model incorrectly focuses on the background. Figure (b) illustrates severe **visual artifacts** in the image generated for the class great white shark, where strong corruption and unnatural textures obscure the object structure, and Grad-CAM correspondingly focuses on the corrupted texture instead of meaningful object regions. In contrast, Figures (c) and (d) illustrate high-quality synthetic images without semantic inconsistencies and visual artifacts for the classes sunglass and great white shark, respectively, where Grad-CAM correctly focuses on the ground-truth objects. More qualitative examples are illustrated in Figure 4 in Section D.9 of the appendix.

**Our Contributions.** The contributions of this paper are presented as follows.

First, to the best of our knowledge, IDR is among the first to perform sample reweighting in GDA by employing a principled IB framework in the sample reweighting process, where each sample in the augmented training set receives an importance weight, aiming to improve the accuracy of the classifier trained on the reweighted training samples. A sample reweighting network is optimized

for reducing the IB loss on the augmented training set, such that the IB principle, learning features more strongly correlated with class labels while decreasing their correlation with the inputs, is better adhered to. In contrast to existing works in sample reweighting (Shu et al., 2019; Guo et al., 2022; Jain et al., 2024) and data selection (Chhabra et al., 2024; He et al., 2023), IDR does not require either clean metadata or external expert models.

Second, to facilitate reducing the IB loss under mini-batch–based SGD optimization, we derive a novel, theoretically justified variational upper bound for the IB loss, termed VUIB, which is amenable to optimization with standard SGD algorithms. In contrast to existing upper bounds for the IB loss, such as VIB (Dai et al., 2018; Srivastava et al., 2021), which relies on an impractical Gaussian assumption on the hidden features of deep neural networks, and APIB (Guo et al., 2023), which only reduces an approximation of the IB loss, IDR directly reduces the variational upper bound VUIB without introducing any distributional assumptions on the hidden representations or approximation to the IB loss. Furthermore, the proposed VUIB is computationally efficient, with a complexity of $\Theta(N'T_0 + N'C)$, where $C$ denotes the number of prototypes (to be detailed in Section 3.2), $N'$ represents the number of training samples, and $T_0$ is the computational cost of a forward and backward pass of the neural network for each sample. By comparison, the mutual-information upper bound employed in CLUB (Cheng et al., 2020), while avoiding distributional assumptions on hidden features, incurs a significantly higher computational complexity of $\Theta(N'^2T_0)$, since $N' \gg C$. As shown in Table 2, models trained using VUIB achieve substantially better performance than those based on CLUB, VIB, and APIB. As demonstrated by the results in Section 4.1, IDR significantly outperforms state-of-the-art data reweighting and data selection methods for image classification on ImageNet-1K.

Recent works propose improved conventional data augmentation strategies for image classification based on transformations of the original training samples (Kim & Kim, 2025; Kang et al., 2025). In comparison, IDR leverages GDA to expand the training set and employs informative reweighting to mitigate the noise introduced by synthetic samples, making it orthogonal and complementary to conventional data augmentation approaches. Experiments in Section D.1 demonstrate that IDR outperforms existing conventional augmentation methods and achieves further performance improvements when combined with them. It is noted that while the reweighting is originally developed to render importance scores of synthetic data, it is revealed in this work that reweighting can also be applied to the original real training samples so that more informative real samples contributing more to the classification task receive higher importance weights. Section D.2 presents an ablation study showing that jointly reweighting real and synthetic samples in IDR is more effective than reweighting synthetic samples alone. We use $[N]$ to denote all the natural numbers between 1 and $N$.

It is also worthwhile to mention that the connection between reducing the IB loss in the sample reweighting network and mitigating synthetic data noise is well supported by the t-SNE visualizations in Figures 7 and 8, where samples assigned higher importance weights by IDR form more compact and class-consistent clusters, as well as by the qualitative examples in Figure 2.

## 2 Related Works

**Information Bottleneck (IB) Principle.** The IB principle (Tishby et al., 2000) posits that an optimal DNN compresses its input data, preserving only the information that is essential for predicting the target outputs, thereby increasing the mutual information between the representations and the target outputs while reducing the mutual information between the input and the representations. Deep VIB (Alemi et al., 2017) first integrates the IB principle as a training objective. Both empirical (Lai et al., 2021; Zhou et al., 2022) and theoretical (Amjad & Geiger, 2020; Kawaguchi et al., 2023) works prove that DNNs better adhering to the IB principle show stronger performance.

**Generative Data Augmentation, Data Selection and Sample Reweighting.** Generating informative synthetic training data for data augmentation, commonly referred to as generative data augmentation (GDA), remains an important yet challenging problem. Despite reported performance gains in image classification, such synthetic samples often introduce nontrivial noise (Azizi et al., 2023; Trabucco et al., 2024; Na et al., 2024), highlighting the need for effective quality control. To address this issue, recent studies mainly follow three directions, including improving synthetic data quality (Sarıyıldız et al., 2023; Lei et al., 2023; Zhou et al., 2023), data selection methods that identify high-quality subsets from synthetic data (Song et al., 2023; Lin et al., 2023; He et al., 2023; Chhabra et al., 2024; Li et al., 2025), and data reweighting approaches assigning importance

weights to training samples (Mo et al., 2019; Shu et al., 2019; Guo et al., 2022; Jain et al., 2024). Among existing methods, CBF (He et al., 2023) and GenDataAgent (Li et al., 2025) aim to select high-quality synthetic images for GDA, where CBF relies on CLIP zero-shot classification confidence (Radford et al., 2021), and GenDataAgent filters out-of-distribution samples generated by Stable Diffusion (Rombach et al., 2022) using VoG-based logit gradients (Agarwal et al., 2022).

# 3 Informative Data Reweighting

Given the original training set $\mathcal{D}_{\text{real}} = \{x_i, y_i\}_{i=1}^N$ for image classification where $x_i$ is the input feature with class label $y_i \in [C']$ and $C'$ is the number of classes in the training set, we aim to generate synthetic training set $\mathcal{D}_{\text{syn}} = \{\widehat{x}_i, \widehat{y}_i\}_{i=1}^M$ with diffusion models and train a classifier on the augmented training set $\mathcal{D}_{\text{aug}} = \{\tilde{x}_i, \tilde{y}_i\}_{i=1}^{N'} = \{x_i, y_i\}_{i=1}^N \cup \{\widehat{x}_i, \widehat{y}_i\}_{i=1}^M$ with $N' = N + M$. To mitigate the negative impact of noise in the synthetic training samples, we propose Informative Data Reweighting (IDR) to reweight the training samples in $\mathcal{D}_{\text{aug}}$ with a sample reweighting network. The sample reweighting network is trained by optimizing the variational upper bound for the Information Bottleneck (IB) loss on the augmented training set in the hope that more informative training samples receive higher weights, thus improving the performance of the classifier trained on the augmented training set. In Section 3.1, we describe the details for generating the synthetic training samples with diffusion models. We then introduce our novel variational upper bound for the IB loss in Section 3.2. In Section 3.3, we describe the training of the reweighting network and the classifier network in IDR.

## 3.1 Generating Synthetic Training Samples with Diffusion Models

To generate labeled synthetic training samples, we train a conditional Latent Diffusion Model (LDM) (Rombach et al., 2022) with Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) on the latent features of the images in the training set, which are generated by an off-the-shelf pretrained variational autoencoder (VAE) model from Stable Diffusion (Rombach et al., 2022). Detailed formulations of the training and inference of diffusion models, LDM, and CFG are deferred to Section A.1 of the appendix. We use Diffusion Transformers (DiTs) (Peebles & Xie, 2023) as the backbones of the LDMs in our work. Let $v_{\text{e}}$ and $v_{\text{d}}$ be the fixed pretrained encoder and decoder. The encoder of the VAE is first applied to generate the latent features $\{h_i\}_{i=1}^N$ of $\mathcal{D}_{\text{real}}$, where $h_i = v_{\text{e}}(x_i)$ is the latent feature of the $i$-th image. The LDM is then trained on $\{h_i, y_i\}_{i=1}^N$ by minimizing the loss in Equation 10 in Section A.1 of the appendix. Algorithm 1 in Section A.1 of the appendix describes the training algorithm of the LDM. Once the training of the LDM is finished, the latent features $\{\widehat{h}_i\}_{i=1}^M$ are generated for a set of pre-defined synthetic labels $\{\widehat{y}_i\}_{i=1}^M$. The synthetic labels are obtained by uniformly sampling class indices from all the classes following the convention in existing GDA methods (Azizi et al., 2023; Trabucco et al., 2024), which ensures balanced coverage across all classes. The number of synthetic samples is decided by cross-validation as detailed in Section 4.1. The synthetic training images $\{\widehat{x}_i\}_{i=1}^M$ are then generated by applying the pretrained decoder on the latent features $\{\widehat{h}_i\}_{i=1}^M$, where $\widehat{x}_i = v_{\text{d}}(\widehat{h}_i)$. Algorithm 2 in Section A.1 of the appendix describes the generation process of the synthetic training set. After obtaining the synthetic training set $\mathcal{D}_{\text{syn}}$ with the LDM, we combine it with the original training set $\mathcal{D}_{\text{real}}$ to obtain the augmented training set $\mathcal{D}_{\text{aug}}$. Next, the classifier network in IDR can be trained together with the sample reweighting network on $\mathcal{D}_{\text{aug}}$ to be described in Section 3.3.

## 3.2 Variational Upper Bound for the IB Loss

Although data reweighting methods have achieved remarkable success in image classification, existing data reweighting methods do not explicitly ensure that more informative training samples which contribute more to classification tasks receive higher importance weights when applied to GDA. As a result, synthetic training samples with noise introduced by the generative models, such as local visual artifacts (unnatural textures, color bleeding, or over-smoothed regions) (Corvi et al., 2023) or semantic inconsistencies (Azizi et al., 2023; He et al., 2023), could potentially degrade the performance of the classification model trained on the augmented training set. To address this issue, we propose a novel information-theoretic data reweighting method for GDA, termed Informative Data Reweighting (IDR), which encourages the sample reweighting network to assign higher importance weights to more informative training samples by explicitly reducing the IB loss on the augmented training set. Again, we note that our reweighting is applied to both real and synthetic training samples for maximal benefit of our reweighting scheme. IDR aims to reduce the IB loss,

$\text{IB}(\tilde{Z}, \tilde{X}, \tilde{Y}) = I(\tilde{Z}, \tilde{X}) - I(\tilde{Z}, \tilde{Y})$, where $I(\cdot, \cdot)$ stands for the mutual information (MI). $\tilde{X}$, $\tilde{Z}$, and $\tilde{Y}$ denote the random variables representing the input feature, learned feature, and ground truth training class label of the samples in the augmented training set, respectively. The trade-off parameter of this IB loss is discussed in Section G.1 of the appendix. Reduction of the IB loss ensures that the learned features of the augmented training set are more correlated with their class labels and less correlated with their input features. Lacking such an information-theoretic mechanism, existing sample reweighting methods, which do not explicitly learn informative importance weights, may introduce noisy information to classification tasks by assigning high importance weights to noisy training samples. By reducing the IB loss, IDR explicitly encourages more informative training samples in the augmented training set to receive higher importance weights by increasing the correlation of the learned features with class labels while decreasing their correlation with the noise in the training samples, ultimately leading to more discriminative features.

In order to assign higher importance weights to more informative training samples, we propose to train the reweighting network by explicitly reducing the IB loss on the augmented training set. To this end, we first derive a variational upper bound for the IB loss, which can be optimized by standard SGD algorithms. Given the augmented training set $\mathcal{D}_{\text{aug}}$, we first specify how to compute the IB loss, $\text{IB}(\Theta) = I(\tilde{Z}(\Theta), \tilde{X}) - I(\tilde{Z}(\Theta), \tilde{Y})$, where $\Theta$ is the weights of the classification network, $\tilde{X}$ is a random variable representing the input feature which takes values in $\{\tilde{x}_i\}_{i=1}^{N'}$, $\tilde{Z}(\Theta)$ is a random variable representing the learned feature which takes values in $\{\tilde{z}_i(\Theta)\}_{i=1}^{N'}$ with $\tilde{z}_i(\Theta)$ being the learned feature for the $i$-th training sample in $\mathcal{D}_{\text{aug}}$. $\tilde{Y}$ is a random variable representing the class label, which takes values in $\{\tilde{y}_i\}_{i=1}^{N'}$.

We use prototype-based feature distribution modeling in prototypical learning methods (Caron et al., 2020; Li et al., 2021; Yang et al., 2018; Snell et al., 2017) to model the probabilities used in the computation of MI terms. We define $\mathcal{C}(\theta, \Theta) = \left\{c_k^{(\text{feat})}(\theta, \Theta)\right\}_{k=1}^{C}$ as the learnable prototypes of the learned features on $\mathcal{D}_{\text{aug}}$, where $C$ is the number of prototypes and $\theta$ denotes the parameters of the sample reweighting network. The formulas for the computation of $\mathcal{C}(\theta, \Theta)$ can be found in Equation (2). We abbreviate $\tilde{Z}(\Theta)$ as $\tilde{Z}$ and $c_k^{(\text{feat})}(\theta, \Theta)$ as $c_k^{(\text{feat})}$. Then we define the probability that $\tilde{z}_i$ belongs to prototype $a$ as $\Pr\left[\tilde{Z} \in a\right] = \frac{1}{N'} \sum_{i=1}^{N'} \phi(\tilde{z}_i, c_a^{(\text{feat})})$ with $\phi(\tilde{z}_i, c_a^{(\text{feat})}) = \frac{\exp\left(-\left\|\tilde{z}_i - c_a^{(\text{feat})}\right\|_2^2\right)}{\sum_{a'=1}^{C} \exp\left(-\left\|\tilde{z}_i - c_{a'}^{(\text{feat})}\right\|_2^2\right)}$ following (Wang et al., 2024; 2025). Similarly, we define the joint probabilities $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right] = \frac{1}{N'} \sum_{i=1}^{N'} \phi(\tilde{z}_i \in a, \tilde{x}_i \in y)$, where $\phi(\tilde{z}_i \in a, \tilde{x}_i \in y)$ is defined in (13) in the appendix.

We note that the computation of $\Pr\left[\tilde{Z} \in a\right]$ and $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]$ follows prototype-based feature distribution modeling in prototypical learning methods (Caron et al., 2020; Li et al., 2021; Yang et al., 2018; Snell et al., 2017), which do not require a strong discrete clustering assumption on the learned features. The success of the prototype-based methods (Yang et al., 2018; Snell et al., 2017) shows that neural networks can produce latent features that are well separated by the prototypes. The optimization of the information-theoretic objective, VUIB, encourages learning latent features aligned with the prototypes. The number of prototypes is not less than the number of classes in each dataset, which is selected by cross-validation (Caron et al., 2020; Li et al., 2021) from $C'$ to $20C'$ with a step of $C'$, where $C'$ is the number of classes. The distribution of the latent feature space is better captured with more prototypes than classes (Caron et al., 2020; Li et al., 2021).

As a result, we can compute the mutual information by

$$I(\tilde{Z}, \tilde{X}) = \sum_{a=1}^{C} \sum_{y=1}^{C} \Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right] \log \frac{\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]}{\Pr\left[\tilde{Z} \in a\right] \Pr\left[\tilde{X} \in y\right]},$$

$$I(\tilde{Z}, \tilde{Y}) = \sum_{a=1}^{C} \sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{\Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right]}{\Pr\left[\tilde{Z} \in a\right] \Pr\left[\tilde{Y} = y\right]},$$

and then compute the IB loss $\text{IB}(\Theta) = \text{IB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}}) = I(\tilde{Z}, \tilde{X}) - I(\tilde{Z}, \tilde{Y})$. The consistency of the MI estimators is discussed in Section A.2 of the appendix. Given a variational distribution $Q(\tilde{Z} \in a | \tilde{Y} = y)$ for $a \in [C]$ and $y \in [C']$, the following theorem gives our new variational upper bound, $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}})$, for the IB loss $\text{IB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}})$.

**Theorem 3.1.** Let $\Pr\left[\tilde{X} \in y\right] = \sum_{i=1}^{N'} \mathbb{1}_{\{\tilde{y}_i = y\}} / N' := p_y$ be the prior probability for every $y \in [C]$. Then

$$\text{IB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}}) \leq \text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}}), \tag{1}$$

where

$$\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}}) := \frac{1}{N'} \sum_{i=1}^{N'} \text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \tilde{x}_i), \quad \text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \tilde{x}_i) := U_i - V_i.$$

Here

$$U_i := \sum_{a=1}^{C} \sum_{y=1}^{C} \phi(i, a, y) \log \left( \frac{\phi(i, a, y)}{p_y \phi(\tilde{z}_i, c_a^{(\text{feat})})} \right), \quad V_i := \sum_{a=1}^{C} \sum_{y=1}^{C'} \phi(\tilde{z}_i, c_a^{(\text{feat})}) \mathbb{1}_{\{\tilde{y}_i = y\}} \log Q(\tilde{Z} \in a | \tilde{Y} = y),$$

where $\phi(i, a, y) = \phi(\tilde{z}_i \in a, \tilde{x}_i \in y)$. $Q(\tilde{Z} \in a | \tilde{Y} = y)$ is the variational conditional probability that $\tilde{Z}$ belongs to prototype $a$ given its label $\tilde{Y}$ being $y$, which is computed efficiently by Algorithm 5 in the appendix. $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \tilde{x}_i)$ can be interpreted as the information bottleneck upper bound for the $i$-th sample in $\mathcal{D}_{\text{aug}}$. The proof of this theorem follows by applying Lemma E.1 and Lemma E.2 in Section E of the appendix. We remark that $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \tilde{x}_i)$ is ready to be optimized by standard SGD algorithms because it is the summation of losses on individual training points.

**Efficient and Distribution-Free Variational Upper Bound.** Since our novel variational upper bound is distribution-free, we demonstrate the computational efficiency of VUIB over the existing distribution-free upper bound for the IB loss in CLUB (Cheng et al., 2020). Let $T_0$ denote the computational cost of a forward and backward pass of the neural network for each individual training sample. The overall computational complexity for VUIB is $\Theta(N'T_0 + N'C)$. By comparison, computing the upper bound for the IB loss introduced in CLUB (Cheng et al., 2020) incurs a significantly higher computational complexity exceeding $\Theta(N'^2 T_0)$ since $N' \gg C$. Notably, the term $\Theta(N'^2 T_0)$ accounts solely for the computation of the upper bound for the mutual information $I(\tilde{Z}, \tilde{X})$. In addition, CLUB (Cheng et al., 2020) requires the evaluation of a lower bound on the mutual information $I(\tilde{Z}, \tilde{Y})$ in order to construct an upper bound on the IB loss $I(\tilde{Z}, \tilde{X}) - I(\tilde{Z}, \tilde{Y})$. A detailed analysis of the computational complexity of both CLUB and VUIB is provided in Section F of the appendix. Unlike existing upper bounds for the IB loss, such as VIB (Dai et al., 2018; Srivastava et al., 2021), which rely on an impractical Gaussian assumption over the hidden features of DNNs, and APIB (Guo et al., 2023), which only reduces an approximation of the IB loss, IDR directly optimizes a variational upper bound of the IB loss, namely VUIB, without introducing distributional assumptions on the hidden features or approximations to the IB objective. As reported in Table 2 in Section 4.3, VUIB consistently achieves substantially better performance than the competing approaches, and requires much less training time than the distribution-free baseline method, CLUB (Cheng et al., 2020).

### 3.3 Informative Data Reweighting (IDR)

IDR aims to assign higher importance weights to more informative training samples in the augmented training set that contribute more to the image classification task. To this end, IDR trains a sample reweighting network $g_\theta(\cdot)$ to learn importance weights $\{g_\theta(\tilde{x}_i) \in [0, 1]\}_{i=1}^{N'}$ of the samples in the augmented training set by reducing the IB loss on them, where $g_\theta(\cdot)$ is a DNN and $\theta$ denotes its parameters. Following (Shu et al., 2019), we employ a two-layer multilayer perceptron as the sample reweighting network, where a sigmoid activation is applied to the output to produce importance weights. We remark that the reweighting network plays a role similar to that of the meta-networks in (Shu et al., 2019; Jain et al., 2024), which generate the importance weights for training samples. Inspired by existing works (Wang et al., 2025), our reweighting scheme applies to both synthetic and real training samples.

To train the sample reweighting network $g_\theta(\cdot)$ to reduce the IB loss on $\mathcal{D}_{\text{aug}}$, such that more informative samples receive higher importance weights, we optimize $g_\theta(\cdot)$ by reducing the variational upper bound for the IB loss, VUIB. Let $f'_\Theta(\cdot)$ denote the representation learning backbone of the image classifier $f_\Theta(\cdot)$ excluding the last linear layer. To compute the VUIB on the augmented training set $\mathcal{D}_{\text{aug}}$, we first compute the prototype centroids for the input features and the learned features using all the samples in $\mathcal{D}_{\text{aug}}$. The prototypes of the learned features can be computed by

$$c_k^{(\text{feat})}(\theta, \Theta) = \frac{\sum_{i=1}^{N'} g_\theta(\tilde{x}_i) f'_\Theta(\tilde{x}_i) \mathbb{1}_{\{\tilde{x}_i \in k\}}}{\sum_{i=1}^{N'} g_\theta(\tilde{x}_i) \mathbb{1}_{\{\tilde{x}_i \in k\}}}, \tag{2}$$

where $k \in [C]$ is the prototype index and $C$ is the number of prototypes, and $\mathbb{1}_{\{\}}$ is an indicator function. Next, the VUIB on $\mathcal{D}_{\text{aug}}$ can be computed using Equation (1). With the sample reweighting network $g_\theta(\cdot)$, the overall training loss for the classifier $f_\Theta(\cdot)$ on the augmented training set $\mathcal{D}_{\text{aug}}$ is $\mathcal{L}_{\text{train}}(\theta, \Theta, \mathcal{D}_{\text{aug}}) = \frac{1}{N'} \sum_{i=1}^{N'} g_\theta(\tilde{x}_i) \text{CE}\left(f_\Theta(\tilde{x}_i), \tilde{y}_i\right)$, where $\text{CE}(\cdot, \cdot)$ is the cross-entropy function. To train the classifier $f_\Theta(\cdot)$ by optimizing $\mathcal{L}_{\text{train}}(\theta, \Theta, \mathcal{D}_{\text{aug}})$ while training the sample reweighting network $g_\theta$ by optimizing $\text{VUIB}(\theta, \Theta, \mathcal{D}_{\text{aug}})$, we formulate a bi-level optimization objective for IDR as

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}_{\text{train}}(\theta^*, \Theta, \mathcal{D}_{\text{aug}}), \quad \text{s.t. } \theta^* = \arg\min_{\theta} \text{VUIB}(\mathcal{C}(\theta, \Theta^*), \Theta^*, \mathcal{D}_{\text{aug}}), \tag{3}$$

where $\Theta^*$ and $\theta^*$ are the optimal parameters for the classifier $f_\Theta(\cdot)$ and the sample reweighting network $g_\theta(\cdot)$.

**Optimization of IDR.** To train the classifier $f_\Theta(\cdot)$ and the sample reweighting network $g_\theta(\cdot)$ with the optimization objective in Equation (3), we adopt an alternating stochastic gradient descent update strategy commonly used for solving bi-level optimization problems (Shu et al., 2019; Algan & Ulusoy, 2021; Jain et al., 2024). In the bi-level optimization framework, the lower level learns sample importance weights via a reweighting network, while the upper level trains the classifier using these weights to improve generalization. At the $t$-th epoch, the parameters of the sample reweighting network are first updated by $\theta^{(t)} = \theta^{(t-1)} - \eta_\theta \nabla_\theta \text{VUIB}(\mathcal{C}(\theta, \Theta^{(t-1)}), \Theta^{(t-1)}, \mathcal{D}_{\text{aug}})$, where $\eta_\theta$ is the learning rate of $\theta$. $\theta^{(t)}$ and $\Theta^{(t)}$ are the parameters of the sample reweighting network and the classifier network at the $t$-th epoch. Next, the parameters of the classifier are updated by $\Theta^{(t)} = \Theta^{(t-1)} - \eta_\Theta \nabla_\Theta \mathcal{L}_{\text{train}}(\theta^{(t-1)}, \Theta, \mathcal{D}_{\text{aug}})$, where $\eta_\Theta$ is the learning rate of $\Theta$. Since both VUIB and $\mathcal{L}_{\text{train}}$ are separable and amenable to mini-batch stochastic gradient descent (SGD), the entire optimization process of IDR can be efficiently conducted using mini-batch SGD. Algorithm 3 in Section B of the appendix describes the training process of IDR. $Q^{(t)}(\tilde{Z} \in a | \tilde{Y} = y)$ is the variational conditional probability that a feature $\tilde{Z}$ belongs to prototype $a$ given class label $y$ at the $t$-th epoch, which is computed efficiently by Algorithm 5 in the appendix.

## 4 Experiments

We present a comprehensive evaluation of our proposed IDR. The performance of IDR on large-scale image classification is evaluated on ImageNet-1K in Section 4.1. The impact of varying the number of synthetic training samples is investigated in Section 4.2, with matched synthetic data budgets and the computational cost analyzed in Sections D.16 and D.17 of the appendix. A comparison between the proposed variational upper bound for the IB loss and existing IB formulations is presented in Section 4.3. The transferability of pretrained IDR models is evaluated for object detection in Section 4.4. Additional experiment results are presented in Section D of the appendix. Section D.1 compares IDR with recent conventional data augmentation methods, which do not rely on generative models, and shows that IDR is complementary to the conventional data augmentation methods. Section D.2 presents an ablation study demonstrating the advantages of jointly reweighting real and synthetic samples in IDR over reweighting synthetic samples alone. The transferability of IDR models pretrained on ImageNet-1K is further evaluated for semantic segmentation and fine-grained image classification tasks in Section D.3 and Section D.4. The effect of different diffusion models used for synthetic data generation in IDR is examined in Section D.5. Ablation studies and training time analysis of the individual components of IDR, including the VUIB and the reweighting mechanism, are provided in Section D.6. The comparison between IDR and representative active learning methods is reported in Section D.7. The statistical significance of the performance improvements achieved by IDR is analyzed in Section D.8. We provide additional qualitative examples of

Table 1: Performance comparisons between IDR and existing data selection and data reweighting methods on ImageNet-1K. $N$ denotes the size of the original ImageNet-1K training set. $M$ denotes the synthetic data size in multiples of $N$.

| Method | ViT-S/16 | | ViT-B/16 | | Swin-T | | Swin-B | |
|---|---|---|---|---|---|---|---|---|
| | $M$ ($\times N$) | Top-1 | $M$ ($\times N$) | Top-1 | $M$ ($\times N$) | Top-1 | $M$ ($\times N$) | Top-1 |
| Base Model (w/o GDA) | - | 81.2 | - | 83.7 | - | 81.3 | - | 83.5 |
| GDA w/o Reweighting | 2 | 81.5 | 2 | 83.9 | 2 | 81.7 | 2 | 84.0 |
| MW-Net (Shu et al., 2019) | 2 | 81.8 | 2 | 84.1 | 2 | 81.9 | 2 | 83.9 |
| OTR (Guo et al., 2022) | 3 | 81.6 | 2 | 84.0 | 2 | 81.6 | 2 | 84.0 |
| IE (Chhabra et al., 2024) | 2 | 81.7 | 3 | 84.0 | 2 | 81.9 | 2 | 83.9 |
| CBF (He et al., 2023) | 2 | 81.9 | 2 | 84.4 | 2 | 82.2 | 2 | 84.4 |
| REVAR (Jain et al., 2024) | 3 | 81.9 | 2 | 84.2 | 2 | 82.0 | 3 | 84.3 |
| IDS (Wang et al., 2025) | 3 | 82.1 | 3 | 84.4 | 3 | 82.2 | 3 | 84.5 |
| GenDataAgent (Li et al., 2025) | 3 | 82.1 | 3 | 84.5 | 3 | 82.3 | 3 | 84.6 |
| **IDR (Ours)** | 4 | **83.2** $\pm$ 0.18 | 5 | **85.6** $\pm$ 0.22 | 4 | **83.4** $\pm$ 0.27 | 5 | **85.7** $\pm$ 0.19 |

synthetic images with high or low importance weights in Section D.9. We evaluate the robustness of the IDR models pretrained on ImageNet-1K and tested on ImageNet-A and ImageNet-R in Section D.10. We study the distribution of the importance weights learned by IDR in Section D.11. We evaluate the performance of IDR with different generative models in Section D.12. We study the performance of IDR for medical image classification in Section D.13. We study the performance of IDR models trained from scratch in Section D.14. A qualitative analysis of the data reweighting with IDR using t-SNE visualization is conducted in Section D.15.

### 4.1 Image Classification on ImageNet-1K

**Implementation Details.** We evaluate IDR for image classification on ImageNet-1K (Deng et al., 2009). ViT-S, ViT-B (Dosovitskiy et al., 2021), Swin-T, and Swin-B (Liu et al., 2021) are used as the base classification networks. All models are fine-tuned on the augmented training set from checkpoints pretrained on the original ImageNet-1K training set. Detailed settings for the fine-tuning process are presented in Section C of the appendix. DiT-B (Peebles & Xie, 2023) is employed for synthetic data generation, with detailed settings deferred to Section C of the appendix. The same set of synthetic images generated by DiT-B on ImageNet-1K is used for IDR and all baseline models in this section. The number of synthetic images for IDR and baselines is selected via five-fold cross-validation from $\{N, 2N, \ldots, 10N\}$, where $N$ is the size of the ImageNet-1K training set.

**Results** We compare the performance of competing data selection and data reweighting methods, including IE (Chhabra et al., 2024), CBF (He et al., 2023), MW-Net (Shu et al., 2019), OTR (Guo et al., 2022), REVAR (Jain et al., 2024), IDS (Wang et al., 2025), and GenDataAgent (Li et al., 2025), with our IDR for GDA on ImageNet-1K (Deng et al., 2009). Similar to IDR, all baseline methods reweight both the real and synthetic samples in the augmented training set. The number of synthetic samples added to the augmented training set for the baseline methods is determined via cross-validation, following the same setting as IDR. It is observed in Table 1 that IDR consistently outperforms all competing methods across different backbones. For instance, IDR achieves a top-1 accuracy of $85.7\%$, outperforming the strongest baseline, GenDataAgent, by $1.1\%$, when using Swin-B as the classification network, demonstrating the effectiveness of the IB-based data reweighting method on the augmented training set. We also compare IDR with the baseline that performs GDA without reweighting the training samples. As shown in Table 1, IDR significantly outperforms the GDA w/o Reweighting. For instance, IDR outperforms GDA w/o Reweighting by $1.7\%$ in accuracy with the backbone, Swin-B.

### 4.2 Impact of Synthetic Sample Quantity

We conduct an ablation study by progressively increasing the number of synthetic samples in the augmented training set, ranging from $N$ to $10N$, where $N$ denotes the size of the original training set. ViT-B is used as the classification network. It can be observed from Figure 3 that although a moderate amount of synthetic samples benefits the classification accuracy, the performance of all the competing baselines and IDR eventually decreases with excessive synthetic samples due to the inherent noise in the synthetic samples (Azizi et al., 2023; Trabucco et al., 2024; Na et al., 2024). However, our IDR still exhibits a

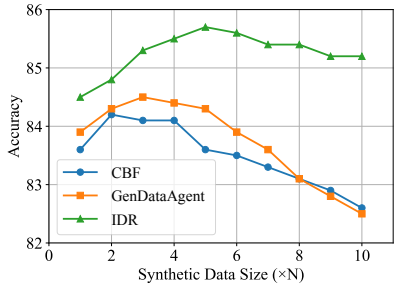



Figure 3: Performance comparison on ImageNet-1K with different amounts of synthetic training data.

clear advantage over competing baselines. As illustrated in Figure 3, both CBF and GenDataAgent achieve their best performance with a relatively small amount of synthetic samples. In particular, CBF and GenDataAgent perform best with $2N$ and $3N$ synthetic samples, respectively. Both CBF and GenDataAgent exhibit a significant decrease in top-1 accuracy once the number of synthetic samples exceeds $3N$, indicating that noise in synthetic samples significantly degrades the classification network. In a strong contrast, IDR is more robust to noise in the generated synthetic samples compared with the baselines, and benefits from a larger amount of synthetic samples. In particular, the performance of IDR continues to improve until the number of synthetic samples reaches $5N$. Importantly, the top-1 accuracy of IDR with excessive synthetic samples ($> 5N$) still significantly surpasses the best accuracy of all baselines, with only a marginal decrease from the peak performance at $5N$. It can also be observed from Figure 3 that IDR consistently outperforms CBF and GenDataAgent when the same amount of synthetic samples is used, ranging from $N$ to $10N$.

### 4.3 Comparison between VUIB and Existing Upper Bounds for the IB Loss

We compare the proposed variational upper bound for the IB loss, VUIB, with existing works deriving the upper bound for the IB loss (Cheng et al., 2020; Dai et al., 2018; Srivastava et al., 2021). To compare the performance of VUIB with VIB, APIB, and CLUB, we conduct an ablation study by replacing VUIB with VIB, APIB, and CLUB in IDR for image classification on ImageNet-1K. The training time is evaluated on four NVIDIA A100 GPUs using the augmented training set of ImageNet-1K with $5N$ synthetic samples, where $N$ is the size of the original training set. It is observed in Table 2 that the unrealistic Gaussian distribution assumption on the hidden features imposed by VIB (Dai et al., 2018) and the IB approximation in APIB (Guo et al., 2023) lead to degraded performance compared with our reduction of VUIB. For instance, the model based on VUIB outperforms the model based on APIB by $1.2\%$ in top-1 classification accuracy. In addition, the model based on VUIB outperforms the model based on CLUB by $1.0\%$ in accuracy while using only $49.7\%$ of the training time of the CLUB-based model.

Table 2: Comparison of different methods for the training overhead incurred by IB bound. ViT-B is used as the base model.

| Methods | Top-1 | Training Time (Hours / Epoch) |
|---|---|---|
| VIB (Dai et al., 2018) | 84.2 | 1.69 |
| APIB (Guo et al., 2023) | 84.4 | 1.70 |
| CLUB (Cheng et al., 2020) | 84.6 | 3.47 |
| **VUIB (Ours)** | **85.6** | 1.72 |

### 4.4 Object Detection

To evaluate the transferability of the features learned by IDR, we assess models pretrained by IDR on ImageNet-1K for the object detection task. We incorporate the ImageNet pre-trained models based on Swin-B in Table 1 into the Cascade Mask R-CNN framework (Cai & Vasconcelos, 2021) for object detection on the MS-COCO dataset (Lin et al., 2014). We follow the Swin training setup (Liu et al., 2021) with standard image resizing, AdamW optimization with learning rate $1 \times 10^{-4}$, weight decay $0.05$, batch size $16$, and a $3\times$ schedule ($36$ epochs), and report COCO box and mask mAP metrics following (Cai & Vasconcelos, 2021). It is observed in Table 3 that IDR consistently outperforms all competing methods across $\text{mAP}^{\text{box}}$ and $\text{mAP}^{\text{m}}$. For instance, IDR outperforms the strongest baseline by $1.0\%$ in $\text{mAP}^{m}$, demonstrating better transferability of learned features to dense prediction tasks. The relation to task-specific GDA methods for dense prediction is discussed in Section G.2 of the appendix.

Table 3: Object Detection Results on COCO.

| Methods | $\text{mAP}^{box}$ | $\text{mAP}^{m}$ |
|---|---|---|
| MW-Net (Shu et al., 2019) | 52.0 | 45.2 |
| OTR (Guo et al., 2022) | 52.4 | 45.4 |
| IE (Chhabra et al., 2024) | 52.1 | 45.3 |
| CBF (He et al., 2023) | 52.5 | 45.5 |
| REVAR (Jain et al., 2024) | 52.5 | 45.6 |
| IDS (Wang et al., 2025) | 52.7 | 45.6 |
| GenDataAgent (Li et al., 2025) | 52.8 | 45.8 |
| **IDR (Ours)** | **53.6** | **46.8** |

## 5 Conclusion

In this paper, we propose Informative Data Reweighting (IDR), a novel method designed to reweight samples in the augmented training set for Generative Data Augmentation (GDA) based on an information-theoretic measure, the Information Bottleneck (IB). IDR trains a sample reweighting network to reduce the IB loss on the augmented training set, such that the IB principle, learning features more correlated with the outputs and less correlated with the inputs, is better adhered to. Extensive experiments and ablation studies demonstrate that IDR successfully assigns higher weights to more informative samples in the augmented training set for image classification, and significantly outperforms existing data selection and data reweighting methods for GDA.

### AI USE STATEMENT

In this work, we used generative AI tools for interpreting results. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, and generating synthetic data sets, implementing methods, cleaning and reformatting dataset, assisting with translation, supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for improving readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

The code of IDR is available at https://anonymous.4open.science/r/IDR-3BE0/. The generation of the synthetic training set is described in Section 3.1 and Section A of the appendix, the variational upper bound VUIB and the training of IDR are described in Section 3.2, Section 3.3, and Algorithm 3 in Section B of the appendix, and the complete proofs are provided in Section E of the appendix. The main experimental settings, including the hyperparameters of the classifiers, the sample reweighting network, and the diffusion model, are provided in Section C of the appendix, and the settings of the additional experiments are provided in the corresponding subsections of Section D of the appendix.

## REFERENCES

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10358–10368. IEEE, 2022.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Görkem Algan and Ilkay Ulusoy. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7142–7148. IEEE, 2021.

Omar Ali, Wiem Abdelbaki, Anup Shrestha, Ersin Elbasi, Mohammad Abdallah Ali Alryalat, and Yogesh K Dwivedi. A systematic literature review of artificial intelligence in the healthcare sector: Benefits, challenges, methodologies, and functionalities. *Journal of Innovation & Knowledge*, 8 (1):100333, 2023.

Rana Ali Amjad and Bernhard C. Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42 (9):2225–2239, 2020.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves ImageNet classification. *Trans. Mach. Learn. Res.*, 2023.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv preprint arXiv:2302.02503*, 2023.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1483–1498, 2021.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119, pp. 1779–1788. PMLR, 2020.

Anshuman Chhabra, Peizhao Li, Prasant Mohapatra, and Hongfu Liu. "what data benefits my classifier?" enhancing model performance and interpretability through influence-based data selection. In *The Twelfth International Conference on Learning Representations*, 2024.

Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pp. 973–982. IEEE, 2023.

Bin Dai, Chen Zhu, Baining Guo, and David P. Wipf. Compressing neural networks using the variational information bottleneck. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, pp. 1143–1152. PMLR, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Uwe Einmahl and David M Mason. Uniform in bandwidth consistency of kernel-type function estimators. *Annals of statistics*, 33(3):1380–1403, 2005.

Laila El Jiani, Sanaa El Filali, et al. Overcome medical image data scarcity by data augmentation techniques: A review. In *2022 International Conference on Microelectronics (ICM)*, pp. 21–24. IEEE, 2022.

Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):5, 2021.

Haoyang Fang, Boran Han, Shuai Zhang, Su Zhou, Cuixiong Hu, and Wen-Ming Ye. Data augmentation for object detection via controllable diffusion models. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024.

Ruibin Feng, Zongwei Zhou, Michael B Gotway, and Jianming Liang. Parts2Whole: Self-supervised contrastive learning via reconstruction. In *MICCAI Workshop on Domain Adaptation and Representation Transfer*, pp. 85–95. Springer, 2020.

Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pp. 510–526. Springer, 2020.

Larry Goldstein and Karen Messer. Optimal plug-in estimators for nonparametric functional estimation. *The annals of statistics*, pp. 1306–1328, 1992.

Dandan Guo, Zhuo Li, He Zhao, Mingyuan Zhou, Hongyuan Zha, et al. Learning to re-weight examples with optimal transport for imbalanced classification. *Advances in Neural Information Processing Systems*, 35:25517–25530, 2022.

Song Guo, Lei Zhang, Xiawu Zheng, Yan Wang, Yuchao Li, Fei Chao, Chenglin Wu, Shengchuan Zhang, and Rongrong Ji. Automatic network pruning via Hilbert-Schmidt independence criterion lasso under information bottleneck principle. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023*, pp. 17412–17423. IEEE, 2023.

Yeho Gwon, Sehyun Hwang, Hoyoung Kim, Jungseul Ok, and Suha Kwak. Enhancing cost efficiency in active learning with candidate set query. *Trans. Mach. Learn. Res.*, 2025.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip H. S. Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Nishant Jain, Karthikeyan Shanmugam, and Pradeep Shenoy. Learning model uncertainty as variance-minimizing instance weights. In *The Twelfth International Conference on Learning Representations*, 2024.

Minsoo Kang, Seong-Whan Lee, and Suhyun Kim. Model-agnostic and efficient mixup augmentation guided by saliency maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(8):6946–6958, 2025.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16049–16096. PMLR, 2023.

Bum Jun Kim and Sang Woo Kim. Configuring data augmentations to reduce variance shift in positional embedding of vision transformers. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 17840–17849. AAAI Press, 2025.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (BiT): General visual representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pp. 491–507. Springer, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pp. 554–561. IEEE Computer Society, 2013.

Dan Kushnir and Luca Venturi. Diffusion-based sampling for deep active learning. In *2023 International Conference on Sampling Theory and Applications (SampTA)*, pp. 1–9, 2023. doi: 10.1109/SampTA59647.2023.10301392.

Qiuxia Lai, Yu Li, Ailing Zeng, Minhao Liu, Hanqiu Sun, and Qiang Xu. Information bottleneck approach to spatial attention learning. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 779–785. ijcai.org, 2021.

Shiye Lei, Hao Chen, Sen Zhang, Bo Zhao, and Dacheng Tao. Image captions are natural prompts for text-to-image models. *arXiv preprint arXiv:2307.08526*, 2023.

Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. BigDatasetGAN: Synthesizing ImageNet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21330–21340, 2022.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Zhiteng Li, Lele Chen, Jerone T. A. Andrews, Yunhao Ba, Yulun Zhang, and Alice Xiang. GenDataAgent: On-the-fly dataset augmentation with synthetic data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. Explore the power of synthetic data on few-shot object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pp. 638–647. IEEE, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*. Springer, 2014.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[MatchMask authors – TO BE VERIFIED]. MatchMask: Mask-centric generative data augmentation for label-scarce semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Sangwoo Mo, Chiheon Kim, Sungwoong Kim, Minsu Cho, and Jinwoo Shin. Mining GOLD samples for conditional GANs. *Advances in Neural Information Processing Systems*, 32, 2019.

Byeonghu Na, Yeongmin Kim, HeeSun Bae, Jung Hyun Lee, Se Jung Kwon, Wanmo Kang, and Il chul Moon. Label-noise robust diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pp. 722–729. IEEE Computer Society, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pp. 3498–3505. IEEE Computer Society, 2012.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

David Pollard. A Central Limit Theorem for $k$-Means Clustering. *The Annals of Probability*, 10(4): 919 – 926, 1982.

W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25 (1):37–43, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Kama Ramudu, V Murali Mohan, D Jyothirmai, DVSSSV Prasad, Ruchi Agrawal, and Sampath Boopathi. Machine learning and artificial intelligence in disease prediction: Applications, challenges, limitations, case studies, and future directions. In *Contemporary Applications of Data Fusion for Advanced Healthcare Informatics*, pp. 297–318. IGI Global, 2023.

Philippe Rigollet and Régis Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, pp. 1154–1178, 2009.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022.

Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic ImageNet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8011–8021, June 2023.

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In Munkhtsetseg Nandigjav, Niloy J. Mitra, and Aaron Hertzmann (eds.), *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference, Vancouver, BC, Canada, August 7 - 11, 2022*, pp. 49:1–49:10. ACM, 2022.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-Weight-Net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5972–5981, 2019.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4077–4087, 2017.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. Neural Networks Learn. Syst.*, 34(11): 8135–8153, 2023.

Ayush Srivastava, Oshin Dutta, Jigyasa Gupta, Sumeet Agarwal, and Prathosh AP. A variational information bottleneck based method to compress sequential networks for human action recognition. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pp. 2744–2753. IEEE, 2021.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. StableRep: Synthetic images from text-to-image models make strong visual representation learners. *arXiv preprint arXiv:2306.00984*, 2023.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*, 2023.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Yancheng Wang, Ping Li, and Yingzhen Yang. Visual transformer with differentiable channel selection: An information bottleneck inspired approach. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Yancheng Wang, Rajeev Goel, Marko Jojic, Alvin C. Silva, Teresa Wu, and Yingzhen Yang. Informative synthetic data generation for thorax disease classification. In *Conference on Uncertainty in Artificial Intelligence, Rio Othon Palace, Rio de Janeiro, Brazil, 21-25 July 2025*, volume 286 of *Proceedings of Machine Learning Research*, pp. 4489–4514. PMLR, 2025.

Junfei Xiao, Yutong Bai, Alan Yuille, and Zongwei Zhou. Delving into masked autoencoders for multi-label thorax disease classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3588–3600, 2023.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, volume 11209 of *Lecture Notes in Computer Science*, pp. 432–448. Springer, 2018.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3474–3482. Computer Vision Foundation / IEEE Computer Society, 2018.

Jianan Yang, Haobo Wang, Sai Wu, Gang Chen, and Junbo Zhao. Towards controlled data augmentations for active learning. In *International Conference on Machine Learning*, pp. 39524–39542. PMLR, 2023.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 93–102, 2019.

Jianhao Yuan, Francesco Pinto, Adam Davies, Aarushi Gupta, and Philip Torr. Not just pretty pictures: Text-to-image generators enable interpretable interventions for robust representations. *arXiv preprint arXiv:2212.11237*, 2022.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.*, 127(3):302–321, 2019.

Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pp. 27378–27394. PMLR, 2022.

Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv preprint arXiv:2305.15316*, 2023.

# A Information on Diffusion Models

## A.1 Formulations of Diffusion Models

**Diffusion models (DMs)** are probabilistic latent-variable frameworks that represent data $x^0$ through a Markovian transformation from a highly noisy variable $x_T$ back to the original signal, while preserving identical dimensionality across all intermediate states. They are characterized by two coupled Markov chains: a forward noising procedure given by $q(x^{(1:T)} \mid x^0) = \prod_{t=1}^{T} q(x^{(t)} \mid x^{(t-1)})$ and a learnable reverse-time denoising chain formulated as $p_\omega(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\omega(x^{(t-1)} \mid x^{(t)})$. In the forward direction, Gaussian perturbations are incrementally injected into the data:

$$q(x^{(t)} \mid x^{(t-1)}) = \mathcal{N}(x^{(t)}; \sqrt{1-\beta^{(t)}}x^{(t-1)}, \beta^{(t)}\mathbf{I}), \tag{4}$$

where the noise schedule $\beta^{(1:T)}$ controls the variance added at each diffusion step $t$. This schedule is selected such that the terminal distribution $x_T$ closely follows a standard normal distribution, i.e., $q(x_T) \approx \mathcal{N}(0, \mathbf{I})$. Once specified, the forward diffusion process $q$ remains fixed and is not learned.

Sample synthesis in DMs relies on training a parametric reverse process that progressively removes noise from $x_{T:1}$ to recover the clean sample $x^0$:

$$p_\omega(x^{(t-1)} \mid x^{(t)}) = \mathcal{N}(x^{(t-1)}; \mu_\omega(x^{(t)}, t), (\rho^{(t)})^2\mathbf{I}), \tag{5}$$

with the prior $p(x_T)$ defined as $\mathcal{N}(0, \mathbf{I})$. In practice, neural architectures such as U-Nets or Transformers are employed to model the mean function $\mu_\omega$, while the variance $\rho^{(t)}$ is typically fixed in advance.

For learning, the forward chain $q(x^{(1:T)} \mid x^0)$ is treated as a known posterior, and the parameters of the reverse process $p_\omega(x_{0:T})$ are optimized to maximize a variational lower bound on the data likelihood. Since directly optimizing the likelihood objective often causes numerical instability, a commonly adopted and more stable surrogate loss is:

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{x^0, \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\mathbf{I}),t}\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\omega(x^{(t)}, t)\right\|_2^2, \tag{6}$$

where $\boldsymbol{\varepsilon}_\omega$ denotes a neural predictor of the injected noise $\boldsymbol{\varepsilon}$, enabling the model to infer $x^{(t-1)}$ from $x^{(t)}$ at each timestep. After training, new samples are obtained via iterative ancestral sampling:

$$x^{(t-1)} = \frac{1}{\sqrt{1-\beta^{(t)}}}(x^{(t)} - \frac{\beta^{(t)}}{\sqrt{1-(\alpha^{(t)})^2}}\boldsymbol{\varepsilon}_\omega(x^{(t)}, t)) + \rho^{(t)}\boldsymbol{\varepsilon}, \tag{7}$$

initialized from the Gaussian prior $x_T \sim p(x_T) = \mathcal{N}(x_T; 0, \mathbf{I})$, where $\alpha^{(t)} \coloneqq \prod_{s=1}^{t}\sqrt{1-\beta^{(s)}}$, i.e., $(\alpha^{(t)})^2 = \prod_{s=1}^{t}(1-\beta^{(s)})$.

**Latent Diffusion Models (LDMs)** extend conventional diffusion models by performing the diffusion process in a compressed latent space, thereby significantly reducing computational cost. Specifically, the original data $x^0$ is first mapped into a latent representation $h^0$ using an encoder. Diffusion in the latent domain follows:

$$q(h^{(t)} \mid h^{(t-1)}) = \mathcal{N}(h^{(t)}; \sqrt{1-\beta^{(t)}}h^{(t-1)}, \beta^{(t)}I), \tag{8}$$

while the corresponding reverse-time dynamics aim to reconstruct the clean latent code $h^0$ from $h_T$:

$$p_\omega(h^{(t-1)} \mid h^{(t)}) = \mathcal{N}(h^{(t-1)}; \mu_\omega(h^{(t)}, t), (\rho^{(t)})^2 I), \tag{9}$$

after which $h^0$ is decoded back into the original data space. The optimization objective for LDMs is defined as

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{v_e(x),\epsilon\sim\mathcal{N}(0,I),t}\left\|\epsilon - \epsilon_\omega(h^{(t)}, t, y)\right\|_2^2, \tag{10}$$

which mirrors the noise-prediction formulation used in standard diffusion models.

**Classifier-Free Guidance (CFG)** enhances conditional generation by combining conditional and unconditional noise estimators during sampling, allowing explicit control over class-conditional

behavior without an auxiliary classifier. When applied to LDMs, the guided sampling update is expressed as:

$$h^{(t-1)} = \frac{1}{\sqrt{1-\beta^{(t)}}}(h^{(t)} - \frac{\beta^{(t)}}{\sqrt{1-(\alpha^{(t)})^2}}\tilde{\varepsilon}^{(t)}) + \rho^{(t)}\varepsilon, \tag{11}$$

where $\tilde{\varepsilon}^{(t)} = (1+\gamma)\varepsilon_\omega(h^{(t)}, y, t) - \gamma\varepsilon_\omega(h^{(t)}, t)$, and $\gamma$ controls the strength of guidance, steering the sampling trajectory toward desired conditions.

Algorithm 1 outlines the training procedure for the LDM, while Algorithm 2 details the generation process used to construct the synthetic training set.

**Algorithm 1** Training Procedure for the LDM

**Input:** The original training dataset $\mathcal{D}_{\text{real}} = \{x_i, y_i\}_{i=1}^N$, the encoder $v_{\text{e}}$ from a fixed pretrained VAE, and the total number of LDM training epochs $t_{\text{LDM}}$.
**Output:** The learned parameters $\omega$ of the LDM.
1: Initialize the LDM parameters $\omega$.
2: Map the input samples $\{x_i\}_{i=1}^N$ into latent representations $\{h_i\}_{i=1}^N$ using the encoder $v_{\text{e}}$, i.e., $h_i = v_{\text{e}}(x_i)$.
3: **for** $\tau = 1, 2, \ldots, t_{\text{LDM}}$ **do**
4: Optimize $\omega$ via mini-batch stochastic gradient descent over $\{h_i\}_{i=1}^N$ by minimizing the loss $\mathcal{L}_{\text{LDM}}$ defined in Equation (10).
5: **end for**
6: **return** The optimized LDM parameters $\omega$.

**Algorithm 2** Synthetic Training Set Generation

**Input:** The target labels of the synthetic dataset $\{\widehat{y}_i\}_{i=1}^M$, the trained LDM parameters $\omega$, and the decoder $v_{\text{d}}$ from the fixed pretrained VAE.
**Output:** The generated synthetic dataset $\mathcal{D}_{\text{syn}} = \{\widehat{x}_i, \widehat{y}_i\}_{i=1}^M$.
1: **for** $i = 1, 2, \ldots, M$ **do**
2: Draw a noise vector $\epsilon \sim \mathcal{N}(0, I)$.
3: Produce a synthetic latent representation $\widehat{h}_i$ from $\epsilon$ using the LDM according to Equation (9) in Section A of the appendix.
4: Recover the synthetic input feature by decoding the latent code, $\widehat{x}_i = v_{\text{d}}(\widehat{h}_i)$.
5: **end for**
6: **return** The synthetic training dataset $\mathcal{D}_{\text{syn}} = \{\widehat{x}_i, \widehat{y}_i\}_{i=1}^M$.

### A.2 Consistency of the Mutual Information Estimators

We now discuss the accuracy (consistency) of the MI estimators in Section 3.2. The consistency of such estimators depends on the consistency of the estimator $\Pr\left[\tilde{Z} \in a\right]$ for the probability that the feature $\tilde{Z}$ belongs to prototype $a$, that is, $\tilde{Z} \in a$, and the consistency of the estimator $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]$ for the ground truth joint probability that $\tilde{Z} \in a, \tilde{X} \in y$. We now analyze the consistency of these two probability estimators.

The estimator $\Pr\left[\tilde{Z} \in a\right] = \frac{1}{N'}\sum_{i=1}^{N'} \phi(\tilde{z}_i, c_a^{(\text{feat})})$ is a consistent plugin estimator. Recall that $\mathcal{C}(\theta, \Theta) = \left\{c_k^{(\text{feat})}(\theta, \Theta)\right\}_{k=1}^C$ are the set of $C$ learnable prototypes, which is abbreviated as $\mathcal{C}$ here without confusion. According to the theory of asymptotic quantization in (Pollard, 1982), for a fixed $C$, the prototypes $\mathcal{C}$ optimized via empirical risk minimization converge almost surely to the optimal prototypes $\mathcal{C}^*$ that minimize the expected quantization error:

$$\lim_{N'\to\infty} \min_{\mathcal{C}} \frac{1}{N'}\sum_{i=1}^{N'} \left\|\tilde{z}_i - c_k^{(\text{feat})}\right\|_2^2 = \min_{\mathcal{C}} \mathbb{E}\left[\left\|\tilde{z}_i - c_k^{(\text{feat})}\right\|_2^2\right]. \tag{12}$$

We note that each $c_k^{(\text{feat})}$ is computed as the weighted average of the features in prototype $k \in [C]$, so $\mathcal{C} = \left\{c_k^{(\text{feat})}\right\}_{k=1}^{C}$ would minimize the LHS of (12), guaranteeing the consistency of $\mathcal{C}$.

We note that the asymptotic result (12) can be converted to a finite-sample result presented as follows. We define

$$R_{N'}(\mathcal{C}) \coloneqq \frac{1}{N'} \sum_{i=1}^{N'} \|\tilde{z}_i - c\|_2^2, \qquad R(\mathcal{C}) \coloneqq \mathbb{E}\left[\left\|\tilde{Z} - c\right\|_2^2\right],$$

and let

$$\widehat{\mathcal{C}}_{N'} \in \arg\min_{|\mathcal{C}|=C} R_{N'}(\mathcal{C}), \qquad \mathcal{C}^\star \in \arg\min_{|\mathcal{C}|=C} R(\mathcal{C}).$$

Then, under standard regularity conditions, for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$R(\widehat{\mathcal{C}}_{N'}) - R(\mathcal{C}^\star) \le \varepsilon_{N',\delta},$$

where $\varepsilon_{N',\delta} = O\left(\sqrt{\frac{1}{N'} \log \frac{1}{\delta}}\right)$.

Furthermore, under the softmax formulation, $\phi(\tilde{z}_i, c_a^{(\text{feat})})$ acts as a continuous relaxation of the hard assignment. The consistency of the estimator $\Pr\left[\tilde{Z} \in a\right]$ then follows from the standard kernel density estimation literature (Einmahl & Mason, 2005; Goldstein & Messer, 1992; Rigollet & Vert, 2009), which establishes consistency of the estimator $\Pr\left[\tilde{Z} \in a\right]$ together with the consistency result in Eq. (12) above.

Similarly, following a similar argument, with the following estimator for the probability $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right] = \frac{1}{N'} \sum_{i=1}^{N'} \phi(\tilde{z}_i \in a, \tilde{x}_i \in y)$, where

$$\phi(\tilde{z}_i \in a, \tilde{x}_i \in y) = \frac{\exp(-\left\|\tilde{z}_i - c_a^{(\text{feat})}\right\|_2^2 - \left\|\tilde{x}_i - c_y^{(\text{input})}\right\|_2^2)}{\sum_{a'=1}^{C}\sum_{y'=1}^{C} \exp(-\left\|\tilde{z}_i - c_{a'}^{(\text{feat})}\right\|_2^2 - \left\|\tilde{x}_i - c_{y'}^{(\text{input})}\right\|_2^2)},$$

which are also specified in (15) and (13) of this appendix, it can be shown that $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]$ converges to the ground truth joint probability that $\tilde{Z} \in a, \tilde{X} \in y$. The consistency of the two estimators, $\Pr\left[\tilde{Z} \in a\right]$ and $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]$, guarantee the consistency of the estimators for the two MI terms, $I(\tilde{Z}, \tilde{X})$ and $I(\tilde{Z}, \tilde{Y})$.

## B Training Algorithm of IDR

Algorithm 3 describes the training process of IDR. $Q^{(t)}(\tilde{Z} \in a | \tilde{Y} = y)$ is the variational conditional probability that a feature $\tilde{Z}$ belongs to prototype $a$ given class label $y$ at the $t$-th epoch, which is computed efficiently by Algorithm 5. We note that the optimization of IDR uses first-order alternating updates: $\theta$ is updated with $\nabla_\theta$VUIB while $\Theta$ is fixed, and $\Theta$ is updated with $\nabla_\Theta \mathcal{L}_{\text{train}}$ while $\theta$ is fixed. Therefore, the dependence of $\theta^*$ on $\Theta$ in Equation (3) is not differentiated through, and no implicit hypergradient is computed.

**Algorithm 3** Algorithm of IDR

**Input:** The augmented training set $\mathcal{D}_{\text{aug}}$, epoch number $t_{\max}$, the learning rates $\eta_\theta$ and $\eta_\Theta$.

1: Initialize the classifier network parameters $\Theta^{(0)}$ and the sample reweighting network parameters $\theta^{(0)}$.
2: **for** $t = 1, 2, \ldots, t_{\max}$ **do**
3: Compute the prototypes of the learned features $\mathcal{C}(\theta^{(t-1)}, \Theta^{(t-1)})$ by Equation (2).
4: Update $\theta^{(t)}$ by applying mini-batch gradient descent on $\mathcal{D}_{\text{aug}}$ using $\theta^{(t)} = \theta^{(t-1)} - \eta_\theta \nabla_\theta \text{VUIB}(\mathcal{C}(\theta, \Theta^{(t-1)}), \Theta^{(t-1)}, \mathcal{D}_{\text{aug}})$.
5: Update $\Theta^{(t)}$ by applying mini-batch gradient descent on $\mathcal{D}_{\text{aug}}$ using $\Theta^{(t)} = \Theta^{(t-1)} - \eta_\Theta \nabla_\Theta \mathcal{L}_{\text{train}}(\theta^{(t-1)}, \Theta, \mathcal{D}_{\text{aug}})$.
6: Compute $Q^{(t)}(\tilde{Z} \in a | \tilde{Y} = y)$ by Algorithm 5 in Section F of the appendix.
7: **end for**
8: **return** The trained weights $\Theta$ of the classifier network and the trained weights $\theta$ of the sample reweighting network.

# C Detailed Experiment Settings

We evaluate IDR for general image classification on ImageNet-1K (Deng et al., 2009), which is a large-scale general image classification benchmark containing approximately $1.28$ million training images and $50,000$ validation images across $1,000$ classes. We fine-tune ViT-S, ViT-B (Dosovitskiy et al., 2021), Swin-T, and Swin-B (Liu et al., 2021) on the augmented ImageNet-1K training set, starting from checkpoints pretrained on the original ImageNet-1K training set. The fine-tuning is performed for $100$ epochs using the AdamW optimizer, with an initial learning rate of $1 \times 10^{-4}$, weight decay of $1 \times 10^{-4}$, and momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$. We adopt a cosine learning-rate decay schedule with a five-epoch linear warm-up. All experiments are conducted on four NVIDIA A100 GPUs with a total batch size of $1024$, corresponding to $256$ images per GPU. Input images are resized to $224 \times 224$, and standard ImageNet data augmentation is applied, including random resized cropping, horizontal flipping, and color jittering. Label smoothing with a factor of $0.1$ is used during training. We apply the Exponential Moving Average (EMA) to model parameters with a decay rate of $0.9999$ for evaluation. The parameters of the sample reweighting network are optimized by the AdamW optimizer with the learning rate fixed to $1 \times 10^{-5}$. The hidden dimension of the sample reweighting network is set to $512$.

The training of IDR starts with a warmup process for 5 epochs to minimize the CE loss of classifier training first, and then alternately optimizes the CE loss and the IB bound, VUIB, as in Algorithm 3 of Section B. Such a warmup strategy, also used in (Guo et al., 2022), effectively prevents the trivial solution that easy samples receive high importance weights.

For synthetic data generation, we adopt the DiT-B backbone (Peebles & Xie, 2023). The diffusion model is trained on the latent representations encoded by the pretrained VAE from Stable Diffusion using classifier-free guidance, optimized with a mean-squared error noise prediction objective for $2{,}800$ epochs using AdamW with a learning rate of $1 \times 10^{-4}$ following (Peebles & Xie, 2023). During inference, synthetic samples are generated with a guidance scale of $4.0$ and $128$ sampling steps, and the resulting latent features are decoded back to the image space using the fixed VAE decoder. The number of synthetic images added to the augmented training set for IDR and the competing baseline methods is selected by five-fold cross-validation from $\{N, 2N, ..., 10N\}$, where $N$ denotes the number of training samples in the original training set of ImageNet-1K.

# D Additional Experiment Results

## D.1 Comparison with Conventional Data Augmentation Methods

Recent works have proposed better conventional data augmentation strategies to improve image classification performance (Kim & Kim, 2025; Kang et al., 2025) without using generative models. In particular, Position-Aware Augmentation (PAA) (Kim & Kim, 2025) proposes to adapt existing data augmentation methods, such as random resize crop, Mixup, CutMix, and random erasing, to explicitly account for the sensitivity of Vision Transformer positional embeddings to spatial transformations. In parallel, GuidedMixup (Kang et al., 2025) proposes saliency-guided mixup by optimizing sample pairings and pixel-wise mixing ratios to preserve semantically salient regions and reduce la-

bel noise during data mixing. These conventional data augmentation strategies do not rely on GDA and are orthogonal to IDR, as they operate directly on the original training samples without selection or reweighting. As shown in Table 4, IDR outperforms both PAA and GuidedMixup, demonstrating the effectiveness of GDA with the proposed informative data reweighting. For instance, IDR outperforms PAA by $1.1\%$ in top-1 classification accuracy on ImageNet-1K. Moreover, combining IDR with either PAA or GuidedMixup further improves performance. Specifically, IDR+PAA achieves a $0.5\%$ improvement over IDR alone and a $1.6\%$ improvement over PAA alone, demonstrating that IDR is complementary to conventional data augmentation strategies and can be seamlessly integrated with them to yield additive performance gains.

Table 4: Comparison with conventional data augmentation methods and their combinations with IDR on ImageNet-1K using ViT-B.

| Method | Top-1 (%) |
|---|---|
| ViT-B | 83.7 |
| PAA (Kim & Kim, 2025) | 84.5 |
| GuidedMixup (Kang et al., 2025) | 84.3 |
| **IDR** | 85.6 |
| **IDR+PAA** | 86.1 |
| **IDR+GuidedMixup** | 86.0 |

### D.2 Ablation Study on Reweighting Real and Synthetic Samples

We conduct an ablation study to examine the effect of jointly reweighting real and synthetic samples within the proposed IDR framework, in comparison to reweighting only synthetic training samples. To this end, we build an ablation model, IDR (Synthetic Only), in which the reweighting network $g_\theta(\cdot)$ is applied exclusively to reweight synthetic samples. All experiments are conducted on ImageNet-1K using ViT-B as the classification backbone under identical training configurations. The number of synthetic samples is determined via cross-validation. All baseline methods compared in this study reweight both the real and synthetic training samples. As shown in Table 5, reweighting synthetic samples alone already yields a substantial performance improvement over the baseline methods, highlighting the effectiveness of importance weighting in mitigating noise introduced by synthetic samples. Nevertheless, jointly reweighting both real and synthetic samples results in additional performance gains. For instance, IDR (Synthetic Only) outperforms the best baseline method, GenDataAgent, by $0.7\%$ in top-1 accuracy, while the IDR model further improves performance by $0.4\%$, highlighting the benefits of jointly reweighting real and synthetic samples.

Table 5: Ablation study on reweighting real and synthetic samples on ImageNet-1K using ViT-B.

| Method | Synthetic Data Size ($\times N$) | Top-1 |
|---|---|---|
| ViT-B/16 | - | 83.7 |
| MW-Net (Shu et al., 2019) | 2 | 84.1 |
| OTR (Guo et al., 2022) | 2 | 84.0 |
| IE (Chhabra et al., 2024) | 3 | 84.0 |
| CBF (He et al., 2023) | 2 | 84.4 |
| REVAR (Jain et al., 2024) | 2 | 84.2 |
| IDS (Wang et al., 2025) | 3 | 84.4 |
| GenDataAgent (Li et al., 2025) | 3 | 84.5 |
| **IDR (Synthetic Only)** | 5 | <u>85.2</u> |
| **IDR** | 5 | **85.6** |

### D.3 Semantic Segmentation

In this section, we evaluate the performance of IDR models pre-trained on ImageNet for segmentation on the ADE20K (Zhou et al., 2019) using the UperNet (Xiao et al., 2018) segmentation framework following (Liu et al., 2021). We follow the training and evaluation protocol in (Liu et al.,

2021), where both our model and the baselines are trained on the training split and evaluated on the validation split of the dataset. All models are optimized using AdamW for a total of 160000 iterations with a batch size of 16, an initial learning rate of $6 \times 10^{-5}$, and a weight decay of 0.01. The learning rate follows a linear decay schedule after a warm-up phase of 1500 iterations. To enhance generalization, we employ data augmentation techniques including random horizontal flipping, random rescaling with a ratio range of $[0.5, 2.0]$, and random photometric distortions. Stochastic depth regularization is applied with a drop rate of 0.2. For inference, we use multi-scale testing with scale factors varying from 0.5 to 1.75. It is observed in Table 6 that IDR consistently outperforms all competing methods. For instance, IDR outperforms the strongest baseline by 0.7% in mIoU, demonstrating stronger representation transferability for segmentation.

Table 6: Semantic Segmentation Results on ADE20K.

| Methods | mIoU |
|---|---|
| MW-Net (Shu et al., 2019) | 51.6 |
| OTR (Guo et al., 2022) | 51.9 |
| IE (Chhabra et al., 2024) | 51.6 |
| CBF (He et al., 2023) | 52.1 |
| REVAR (Jain et al., 2024) | 52.2 |
| IDS (Wang et al., 2025) | 52.1 |
| GenDataAgent (Li et al., 2025) | 52.3 |
| **IDR (Ours)** | 53.0 |

### D.4 Transfer Learning Capability of Models Pretrained by IDR

We evaluate the transfer learning capability of IDR-pretrained Swin-B models on three standard fine-grained image classification benchmarks: Oxford Flowers-102 (Nilsback & Zisserman, 2008), Oxford-IIIT Pet (Parkhi et al., 2012), and Stanford Cars (Krause et al., 2013). Following established transfer learning protocols (Kolesnikov et al., 2020), the IDR-pretrained Swin-B model is fine-tuned on the training split of each downstream dataset and evaluated on the corresponding test split. Swin-B is also used as the feature backbone for all compared baseline methods. All models are initialized with ImageNet-pretrained weights and fine-tuned for 50 epochs using the Adam optimizer. The learning rate is set to $1 \times 10^{-5}$ for all datasets. The top-1 classification accuracy on each dataset is reported in Table 7. The results indicate that IDR-pretrained Swin-B models maintain strong transfer learning performance across diverse fine-grained classification tasks. For instance, on the Stanford Cars dataset, IDR achieves a top-1 accuracy of 93.9%, outperforming the strongest baseline by 1.4%.

Table 7: Top-1 classification accuracy comparison for transfer learning on the Oxford Flowers-102, Oxford-IIIT Pet, and Stanford Cars datasets.

| Methods | Flowers | Pet | Cars |
|---|---|---|---|
| MW-Net (Shu et al., 2019) | 97.0 | 95.3 | 91.6 |
| OTR (Guo et al., 2022) | 97.1 | 95.5 | 91.8 |
| IE (Chhabra et al., 2024) | 97.2 | 95.6 | 92.0 |
| CBF (He et al., 2023) | 97.3 | 95.8 | 92.1 |
| REVAR (Jain et al., 2024) | 97.6 | 96.1 | 92.5 |
| IDS (Wang et al., 2025) | 97.5 | 96.0 | 92.3 |
| GenDataAgent (Li et al., 2025) | 97.6 | 96.0 | 92.4 |
| **IDR (Ours)** | **98.4** | **97.2** | **93.9** |

### D.5 Ablation Study on the Diffusion Models for the Data Generation in IDR

To evaluate the impact of the diffusion model used for the data generation in IDR, we compare the performance of IDR using three different diffusion models for GDA, which are DiT-B, DiT-L, and DiT-XL (Peebles & Xie, 2023). ViT-B is used as the classification network for all the models in this ablation study. The data generation time and the classification accuracy on ImageNet-1K

are shown in Table 8. It is observed that the performance of the IDR model is not sensitive to the selection of the diffusion models used for data generation. The IDR model based on the largest DiT model DiT-XL only outperforms the IDR based on the smallest DiT model DiT-B by $0.1\%$ in top-1 classification accuracy on ImageNet-1K, demonstrating that the performance of IDR mainly stems from our informative reweighting instead of the diffusion model. In addition, the results in Table 8 show that the synthetic data generation process with the diffusion models in IDR is efficient, with less than $0.2$ seconds/image.

Table 8: Performance comparison between IDR models employing different diffusion models for data generation. The data generation time is evaluated on one NVIDIA A100 GPU.

| Methods | Accuracy | Generation Time (seconds/image) |
|---|---|---|
| ViT-B | 83.7 | - |
| IDR (DiT-B) | <u>85.6</u> | 0.095 |
| IDR (DiT-L) | **85.7** | 0.151 |
| IDR (DiT-XL) | **85.7** | 0.176 |

### D.6 Ablation Study and Training Time Analysis of the IDR

To evaluate the effectiveness and efficiency of different components in IDR, we compare the classification performance and the training time of IDR with two ablation models, which are IDR without VUIB and IDR without the reweighting network. The ablation model, IDR without VUIB, optimizes the sample reweighting network by minimizing cross-entropy loss on a set of clean metadata selected from the ImageNet training set, following (Shu et al., 2019). The ablation model, IDR without reweighting, optimizes the classification network by incorporating the VUIB as a regularization term in the training objective without reweighting the real and synthetic samples in the augmented training set. All experiments are conducted using ViT-B as the classification backbone, and training time is measured on four NVIDIA A100 GPUs. We report the training time in seconds per batch with the same batch size in Table 9 because different GDA methods in Table 1 have augmented training sets of different sizes. It is observed in Table 9 that IDR outperforms the strongest baseline, GenDataAgent (Li et al., 2025), by $1.1\%$ in top-1 accuracy while requiring only a marginal $2.5\%$ increase in training time per batch. Since the training time per epoch scales with the size of the augmented training set, which is $N + 5N$ for IDR and $N + 3N$ for GenDataAgent, the estimated training time per epoch of IDR is $53.7\%$ longer than that of GenDataAgent. The estimated cost of synthetic data generation and fine-tuning is analyzed in Section D.17. In addition, it is observed in Table 9 that removing either VUIB or the reweighting network leads to a clear degradation in performance. For reference, IDR w/o Reweighting, IDR (Synthetic Only) in Table 5, and IDR achieve top-1 accuracies of $85.0\%$, $85.2\%$, and $85.6\%$, respectively. For instance, removing VUIB decreases the top-1 accuracy of IDR by $1.3\%$, which demonstrates the critical role of VUIB in learning robust and discriminative representations from the augmented training set. Computing the prototypes and the distances between the learned features of all samples in the augmented training set and their corresponding prototypes requires only $6.1$ seconds in each epoch. As a result, it is not necessary to project the input features into a lower-dimensional space using methods such as VAEs (Kingma & Welling, 2014), which require additional training time.

### D.7 Comparison with Active Learning Methods

Active learning (AL) methods aim to minimize the effort required for labeling training data by strategically choosing the most informative instances for annotation (Sinha et al., 2019; Yoo & Kweon, 2019; Gao et al., 2020; Kushnir & Venturi, 2023; Yang et al., 2023; Chhabra et al., 2024). The selection of the data for annotation by active learning methods is usually achieved by identifying the most informative data points. Such a process works similarly to the data reweighting process in IDR for identifying the most informative synthetic samples. To show the advantage of IDR over active learning methods in selecting the most informative synthetic samples, we compare IDR with state-of-the-art active learning methods, including CAMPAL (Yang et al., 2023), SAAL (Chhabra et al., 2024), and CSQ (Gwon et al., 2025). The active learning methods are adopted to select training data in each epoch from the augmented training set. Specifically, for a fair comparison, all active

Table 9: Ablation study of IDR with training time analysis. The training time is evaluated on four NVIDIA A100 GPUs. The setup of all the methods in this table is the same as that in Table 1.

| Methods | Top-1 | Training Time (Seconds / Batch) |
|---|---|---|
| ViT-B | 83.7 | 0.739 |
| MW-Net (Shu et al., 2019) | 84.1 | 0.801 |
| OTR (Guo et al., 2022) | 84.0 | 0.772 |
| IE (Chhabra et al., 2024) | 84.0 | 0.739 |
| CBF (He et al., 2023) | 84.4 | 0.714 |
| REVAR (Jain et al., 2024) | 84.2 | 0.825 |
| IDS (Wang et al., 2025) | 84.4 | 0.825 |
| GenDataAgent (Li et al., 2025) | 84.5 | 0.805 |
| IDR w/o VUIB | 84.3 | 0.753 |
| IDR w/o Reweighting | 85.0 | 0.801 |
| **IDR (Ours)** | **85.6** | **0.825** |

learning baselines use the same per-epoch selection budget, where the top $50\%$ samples ranked by their corresponding acquisition scores are selected from the augmented training set for model training. The results are shown in Table 10. It is observed that IDR significantly outperforms the competing active learning methods. For instance, IDR outperforms the best active learning method, CSQ, by $1.2\%$ in top-1 classification accuracy on ImageNet-1K, demonstrating the superiority of IDR in selecting informative training samples compared to active learning methods.

Table 10: Comparison between IDR and active learning methods.

| Methods | Top-1 |
|---|---|
| ViT-B | 83.7 |
| CAMPAL (Yang et al., 2023) | 84.1 |
| SAAL (Chhabra et al., 2024) | 84.2 |
| CSQ (Gwon et al., 2025) | 84.4 |
| **IDR (Ours)** | **85.6** |

### D.8 Improvement Significance Analysis.

To verify that the improvement of our proposed IDR on existing methods is statistically significant and outside the range of random error, we train both IDR and the best baseline methods on different datasets for 10 times with different seeds for random initialization of the networks. Next, we perform the t-test between the results of IDR and the results of the best baseline methods with different backbones to assess if the improvement of IDR is statistically significant. The mean and standard deviation of the results and the p-values of the t-test are shown in Table 11. It is observed that the largest p-value is $5.4 \times 10^{-6}$, which is less than $0.05$. The t-test results suggest that the improvement of IDR over the baseline methods is statistically significant with $p \ll 0.05$, and it is not caused by random error.

Table 11: P-values of t-test between IDR and the baseline methods with the best performance.

| Backbone | Best Baseline | IDR | p-value |
|---|---|---|---|
| ViT-S/16 | 82.1±0.25 | 83.2±0.18 | $2.6 \times 10^{-6}$ |
| ViT-B/16 | 84.5±0.21 | 85.6±0.22 | $5.4 \times 10^{-6}$ |
| Swin-T | 82.3±0.29 | 83.4±0.27 | $1.3 \times 10^{-9}$ |
| Swin-B | 84.6±0.27 | 85.7±0.19 | $7.2 \times 10^{-8}$ |

### D.9 Additional Qualitative Examples of Learned Importance Weights

To further illustrate the behavior of IDR, we provide additional qualitative examples of synthetic images together with the Grad-CAM visualizations for their ground-truth classes in Figure 4. Similar

to Figure 2, these examples show that the importance weights learned by IDR are highly consistent with the semantic fidelity and visual quality of the generated samples. In the first row of examples, the synthetic images receive very low importance weights, ranging from $0.04$ to $0.10$, indicating that IDR identifies them as less informative training samples. These low-weight examples exhibit either semantic inconsistencies or severe visual artifacts. For instance, the synthetic image labeled as steam locomotive in Figure 4 (a) is visually dominated by an amorphous dark structure that does not clearly resemble the target object, and its Grad-CAM map does not concentrate on a well-defined locomotive region. The rose hip example in Figure 4 (b) contains strong corruption and texture distortion. As a result, the Grad-CAM response is scattered across the branches rather than focusing on the rose hip. The low-weight examples for Platypus and Indigo Bunting in Figure 4 (c)-(d) show a similar pattern, where the generated content only weakly reflects the target semantics and the corresponding Grad-CAM maps fail to highlight clear object-centric regions. These observations suggest that IDR effectively suppresses synthetic samples with visual artifacts or semantic inconsistency that may mislead the classifier during training. In contrast, the second row of Figure 4 presents synthetic images with high learned importance weights, ranging from $0.91$ to $0.96$. These samples exhibit substantially better visual fidelity and clearer object semantics. The Steam Locomotive example in Figure 4 (e) shows a recognizable train structure, and the Grad-CAM map highlights the main body of the locomotive rather than irrelevant background regions. Similarly, the high-weight Rose Hip image in Figure 4 (f) clearly depicts the fruit structure, and the corresponding Grad-CAM activation is concentrated around the target objects. For Platypus, the synthetic image in Figure 4 (g) provides a much more plausible depiction of the animal in water, with Grad-CAM focusing on the central object region. The Indigo Bunting example in Figure 4 (h) contains a clearly visible bird, and its Grad-CAM map is tightly localized on the bird body.

### D.10 Robustness Evaluation on ImageNet-A and ImageNet-R

We further evaluate the robustness of the models pretrained on ImageNet-1K by testing them on two out-of-distribution benchmarks, namely ImageNet-A and ImageNet-R. ImageNet-A contains naturally adversarial examples that are challenging for standard image classifiers, while ImageNet-R consists of renditions of natural images with substantial appearance shifts. These two benchmarks are widely used to assess the robustness and generalization ability of pretrained image classification models under distribution shifts. It is observed in Table 12 that IDR consistently outperforms all competing methods on both ImageNet-A and ImageNet-R across the two backbones. For instance, with Swin-T as the backbone, IDR achieves $26.4\%$ top-1 accuracy on ImageNet-A, outperforming the strongest baseline, CBF, by $2.6\%$, which demonstrates that IDR improves not only the in-distribution classification performance on ImageNet-1K, but also the robustness of the learned representations. Such improvement can be attributed to the information bottleneck objective in IDR, which reduces the mutual information between the learned representations and noisy input features by reducing the IB loss through its variational upper bound, VUIB. As a result, the learned representations become less correlated with nuisance perturbations and noise in the inputs, leading to improved robustness and stronger generalization.

### D.11 Histogram Analysis of Importance Weights

To further analyze the behavior of IDR, we visualize the histograms of the learned importance weights for real and synthetic images, as well as the evolution of the weight distribution across training epochs. Figure 5 compares the weight distributions of real and synthetic samples. It can be observed that the distribution for real images is relatively concentrated, with most samples receiving moderate to high importance weights, indicating that real training samples are generally informative and consistent in quality. In contrast, the distribution for synthetic images is substantially broader, reflecting the coexistence of highly informative synthetic samples and low-quality ones affected by artifacts or semantic inconsistencies. In particular, the histogram indicates that IDR assigns low weights ($< 0.2$) to $11.6\%$ of the synthetic samples, further confirming that a nontrivial portion of the generated data is identified as less informative. This result is consistent with the motivation of IDR, which aims to suppress noisy synthetic samples while assigning larger weights to semantically reliable ones.

Figure 6 further illustrates the histograms of importance weights at different training stages. At epoch 10, the weight distribution is relatively concentrated around the middle range, suggesting that the model has only limited ability to distinguish informative samples from less informative ones

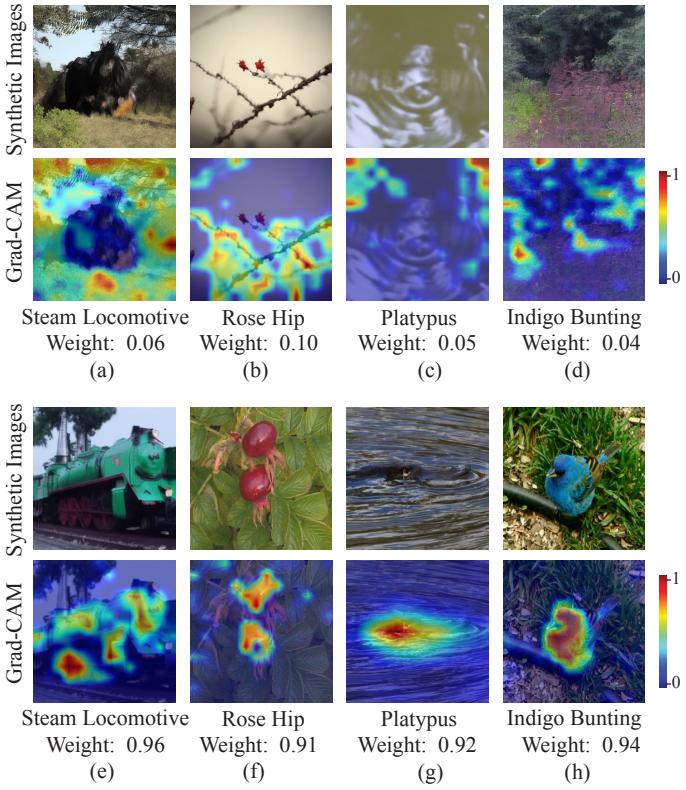
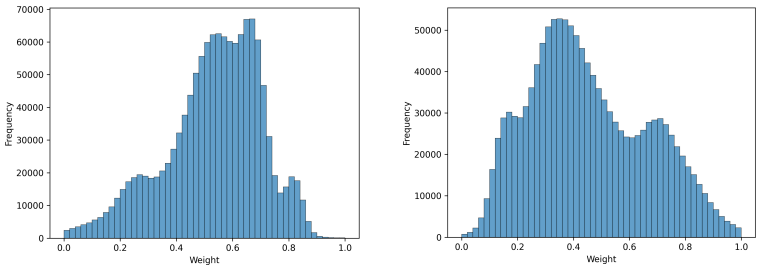



Figure 4: Additional qualitative examples of synthetic images and the corresponding Grad-CAM visualizations for the ground-truth classes. The top two rows show synthetic samples assigned low importance weights by IDR, where the generated images exhibit semantic ambiguity, visual artifacts, or weak object evidence, and the Grad-CAM maps do not focus on clear object-centric regions. The bottom two rows show synthetic samples assigned high importance weights, where the generated images are visually more faithful to the target classes and the Grad-CAM maps concentrate on the corresponding foreground objects. These examples further demonstrate that the learned importance weights in IDR are well aligned with the semantic quality and discriminative usefulness of synthetic training samples.

Table 12: Robustness evaluation of models pretrained on ImageNet-1K and tested on ImageNet-A and ImageNet-R.

| Method | Backbone | ImageNet-1K | ImageNet-A | ImageNet-R |
|---|---|---|---|---|
| Swin-T | Swin-T | 81.3 | 21.6 | 41.3 |
| MW-Net (Shu et al., 2019) | | 81.9 | 23.0 | 41.8 |
| OTR (Guo et al., 2022) | | 81.6 | 23.7 | 43.2 |
| IE (Chhabra et al., 2024) | | 81.9 | 22.3 | 41.8 |
| CBF (He et al., 2023) | | 82.2 | 23.8 | 42.8 |
| REVAR (Jain et al., 2024) | | 82.0 | 21.4 | 42.1 |
| IDS (Wang et al., 2025) | | 82.2 | 23.0 | 42.6 |
| GenDataAgent (Li et al., 2025) | | 82.3 | 23.2 | 42.2 |
| **IDR (Ours)** | | **83.4** | **26.4** | **45.9** |
| Swin-B | Swin-B | 83.5 | 35.8 | 64.2 |
| MW-Net (Shu et al., 2019) | | 83.9 | 36.8 | 64.2 |
| OTR (Guo et al., 2022) | | 84.0 | 36.5 | 63.8 |
| IE (Chhabra et al., 2024) | | 83.9 | 36.2 | 64.7 |
| CBF (He et al., 2023) | | 84.4 | 37.4 | 66.3 |
| REVAR (Jain et al., 2024) | | 84.3 | 35.9 | 65.3 |
| IDS (Wang et al., 2025) | | 84.5 | 37.0 | 65.8 |
| GenDataAgent (Li et al., 2025) | | 84.6 | 36.8 | 66.3 |
| **IDR (Ours)** | | **85.7** | **39.2** | **67.8** |



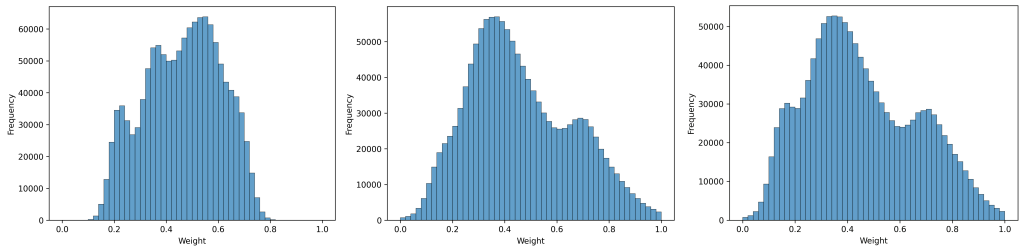

(a) Real Image (b) Synthetic Image

Figure 5: Histograms of importance weights for real and synthetic images. The distribution for real images is relatively concentrated, with most samples receiving moderate to high importance weights, indicating consistent sample quality. In contrast, the distribution for synthetic images is substantially broader, reflecting the coexistence of highly informative samples and low-quality ones with artifacts or semantic inconsistencies. Notably, $11.6\%$ of the synthetic samples receive weights smaller than $0.2$, showing that IDR effectively identifies a nontrivial set of less informative generated samples.

at the early stage of training. By epoch 50, the distribution becomes noticeably broader, indicating increased variance in the learned weights and an emerging separation between low- and high-importance samples. At epoch 100, the distribution spreads more widely across the range $[0, 1]$, with more samples occupying both low- and high-weight regions. This progressive broadening suggests that IDR gradually improves its ability to discriminate between informative and uninformative samples as training proceeds, resulting in a more selective and effective reweighting scheme at later stages.

### D.12 Evaluation with Different Generative Models

To further examine the robustness of IDR to the choice of generative model, we conduct additional experiments using synthetic data generated by StyleGAN-XL (Sauer et al., 2022) and a UNet-based DDPM (Ho et al., 2020) with classifier guidance, both of which are weaker than the DiT-based latent diffusion model adopted in the main experiments. As shown in Table 13, IDR consistently achieves the best performance across both settings when Swin-B is used as the classification backbone. For example, IDR achieves $85.5\%$ top-1 accuracy with DDPM-generated synthetic data, outperforming




(a) Epoch 10 (b) Epoch 50 (c) Epoch 100

Figure 6: Histograms of importance weights at different training stages. At epoch 10, the distribution is relatively concentrated around mid-range values, indicating limited differentiation among samples. At epoch 50, the distribution becomes broader, reflecting increased variance and the emergence of separation between low- and high-importance samples. At epoch 100, the distribution spreads more widely across $[0, 1]$, suggesting that IDR gradually enhances its ability to distinguish informative samples from less informative ones, leading to a more discriminative reweighting scheme at later training stages.

the strongest baseline, GenDataAgent, by $1.1\%$, and achieves $85.4\%$ with StyleGAN-XL-generated synthetic data, outperforming the best competing baseline, IDS, by $1.2\%$. Such results indicate that the effectiveness of IDR does not depend on a specific generator architecture or on highly optimized synthetic data quality. Instead, by reducing the IB loss through VUIB, IDR can consistently emphasize more informative synthetic samples while suppressing less reliable ones, thereby yielding strong and stable performance even when the synthetic data are produced by relatively weaker generative models.

Table 13: Performance comparison using synthetic data generated by different generative models. Swin-B is used as the classification backbone.

| Method | DDPM (Ho et al., 2020) | StyleGAN-XL (Sauer et al., 2022) |
|---|---|---|
| Swin-B | 83.5 | 83.5 |
| GDA w/o Reweighting | 83.6 | 83.5 |
| CBF (He et al., 2023) | 83.8 | 84.0 |
| REVAR (Jain et al., 2024) | 83.9 | 83.9 |
| IDS (Wang et al., 2025) | 84.1 | 84.2 |
| GenDataAgent (Li et al., 2025) | 84.4 | 84.1 |
| **IDR (Ours)** | **85.5** | **85.4** |

### D.13 Evaluation on Medical Image Classification

We further evaluate IDR on two medical image classification benchmarks, NIH ChestX-ray14 (Wang et al., 2017) and CheXpert (Irvin et al., 2019), to examine its effectiveness beyond natural image classification. These datasets are widely used for evaluating medical image understanding models and present additional challenges due to subtle disease patterns and the high visual similarity across categories. For reproducibility, we use ViT-B as the classification backbone and resize all images to $224 \times 224$. Since NIH ChestX-ray14 and CheXpert contain grayscale chest radiographs, each image is converted to a three-channel grayscale image before being fed into the classifier and the latent diffusion model. The classifier is trained for $100$ epochs using AdamW with an initial learning rate of $1 \times 10^{-4}$, weight decay of $1 \times 10^{-4}$, $(\beta_1, \beta_2) = (0.9, 0.999)$, cosine learning-rate decay, a $5$-epoch linear warm-up, batch size $1024$, label smoothing $0.1$, and EMA decay $0.9999$. The sample reweighting network in IDR is trained with AdamW using a learning rate of $1 \times 10^{-5}$ and hidden dimension $512$. For synthetic medical image generation, we train a dataset-specific DiT-B latent diffusion model on the training split of each medical dataset. The diffusion model operates on latent representations encoded by the pretrained Stable-Diffusion VAE, and the generated latents are decoded by the fixed VAE decoder. The DiT-B generator is trained for $2800$ epochs with AdamW using a learning rate of $1 \times 10^{-4}$. During sampling, we use classifier-free guidance with guidance scale $4.0$ and $128$ sampling steps. The number of synthetic images added to the original training set is selected by five-fold cross-validation from $\{N, 2N, \ldots, 10N\}$, where $N$ denotes the number of original training samples. It is observed in Table 14 that IDR consistently

outperforms all competing methods on both NIH ChestX-ray14 and CheXpert. For example, IDR outperforms the strongest baseline, GenDataAgent, by $1.2\%$ in mAUC on CheXpert, demonstrating that the benefits of IDR generalize beyond natural image recognition.

Table 14: Performance comparison on medical image classification datasets.

| Method | NIH ChestX-ray14 (mAUC %) | CheXpert (mAUC %) |
|---|---|---|
| CBF (He et al., 2023) | 82.5 | 89.2 |
| REVAR (Jain et al., 2024) | 82.5 | 89.3 |
| GenDataAgent (Li et al., 2025) | 82.7 | 89.5 |
| **IDR (Ours)** | **83.9** | **90.7** |

### D.14 Training from Scratch on ImageNet-1K

To further evaluate the effectiveness of IDR without relying on pretrained initialization, we conduct additional experiments by training Swin-T from scratch on ImageNet-1K and compare IDR with competing data selection and data reweighting methods under the same training protocol. All competing methods and IDR are trained using the same experimental settings as in (Liu et al., 2021). As shown in Table 15, IDR consistently achieves the best performance even in the more challenging from-scratch training setting. In particular, IDR attains a top-1 accuracy of $83.2\%$, outperforming the strongest baseline, GenDataAgent, by $1.2\%$. These results demonstrate that the advantage of IDR is not limited to fine-tuning from pretrained models, but also extends to end-to-end training from scratch. This further suggests that the proposed IB-based reweighting mechanism can effectively guide the model toward more informative training samples throughout training, thereby improving representation learning and classification performance even in the absence of pretrained features.

Table 15: Performance comparison of training from scratch on ImageNet-1K using Swin-T as the classification backbone.

| Method | Top-1 |
|---|---|
| Swin-T | 81.3 |
| MW-Net (Shu et al., 2019) | 81.5 |
| OTR (Guo et al., 2022) | 81.4 |
| IE (Chhabra et al., 2024) | 81.5 |
| CBF (He et al., 2023) | 81.6 |
| REVAR (Jain et al., 2024) | 81.8 |
| IDS (Wang et al., 2025) | 81.7 |
| GenDataAgent (Li et al., 2025) | 82.0 |
| **IDR (Ours)** | **83.2** |

### D.15 t-SNE Visualization Analysis

To qualitatively examine how IDR reweights synthetic samples in the augmented training set, we illustrate the importance weights of sampled synthetic images from four visually similar dog classes in ImageNet-1K using t-SNE in Figure 7, where marker shapes indicate classes and colors denote importance weights. Equation (2) shows that the prototypes are largely decided by samples with high importance weights. This is reflected by the t-SNE visualization, which shows that synthetic images located closer to the prototypes tend to receive higher importance weights. In the meantime, we can still observe that samples far away from corresponding prototypes and closer to class boundaries can still receive relatively high importance weights, such as those in the green boxes, demonstrating the capability of our IDR framework in identifying complex class decision boundaries. To qualitatively analyze how IDR reweights real training samples, we visualize the importance weights of sampled real images from four visually similar dog classes in ImageNet-1K using t-SNE, as illustrated in Figure 8. The observed distribution is similar to that for synthetic samples.

### D.16 Comparison under Matched Synthetic Data Budgets

In Table 1, the synthetic data budget selected for IDR ($4N$ or $5N$) is larger than those selected for the baselines ($2N$ or $3N$). IDR and all baselines use the same set of synthetic images generated by DiT-B, and the budget of each method is selected by the same five-fold cross-validation from

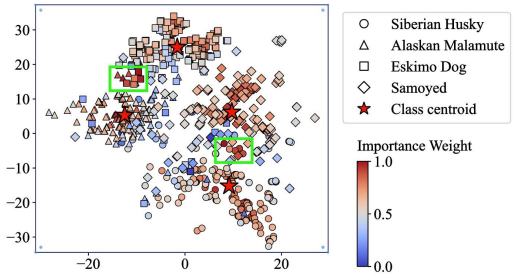



Figure 7: Visualization of the importance weights learned by IDR on synthetic images. For each class, 500 synthetic images are randomly sampled from the augmented training set and projected into a two-dimensional space using t-SNE. Different marker shapes denote different classes, corresponding to four visually similar dog categories from ImageNet-1K, including Siberian Husky, Alaskan Malamute, Eskimo Dog, and Samoyed, which are challenging to distinguish. The color of each point indicates its importance weight, with redder colors indicating higher importance and bluer colors indicating lower importance.

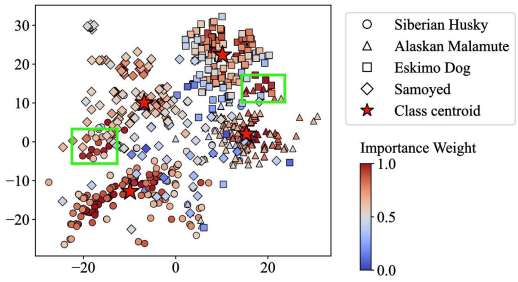



Figure 8: Visualization of the importance weights learned by IDR on real images. For each class, 500 real images are randomly sampled from the augmented training set and projected into a two-dimensional space using t-SNE. Different marker shapes denote different classes, corresponding to four visually similar dog categories from ImageNet-1K, including Siberian Husky, Alaskan Malamute, Eskimo Dog, and Samoyed, which are challenging to distinguish. The color of each point indicates its importance weight, with redder colors indicating higher importance and bluer colors indicating lower importance.

$\{N, 2N, \ldots, 10N\}$ (Section 4.1). To compare the methods under matched budgets, Table 16 lists the top-1 accuracy of CBF, GenDataAgent, and IDR with ViT-B for each budget from $N$ to $10N$.

Table 16: Top-1 accuracy (%) of CBF, GenDataAgent, and IDR on ImageNet-1K with ViT-B under matched synthetic data budgets. The values are those plotted in Figure 3. $\Delta$ denotes the margin of IDR over the stronger of CBF and GenDataAgent at each budget.

| Method | Synthetic Data Size ($\times N$) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| CBF (He et al., 2023) | 83.6 | 84.2 | 84.1 | 84.1 | 83.6 | 83.5 | 83.3 | 83.1 | 82.9 | 82.6 |
| GenDataAgent (Li et al., 2025) | 83.9 | 84.3 | 84.5 | 84.4 | 84.3 | 83.9 | 83.6 | 83.1 | 82.8 | 82.5 |
| **IDR (Ours)** | **84.5** | **84.8** | **85.3** | **85.5** | **85.7** | **85.6** | **85.4** | **85.4** | **85.2** | **85.2** |
| $\Delta$ | +0.6 | +0.5 | +0.8 | +1.1 | +1.4 | +1.7 | +1.8 | +2.3 | +2.3 | +2.6 |

It is observed in Table 16 that IDR outperforms both CBF and GenDataAgent at every budget. The margin of IDR over the stronger baseline is smaller at small budgets, ranging from $0.5\%$ to $0.8\%$ from $N$ to $3N$, and increases to $1.4\%$ at $5N$ and $2.6\%$ at $10N$. The larger margins result from both the improvement of IDR up to $5N$ and the degradation of the baselines with excessive synthetic samples. From their peaks at $2N$ and $3N$ to $10N$, the top-1 accuracy of CBF and GenDataAgent decreases by $1.6\%$ and $2.0\%$, respectively, whereas the accuracy of IDR at $10N$ is $0.5\%$ lower than its peak at $5N$.

Moreover, with ViT-B, IDR does not require a larger synthetic data budget than the baselines to outperform them. With $2N$ synthetic samples, IDR achieves $84.8\%$, which is $0.3\%$ higher than the best ViT-B baseline in Table 1 at its own cross-validated budget, GenDataAgent with $84.5\%$ at $3N$. At $3N$, IDR outperforms GenDataAgent by $0.8\%$, which is smaller than the margin of $1.1\%$ in Table 1, where IDR uses $5N$. Part of the improvement of IDR in Table 1 is therefore obtained with its larger budget. This part is not explained by the larger budget alone, since CBF and GenDataAgent achieve lower accuracy with $4N$ or $5N$ than with $2N$ and $3N$, respectively, which is consistent with the robustness of IDR to noisy synthetic samples discussed in Section 4.2.

**Synthetic data budgets in object detection.** The object detection experiments in Section 4.4 incorporate the Swin-B models trained by each method on ImageNet-1K in Table 1 into the Cascade Mask R-CNN framework. The synthetic data budgets of these models are those listed for Swin-B in Table 1, namely $5N$ for IDR, $3N$ for GenDataAgent, IDS, and REVAR, and $2N$ for CBF, IE, OTR, and MW-Net.

### D.17 Computational Cost of the GDA Pipeline and Scalability

Since the GDA methods in Table 1 use augmented training sets of different sizes, Table 17 estimates the cost of synthetic data generation and fine-tuning, i.e., the GDA pipeline after the LDM is trained, for IDR, GenDataAgent (Li et al., 2025), and CBF (He et al., 2023) with ViT-B on ImageNet-1K. With $N \approx 1.28$ million real and $M$ synthetic training images, generation takes $M \times 0.095$ seconds on one A100 GPU with DiT-B (Table 8), and one epoch has $(N + M)/1024$ batches, each taking the per-batch training time in Table 9 on four A100 GPUs. For IDR with $M = 5N$, this gives $7{,}500 \times 0.825 \approx 6{,}188$ seconds, i.e., $1.72$ hours per epoch, which agrees with the measured training time of VUIB in Table 2. We assume that the per-batch training time does not depend on the size of the augmented training set and that all methods are fine-tuned for $100$ epochs with the total batch size of $1024$ (Section C); costs incurred outside the measured per-batch training time, if any, are not included. We do not report the cost of training the LDM ($2{,}800$ epochs, Section C), which is incurred once and shared by all methods since they use the same set of synthetic images (Section 4.1). The cost of the cross-validation for selecting $M$ is not included either.

It is observed in Table 17 that the estimated cost of IDR is higher than that of GenDataAgent, mainly because the cross-validated budget of IDR is larger ($5N$ versus $3N$). IDR requires about $1.56\times$ the total A100-hours of GenDataAgent, and its training time per epoch is about $53.7\%$ longer, although its per-batch training time is only $2.5\%$ higher. Generation accounts for $18.1\%$ to $19.7\%$ of these estimated totals, which exclude the one-time LDM training and the cross-validation. At the matched budget of $3N$, IDR has the same generation cost as GenDataAgent and an estimated $2.0\%$ higher total cost, and outperforms GenDataAgent by $0.8\%$ in top-1 accuracy. In compute-constrained set-

Table 17: Estimated cost of synthetic data generation and fine-tuning with ViT-B on ImageNet-1K, derived from Tables 8 and 9 as described in Section D.17. The number of synthetic samples is given in parentheses. Fine-tuning A100-hours are four times the wall-clock time of 100 epochs on four A100 GPUs. The one-time LDM training and the cross-validation are excluded. Totals are computed before rounding. †From Figure 3; the other Top-1 values are from Table 1.

| Methods | Top-1 | Generation (A100-Hours) | Training Time (Hours / Epoch) | Fine-tuning (A100-Hours) | Total (A100-Hours) |
|---|---|---|---|---|---|
| CBF ($2N$) | 84.4 | 67.6 | 0.74 | 297.5 | 365.1 |
| GenDataAgent ($3N$) | 84.5 | 101.3 | 1.12 | 447.2 | 548.6 |
| IDR ($3N$) | 85.3† | 101.3 | 1.15 | 458.3 | 559.7 |
| **IDR (Ours,** $5N$) | **85.6** | 168.9 | 1.72 | 687.5 | 856.4 |

tings, a smaller budget can be used for IDR, since IDR outperforms CBF and GenDataAgent at every budget from $N$ to $10N$ in Figure 3.

**Storage and Energy.** The storage of the synthetic images grows linearly with $M$: IDR, GenDataAgent, and CBF with their cross-validated budgets use 6.4, 3.84, and 2.56 million synthetic images, respectively. These numbers and the generation cost in Table 17 count only the selected $M$ images; the cross-validation over $\{N, 2N, \ldots, 10N\}$ uses a pool of up to $10N = 12.8$ million synthetic images shared by all methods, whose generation takes about 337.8 A100-hours. Assuming that each A100 GPU draws at most 400 W on average (the thermal design power of the A100 SXM GPU; the power draw is not measured), the GPU energy is at most 0.4 kW times the total A100-hours in Table 17, i.e., 342.6, 219.4, and 146.0 kWh for IDR, GenDataAgent, and CBF, respectively, excluding the LDM training, the cross-validation, and the energy consumed by CPUs, storage, and cooling.

**Scalability.** The soft assignments, $Q^{(t)}$, and VUIB are computed in $\Theta(N'C)$ time per epoch (Section F), and, given the assignments $\mathbb{1}_{\{\tilde{x}_i \in k\}}$, the prototypes in Equation (2) are obtained from two weighted sums over the $N'$ samples. Since the forward and backward passes of the classifier also take $\Theta(N'T_0)$ time, the 6.1 seconds per epoch for computing the prototypes and the distances between the learned features and the prototypes (Section D.6), which is about 0.1% of the estimated 6,188 seconds per epoch, is not expected to become a larger fraction of the training time at higher augmentation ratios for a fixed number of prototypes $C$. Moreover, Equation (2) is a ratio of two sums over the samples in $\mathcal{D}_{\text{aug}}$, and $Q^{(t)}$ is computed by accumulation over the samples (Algorithm 5). Each of them can therefore be accumulated in a streaming pass over mini-batches, with memory that depends on the number of prototypes $C$, the number of classes $C'$, and the feature dimension, but not on $N'$.

## E Proof of Theorem 3.1

We define

$$\phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right) = \frac{\exp\left(-\left\|\tilde{z}_i - c_a^{(\text{feat})}\right\|_2^2 - \left\|\tilde{x}_i - c_y^{(\text{input})}\right\|_2^2\right)}{\sum_{a'=1}^{C}\sum_{y'=1}^{C}\exp\left(-\left\|\tilde{z}_i - c_{a'}^{(\text{feat})}\right\|_2^2 - \left\|\tilde{x}_i - c_{y'}^{(\text{input})}\right\|_2^2\right)}, \quad a, y \in [C], \tag{13}$$

where $\left\{c_y^{(\text{input})}\right\}_{y=1}^{C}$ are learnable prototypes of the input features $\{\tilde{x}_i\}_{i=1}^{N'}$.

**Lemma E.1.** Let $\Pr\left[\tilde{X} \in y\right] = \sum_{i=1}^{N'} \mathbb{1}_{\{\tilde{y}_i = y\}}/N' := p_y$ for every $y \in [C]$, then

$$I(\tilde{Z}, \tilde{X}) \leq \frac{1}{N'}\sum_{i=1}^{N'}\sum_{a=1}^{C}\sum_{y=1}^{C}\phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right)\log\left(\frac{\phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right)}{p_y \phi(\tilde{z}_i, c_a^{(\text{feat})})}\right). \tag{14}$$

*Proof.* Then the joint probability $\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]$ is specified by

$$\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right] = \frac{1}{N'} \sum_{i=1}^{N'} \phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right). \tag{15}$$

It then follows from (15) and the log sum inequality that

$$\begin{aligned}
I(\tilde{Z}, \tilde{X}) &= \sum_{a=1}^{C}\sum_{y=1}^{C} \Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right] \log \frac{\Pr\left[\tilde{Z} \in a, \tilde{X} \in y\right]}{\Pr\left[\tilde{Z} \in a\right] \Pr\left[\tilde{X} \in y\right]} \\
&\leq \frac{1}{N'} \sum_{i=1}^{N'} \sum_{a=1}^{C} \sum_{y=1}^{C} \phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right) \log\left(\phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right)\right) \\
&\quad - \frac{1}{N'} \sum_{i=1}^{N'} \sum_{a=1}^{C} \sum_{y=1}^{C} \phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right) \log\left(\phi(\tilde{z}_i, c_a^{(\text{feat})}) p_y\right) \\
&= \frac{1}{N'} \sum_{i=1}^{N'} \sum_{a=1}^{C} \sum_{y=1}^{C} \phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right) \log\left(\frac{\phi\left(\tilde{z}_i \in a, \tilde{x}_i \in y\right)}{p_y \phi(\tilde{z}_i, c_a^{(\text{feat})})}\right).
\end{aligned} \tag{16}$$

□

**Lemma E.2.**

$$I(\tilde{Z}, \tilde{Y}) \geq \frac{1}{N'} \sum_{i=1}^{N'} \sum_{a=1}^{C} \sum_{y=1}^{C'} \phi(\tilde{z}_i, c_a^{(\text{feat})}) \mathbb{1}_{\{\tilde{y}_i = y\}} \log Q(\tilde{Z} \in a | \tilde{Y} = y) \tag{17}$$

*Proof.* Let $Q(\tilde{Z}|\tilde{Y})$ be a variational distribution. We have

$$\begin{aligned}
I(\tilde{Z}, \tilde{Y}) &= \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{\Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right]}{\Pr\left[\tilde{Z} \in a\right] \Pr\left[\tilde{Y} = y\right]} \\
&= \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{\Pr\left[\tilde{Z} \in a | \tilde{Y} = y\right] Q(\tilde{Z} \in a | \tilde{Y} = y)}{\Pr\left[\tilde{Z} \in a\right] Q(\tilde{Z} \in a | \tilde{Y} = y)} \\
&\geq \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{\Pr\left[\tilde{Z} \in a | \tilde{Y} = y\right]}{Q(\tilde{Z} \in a | \tilde{Y} = y)} \\
&\quad + \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{Q(\tilde{Z} \in a | \tilde{Y} = y)}{\Pr\left[\tilde{Z} \in a\right]} \\
&= \mathrm{KL}\left(P(\tilde{Z}|\tilde{Y}) \middle\| Q(\tilde{Z}|\tilde{Y})\right) + \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{Q(\tilde{Z} \in a | \tilde{Y} = y)}{\Pr\left[\tilde{Z} \in a\right]} \\
&\geq \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log \frac{Q(\tilde{Z} \in a | \tilde{Y} = y)}{\Pr\left[\tilde{Z} \in a\right]} \\
&= \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log Q(\tilde{Z} \in a | \tilde{Y} = y) + H\left(P(\tilde{Z})\right) \\
&\geq \sum_{a=1}^{C}\sum_{y=1}^{C'} \Pr\left[\tilde{Z} \in a, \tilde{Y} = y\right] \log Q(\tilde{Z} \in a | \tilde{Y} = y)
\end{aligned}$$

$$\geq \frac{1}{N'}\sum_{i=1}^{N'}\sum_{a=1}^{C}\sum_{y=1}^{C'}\phi(\tilde{z}_i, c_a^{(\text{feat})})\mathbb{1}_{\{\tilde{y}_i=y\}}\log Q(\tilde{Z}\in a|\tilde{Y}=y), \tag{18}$$

where the last inequality follows from the joint probability $\Pr\left[\tilde{Z}\in a, \tilde{Y}=y\right] = \frac{1}{N'}\sum_{i=1}^{N'}\phi(\tilde{z}_i, c_a^{(\text{feat})})\mathbb{1}_{\{\tilde{y}_i=y\}}$. □

*Proof of Theorem 3.1.* Equation 1 in Theorem 3.1 of the main paper follows from $\text{IB}(\mathcal{C}(\theta,\Theta),\Theta,\mathcal{D}_{\text{aug}}) = I(\tilde{Z},\tilde{X}) - I(\tilde{Z},\tilde{Y})$, the upper bound for $I(\tilde{Z},\tilde{X})$ in Lemma E.1 and the lower bound for $I(\tilde{Z},\tilde{Y})$ in Lemma E.2. □

### E.1 Computation of $Q^{(t)}(\tilde{Z}|\tilde{Y})$

The variational distribution $Q^{(t)}(\tilde{Z}|\tilde{Y})$ can be computed by

$$Q^{(t)}(\tilde{Z}\in a|\tilde{Y}=y) = \Pr\left[\tilde{Z}\in a|\tilde{Y}=y\right] = \frac{\sum_{i=1}^{N'}\phi(\tilde{z}_i, c_a^{(\text{feat})})\mathbb{1}_{\{\tilde{y}_i=y\}}}{\sum_{i=1}^{N'}\mathbb{1}_{\{\tilde{y}_i=y\}}}, \quad a\in[C], y\in[C'] \tag{19}$$

Algorithm 5 computes $Q^{(t)}(\tilde{Z}|\tilde{Y})$ efficiently with a time complexity of $\Theta(N'C + N'T_0)$, where $C$ is the number of prototypes, $C'$ is the number of classes, $N'$ is the number of training samples, and $T_0$ denotes the computational complexity of a forward and backward pass of the neural network with respect to each training sample. $Q^{(t)}(\tilde{Z}|\tilde{Y})$ is the variational distribution $Q(\tilde{Z}|\tilde{Y})$ computed at the $t$-th epoch in Algorithm 3.

### E.2 Issue about the Upper Bound for the IB Loss in IDS (Wang et al., 2025) and DCS-Transformer (Wang et al., 2024)

The derivation of the upper bound for the IB loss in IDS (Wang et al., 2025) and DCS-Transformer (Wang et al., 2024) follows a similar recipe, with an upper bound for the mutual information between the learned features $\widehat{Z}$ and the input features $\widehat{X}$, $I(\widehat{Z},\widehat{X})$, and a lower bound for the mutual information between the learned features $\widehat{Z}$ and the class label $\widehat{Y}$, $I(\widehat{Z},\widehat{Y})$. However, the upper bounds for $I(\widehat{Z},\widehat{X})$ in both IDS (Wang et al., 2025) and DCS-Transformer (Wang et al., 2024) is

$$\begin{aligned} I(\widehat{Z},\widehat{X}) &\leq \frac{1}{N'}\sum_{i=1}^{N'}\sum_{a=1}^{C}\sum_{y=1}^{C}\phi(\widehat{z}_i, c_a^{(\text{feat})})\phi(\widehat{x}_i, c_y^{(\text{input})})\log\phi(\widehat{x}_i, c_y^{(\text{input})}) \\ &= \frac{1}{N'}\sum_{i=1}^{N'}\sum_{y=1}^{C}\phi(\widehat{x}_i, c_y^{(\text{input})})\log\phi(\widehat{x}_i, c_y^{(\text{input})}), \end{aligned} \tag{20}$$

where the last equality follows from $\sum_{a=1}^{C}\phi(\widehat{z}_i, c_a^{(\text{feat})}) = 1$. In (20), $\widehat{x}_i, \widehat{z}_i$ stand for the input training feature and the learned feature, respectively, and $c_a^{(\text{feat})}, c_y^{(\text{input})}$ stand for the prototypes of the input training features and the learned features. Compared to our derived upper bound for $I(\widehat{Z},\widehat{X})$ on the RHS of (14) of Lemma E.1, the upper bound for the RHS of (20) does not depend on the learned features $\{\widehat{z}_i\}$ and it only depends on the input training data. As a result, the optimization of the upper bound for $I(\widehat{Z},\widehat{X})$ in the IB loss in (20) cannot optimize the weights of the neural network for reducing the IB loss for data reweighting, making the upper bound for $I(\widehat{Z},\widehat{X})$ vacuous. In a strong contrast, our upper bound (14) depends on both the real training data and the learned features $\{\widehat{z}_i\}$, leading to improved performance of our IDR over IDS in Tables 1, 3, 6, and 7 of Sections 4.1, 4.4, D.3, and D.4.

**Algorithm 4** MI Minimization with vCLUB (Algorithm 1 in CLUB (Cheng et al., 2020))

**for** each training iteration **do**
  Sample $\{(x_i, z_i)\}_{i=1}^{N'}$ from $p_\sigma(x, z)$
  Compute log-likelihood $\mathcal{L}(\theta) = \frac{1}{N'} \sum_{i=1}^{N'} \log q_\theta(z_i|x_i)$
  Update $q_\theta(z|x)$ by maximizing $\mathcal{L}(\theta)$
  **for** $i = 1$ to $N'$ **do**
    $L_i = \log q_\theta(z_i|x_i) - \frac{1}{N'} \sum_{j=1}^{N'} \log q_\theta(z_j|x_i)$
  **end for**
  Update $p_\sigma(x, z)$ by minimizing $\widehat{I}_{\text{vCLUB}} = \frac{1}{N'} \sum_{i=1}^{N'} L_i$
**end for**

**Algorithm 5** Efficient Computation of $Q^{(t)}(\tilde{Z} \in a \mid \tilde{Y} = y)$

**Input:** The precomputed soft assignments $\phi(\tilde{z}_i, c_a^{(\text{feat})})$ for $i \in [N']$ and $a \in [C]$, the class labels $\{\tilde{y}_i\}_{i=1}^{N'}$, the number of prototypes $C$, and the number of classes $C'$. Here $\{\tilde{z}_i\}_{i=1}^{N'}$ are computed at the $t$-th epoch in Algorithm 3.

**Output:** Conditional distribution matrix $Q \in \mathbb{R}^{C \times C'}$

Initialize $Q \leftarrow \mathbf{0}^{C \times C'}$ and count vector $M \leftarrow \mathbf{0}^{C'}$.
**for** $i = 1$ to $N'$ **do**
  **for** $a = 1$ to $C$ **do**
    $Q[a, \tilde{y}_i] \leftarrow Q[a, \tilde{y}_i] + \phi(\tilde{z}_i, c_a^{(\text{feat})})$
  **end for**
  $M[\tilde{y}_i] \leftarrow M[\tilde{y}_i] + 1$
**end for**
**for** $y = 1$ to $C'$ **do**
  **for** $a = 1$ to $C$ **do**
    $Q[a, y] \leftarrow Q[a, y]/M[y]$
  **end for**
**end for**
**return** $Q$

# F Computational Complexity Analysis

Given samples $\{(x_i, z_i)\}_{i=1}^{N'}$ drawn from the joint distribution $p_\sigma(x, z) = p_\sigma(z|x)p(x)$, the goal of CLUB is to optimize the parameters of the predictive neural network $p_\sigma(z|x)$ such that the resulting joint model $p_\sigma(x, z)$ induces minimal mutual information between $x$ and $z$. Here, $p_\sigma(z|x)$ serves as the main predictive model, such as an encoding model or a classification model, while $p(x)$ denotes the empirical data distribution. In addition, $q_\theta(z|x)$ is a variational conditional distribution, also implemented as a neural network, used to approximate $p(z|x)$ and to provide a differentiable estimator of mutual information. The optimization alternates between updating $q_\theta(z|x)$ by maximizing the log-likelihood $\mathcal{L}(\theta) = \frac{1}{N'}\sum_{i=1}^{N'} \log q_\theta(z_i|x_i)$ to improve the approximation accuracy, and updating $p_\sigma(x, z)$ by reducing the upper bound for the mutual information, $I(x, z)$, $\widehat{I}_{\text{vCLUB}} = \frac{1}{N'}\sum_{i=1}^{N'} L_i$, where $L_i = \log q_\theta(z_i|x_i) - \frac{1}{N'}\sum_{j=1}^{N'} \log q_\theta(z_j|x_i)$. Through this alternating process, the algorithm jointly learns an accurate variational estimator $q_\theta$ and a predictive model $p_\sigma$. The complete training procedure is summarized in Algorithm 4.

Suppose the computation of $q_\theta(z|x)$ costs $T_q$ and that of $p_\sigma(z|x)$ costs $T_p$. Computing $\mathcal{L}(\theta) = \frac{1}{N'}\sum_{i=1}^{N'} \log q_\theta(z_i|x_i)$ requires $\Theta(N'T_q)$ time. The computation of vCLUB $\widehat{I}_{\text{vCLUB}} = \frac{1}{N'}\sum_{i=1}^{N'}\left[\log q_\theta(z_i|x_i) - \frac{1}{N'}\sum_{j=1}^{N'} \log q_\theta(z_j|x_i)\right]$ requires computing $[\log q_\theta(z_j|x_i)]$ for $i, j \in [N']$, which requires $\Theta(N'^2T_q)$ time. Updating $p_\sigma$ for reducing $\widehat{I}_{\text{vCLUB}}$ over the same $N'$ inputs adds $N'T_p$ time. Hence, one training epoch requires $\Theta(N'^2T_q + N'T_q + N'T_p) = \Theta(N'^2T_q + N'T_p)$. This analysis assumes that $\widehat{I}_{\text{vCLUB}}$ is evaluated over all the $N'$ samples at once. When the samples drawn in each training iteration of Algorithm 4 form a mini-batch of size $B$, the pairwise terms $\log q_\theta(z_j|x_i)$ cost $\Theta(B^2T_q)$ per mini-batch, and one epoch with $N'/B$ mini-batches requires $\Theta(N'BT_q + N'T_p)$ time, which is linear in $N'$. In this case, CLUB still requires the additional $\Theta(N'BT_q)$ time for the pairwise evaluation of $q_\theta$ and the training of the variational network $q_\theta$, neither of which is needed by VUIB, whose per-epoch cost is $\Theta(N'T_0 + N'C)$ as analyzed below. This is consistent with the measured training time with $5N$ synthetic samples in Table 2, where CLUB takes 3.47 hours per epoch and VUIB takes 1.72 hours per epoch, i.e., CLUB takes about $2\times$ the training time of VUIB.

**Computational Complexity of VUIB.** Herein we analyze the computational complexity for computing the upper bound, VUIB, for the IB loss. Let $T_0$ denote the complexity of a forward and backward computation of the model predicting $\tilde{z}_i$ for one sample. For each epoch, the computation of $\{\tilde{z}_i\}_{i=1}^{N'}$ using the classification network takes $\Theta(N'T_0)$ time. Once $\{\tilde{z}_i\}_{i=1}^{N'}$ are computed, we pre-compute $\phi(\tilde{z}_i, c_a^{(\text{feat})})$ for $i \in [N']$ and $a \in [C]$, which takes $\Theta(N'C)$ time.

To compute $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}})$, we separately compute $\frac{1}{N'}\sum_{i=1}^{N'} U_i$, which is the upper bound for $I(\tilde{Z}, \tilde{X})$, and $\frac{1}{N'}\sum_{i=1}^{N'} V_i$, which is the lower bound for $I(\tilde{Z}, \tilde{Y})$. Let $A_{ia} := \exp(-\|\tilde{z}_i - c_a^{(\text{feat})}\|_2^2)$ and $B_{iy} := \exp(-\|\tilde{x}_i - c_y^{(\text{input})}\|_2^2)$ for all $a \in [C]$ and $y \in [C]$. Let $D_i := \sum_{a'=1}^{C}\sum_{y'=1}^{C} A_{ia'}B_{iy'} = \widehat{A}_i\widehat{B}_i$, where $\widehat{A}_i := \left(\sum_{a'=1}^{C} A_{ia'}\right), \widehat{B}_i := \left(\sum_{y'=1}^{C} B_{iy'}\right)$. For each $i \in [N']$, we first pre-compute all $A_{ia}$ and $B_{iy}$ for $a \in [C]$ and $y \in [C]$, which takes $\Theta(C)$ time. For each $i \in [N']$, we then pre-compute $\widehat{A}_i$, $\widehat{B}_i$, and $D_i$, which takes $\Theta(C)$ time. Then $U_i$ can be computed by

$$
\begin{aligned}
U_i &= \sum_{a=1}^{C}\sum_{y=1}^{C} \phi(\tilde{z}_i \in a, \tilde{x}_i \in y) \log\left(\frac{\phi(\tilde{z}_i \in a, \tilde{x}_i \in y)}{p_y \phi(\tilde{z}_i, c_a^{(\text{feat})})}\right) \\
&= \frac{1}{D_i}\sum_{a=1}^{C}\sum_{y=1}^{C} A_{ia}B_{iy} \log\left(\frac{A_{ia}B_{iy}/D_i}{p_y \phi(\tilde{z}_i, c_a^{(\text{feat})})}\right) \\
&= -\log D_i + \frac{1}{D_i}\sum_{a=1}^{C}\sum_{y=1}^{C} A_{ia}B_{iy}\left(\log A_{ia} + \log B_{iy} - \log p_y - \log \phi(\tilde{z}_i, c_a^{(\text{feat})})\right)
\end{aligned}
$$

$$= -\log D_i - \frac{\widehat{A}_i}{D_i}\sum_{y=1}^{C} B_{iy}\log p_y + \frac{1}{D_i}\left(\sum_{a=1}^{C} A_{ia}\log A_{ia}\right)\widehat{B}_i$$
$$+ \frac{\widehat{A}_i}{D_i}\left(\sum_{y=1}^{C} B_{iy}\log B_{iy}\right) - \frac{\widehat{B}_i}{D_i}\left(\sum_{a=1}^{C} A_{ia}\log \phi(\tilde{z}_i, c_a^{(\text{feat})})\right).$$

For each $i \in [N']$, the computation of $\frac{\widehat{A}_i}{D_i}\sum_{y=1}^{C} B_{iy}\log p_y$, $\frac{1}{D_i}\left(\sum_{a=1}^{C} A_{ia}\log A_{ia}\right)\widehat{B}_i$, $\frac{\widehat{A}_i}{D_i}\left(\sum_{y=1}^{C} B_{iy}\log B_{iy}\right)$, and $\frac{\widehat{B}_i}{D_i}\left(\sum_{a=1}^{C} A_{ia}\log \phi(\tilde{z}_i, c_a^{(\text{feat})})\right)$ each takes $\Theta(C)$ time. As a result, the computation of $U_i$ for each $i \in [N']$ takes $\Theta(C + C + 1 + C + C + C + C) = \Theta(C)$ time. Therefore, the computation of $\frac{1}{N'}\sum_{i=1}^{N'} U_i$, which is the upper bound for $I(\tilde{Z}, \tilde{X})$, takes $\Theta(N'C)$ time.

Given the soft assignment matrix $\phi(\tilde{z}_i, c_a^{(\text{feat})})$ for all $i \in [N']$ and $a \in [C]$, the conditional distribution matrix $Q(\tilde{Z} \in a \mid \tilde{Y} = y) \in \mathbb{R}^{C\times C'}$ can be efficiently computed following Algorithm 5. We denote by $Q[a, y] = Q(\tilde{Z} \in a \mid \tilde{Y} = y)$ the $(a, y)$-th entry of $Q$, representing the conditional probability that a learned feature $\tilde{Z}$ belongs to prototype $a$ given class label $y$. Each entry $Q[a, y]$ is computed by aggregating the soft assignment values $\phi(\tilde{z}_i, c_a^{(\text{feat})})$ over all $i \in [N']$ such that $\tilde{y}_i = y$, followed by normalization with respect to the total number of samples in that class. The accumulation step (lines 2–7 in Algorithm 5) requires $\Theta(N'C)$ time, while the normalization step (lines 8–12 in Algorithm 5) requires $\Theta(CC')$ time. Since $N' \gg C$, the computational complexity of computing $Q$ is $\Theta(N'C + CC') = \Theta(N'C)$.

Let $\mathcal{I}_y := \{i \in [N'] | \tilde{y}_i = y\}$ be the index set of the samples from the class $y$ for $y \in [C']$ in the augmented training set. The lower bound for the mutual information $I(\tilde{Z}, \tilde{Y})$ can be computed by

$$\frac{1}{N'}\sum_{i=1}^{N'} V_i = \frac{1}{N'}\sum_{i=1}^{N'}\sum_{a=1}^{C}\sum_{y=1}^{C'} \phi(\tilde{z}_i, c_a^{(\text{feat})})\mathbb{1}_{\{\tilde{y}_i = y\}}\log Q(\tilde{Z} \in a | \tilde{Y} = y)$$
$$= \frac{1}{N'}\sum_{a=1}^{C}\sum_{y=1}^{C'}\sum_{i\in\mathcal{I}_y} \phi(\tilde{z}_i, c_a^{(\text{feat})})\log Q(\tilde{Z} \in a | \tilde{Y} = y).$$

For each $a \in [C]$, the computation of $\sum_{y=1}^{C'}\sum_{i\in\mathcal{I}_y} \phi(\tilde{z}_i, c_a^{(\text{feat})})\log Q(\tilde{Z} \in a | \tilde{Y} = y)$ takes $\sum_{y=1}^{C'} |\mathcal{I}_y| = \Theta(N')$ time. As a result, the computation of $\frac{1}{N'}\sum_{i=1}^{N'} V_i$, which is the lower bound for $I(\tilde{Z}, \tilde{Y})$, takes $\Theta(N'C)$ time.

Since $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}}) = \frac{1}{N'}\sum_{i=1}^{N'} U_i - \frac{1}{N'}\sum_{i=1}^{N'} V_i$, the overall computation cost for $\text{VUIB}(\mathcal{C}(\theta, \Theta), \Theta, \mathcal{D}_{\text{aug}})$ is $\Theta(N'T_0 + N'C + N'C + N'C + N'C) = \Theta(N'T_0 + N'C)$.

# G Further Discussions

## G.1 Discussion on the IB Trade-off Parameter

The IB principle is commonly formulated as the minimization of the IB Lagrangian $I(\tilde{Z}, \tilde{X}) - \beta_{\text{IB}} I(\tilde{Z}, \tilde{Y})$ with a trade-off parameter $\beta_{\text{IB}} > 0$ (Tishby et al., 2000). The IB loss $\text{IB}(\tilde{Z}, \tilde{X}, \tilde{Y}) = I(\tilde{Z}, \tilde{X}) - I(\tilde{Z}, \tilde{Y})$ in Section 3.2 corresponds to $\beta_{\text{IB}} = 1$, and accordingly $U_i$ and $V_i$ have equal weights in VUIB in Theorem 3.1. In IDR, VUIB is only used to update the parameters $\theta$ of the sample reweighting network, while the parameters $\Theta$ of the classifier are updated by the reweighted cross-entropy loss $\mathcal{L}_{\text{train}}$, which does not contain an MI term (Section 3.3 and Algorithm 3). Multiplying VUIB by a constant $\lambda > 0$ multiplies $\nabla_\theta$VUIB by $\lambda$. For the update rule of $\theta$ in Section 3.3, this is equivalent to replacing $\eta_\theta$ by $\lambda\eta_\theta$, and for the AdamW optimizer used to update $\theta$ (Section C), it leaves the update unchanged up to the constant added to the denominator of AdamW for numerical stability. Therefore, a global weight on the IB loss does not introduce a degree of freedom beyond the learning rate $\eta_\theta$. A relative weight $\beta_{\text{IB}} \neq 1$ between the two MI terms, i.e., replacing $U_i - V_i$ by

$U_i - \beta_{\text{IB}} V_i$ in VUIB, still gives an upper bound for $I(\tilde{Z}, \tilde{X}) - \beta_{\text{IB}} I(\tilde{Z}, \tilde{Y})$ for $\beta_{\text{IB}} > 0$ by Lemma E.1 and Lemma E.2. Since $\Theta$ is updated only by $\mathcal{L}_{\text{train}}$, such a $\beta_{\text{IB}}$ can affect the classifier only through the importance weights $g_\theta(\tilde{x}_i)$. In this paper, $\beta_{\text{IB}}$ is fixed to 1.

### G.2 Relation to Task-Specific Generative Data Augmentation for Dense Prediction

Task-specific GDA methods synthesize training data for the dense prediction task itself, such as data augmentation for object detection via controllable diffusion models (Fang et al., 2024) and mask-centric generative data augmentation for label-scarce semantic segmentation ([MatchMask authors – TO BE VERIFIED], 2026). The object detection and semantic segmentation experiments in Section 4.4 and Section D.3 address a different question, namely the transferability of the features learned by IDR on ImageNet-1K. In these experiments, IDR and the baselines are applied when training the backbones on the augmented ImageNet-1K training set. For object detection, the Swin-B models in Table 1 are used, so the numbers of synthetic ImageNet-1K images are those reported for Swin-B in Table 1, e.g., $5N$ for IDR and $3N$ for GenDataAgent. The Cascade Mask R-CNN (Cai & Vasconcelos, 2021) and UperNet (Xiao et al., 2018) models are then trained on MS-COCO and on the training split of ADE20K, respectively, following the protocols of (Liu et al., 2021). Therefore, IDR is evaluated for training the backbone on ImageNet-1K, while task-specific GDA targets the training of the downstream dense prediction model, and the two are complementary in principle, e.g., a backbone trained by IDR could be used in a downstream model trained with task-specific GDA. Moreover, VUIB in Theorem 3.1 is defined with image-level class labels, so applying the reweighting of IDR to synthetic detection or segmentation samples would require extending VUIB beyond image-level class labels, e.g., to box-level or pixel-level annotations.