Generative Semi-supervised Learning with Meta-Optimized Synthetic Samples

Shin'ya Yamaguchi SHINYA.YAMAGUCHI@NTT.COM
NTT / Kyoto University

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Semi-supervised learning (SSL) is a promising approach for training deep classification models using labeled and unlabeled datasets. However, existing SSL methods rely on a large unlabeled dataset, which may not always be available in many real-world applications due to legal constraints (e.g., GDPR). In this paper, we investigate the research question: *Can we train SSL models without real unlabeled datasets?* Instead of using real unlabeled datasets, we propose an SSL method using synthetic datasets generated from generative foundation models trained on datasets containing millions of samples in diverse domains (e.g., ImageNet). Our main concepts are identifying synthetic samples that emulate unlabeled samples from generative foundation models and training classifiers using these synthetic samples. To achieve this, our method is formulated as an alternating optimization problem: (i) meta-learning of generative foundation models and (ii) SSL of classifiers using real labeled and synthetic unlabeled samples. For (i), we propose a meta-learning objective that optimizes latent variables to generate samples that resemble real labeled samples and minimize the validation loss. For (ii), we propose a simple unsupervised loss function that regularizes the feature extractors of classifiers to maximize the performance improvement obtained from synthetic samples. We confirm that our method outperforms baselines using generative foundation models on SSL. We also demonstrate that our methods outperform SSL using real unlabeled datasets in scenarios with extremely small amounts of labeled datasets. This suggests that synthetic samples have the potential to provide improvement gains more efficiently than real unlabeled data.

Keywords: generative models; semi-supervised learning; meta-learning

1. Introduction

Semi-supervised learning (SSL) is a promising approach for training deep neural network models with a limited amount of labeled data and a large amount of unlabeled data. Recent studies on SSL have shown that the labeling cost to achieve high-performance models can be significantly reduced by using the unlabeled dataset to train the models with pseudo-labeling and consistency regularization (Bachman et al., 2014; Xie et al., 2020; Sohn et al., 2020). For example, Wang et al. (2023) have reported that their SSL method can achieve 94.22% test accuracy on CIFAR-10 with only one label per class. This indicates that modern SSL methods can realize practical models with minimal labeling costs. However, whether labeled or not, large-scale datasets are becoming more challenging to obtain and use for machine learning models due to privacy regulations (e.g., GDPR in the EU).

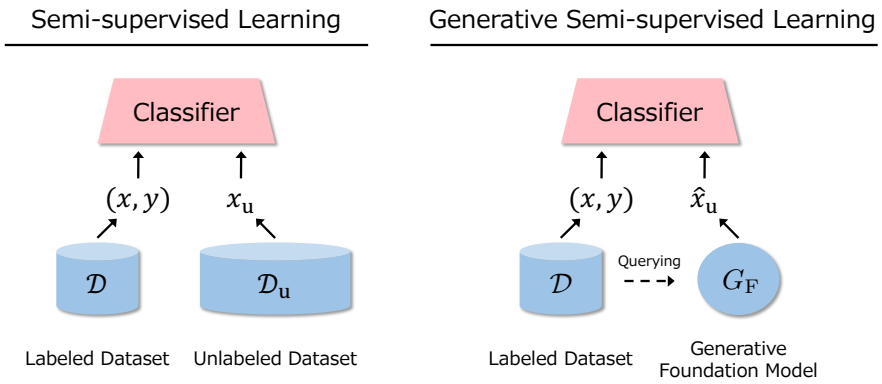


Figure 1: Comparison of semi-supervised learning (SSL) and generative semi-supervised learning (gSSL). In gSSL, we use a generative foundation model G_{F} to compute unsupervised losses instead of a real unlabeled dataset $\mathcal{D}_{\mathrm{u}}$. To this end, we generate synthetic unlabeled samples by querying G_{F} with information of the labeled dataset $\mathcal{D}$.

To train deep models in a situation where it is challenging to obtain datasets, recent studies using synthetic datasets from deep generative models have attracted much attention in the context of proxying real datasets (He et al., 2023; van Breugel et al., 2023). This approach has been intensively discussed in the community[1] and regarded as a promising method for privacy protection since generative models such as GANs (Goodfellow et al., 2014) can produce realistic samples while guaranteeing certain differential privacy (Lin et al., 2021). If the synthetic samples can be used as unlabeled datasets in SSL, we can train a high-performance model without real unlabeled datasets and privacy risks. Thus, we investigate a research question: *Can we train SSL models with synthetic unlabeled datasets instead of real ones?*

In this paper, we explore a new problem setting called *generative semi-supervised learning* (gSSL), where the semi-supervised learners use synthetic unlabeled samples generated from a *generative foundation model* instead of real unlabeled samples (Figure 1). Generative foundation models are conditional generative models pre-trained on large external datasets containing millions of samples from diverse domains (e.g., ImageNet). Thanks to recent advances (Brock et al., 2019; Sauer et al., 2022), generative foundation models can accurately output synthetic samples in various domains from inputs of latent variables and conditional labels. Therefore, we can expect the synthetic samples to perform as the unlabeled datasets in SSL when the training data space overlaps the data space estimated by the generative foundation models.

In gSSL, there are important challenges according to the following two concrete research questions: (i) *How do we find optimal synthetic samples from the generative foundation models for SSL?* and (ii) *How do we train models with synthetic samples that do not belong to the training class categories?* For (i), since generative foundation models do not necessarily have the same class categories in the training datasets, we need to find synthetic samples from the generative models related to training datasets. Furthermore, it is essential to find

1. NeurIPS Synthetic Data Workshop (https://www.syntheticdata4ml.vanderschaar-lab.com/)

synthetic samples that can improve the classifier performance through the unsupervised loss of SSL. For (ii), even if we can find helpful synthetic samples from the generative foundation models, it is not obvious how these samples can be optimally used to maximize the performance of the classifiers. This is because the synthetic samples are matched to training datasets with respect to the domain (data space), not the class label spaces. Since existing SSL methods assume that unlabeled samples belong to the training class categories, the mismatch between real and synthetic samples in the class label spaces can be detrimental to SSL models.

To address these two challenges, we propose a method called *meta-pseudo semi-supervised learning* (MP-SSL). MP-SSL consists of two techniques corresponding to the two research questions: (i) *latent meta-optimization* (LMO) and (ii) *synthetic consistency regularization* (SCR). In LMO, we optimize latent variables that are input to generative foundation models to find synthetic samples that resemble unlabeled training data. To find optimal synthetic samples for SSL models, LMO meta-optimizes the parameters to minimize the validation losses of the target classifier. Furthermore, LMO also minimizes the gaps in the feature spaces between real and synthetic samples to align the domain gap and make the synthetic samples perform as unlabeled data. SRC is a novel unsupervised loss term without the use of pseudo training labels. Unlike existing SSL methods depending on real unlabeled data and the pseudo training labels, the SCR loss is designed as a feature regularization term. This design choice is to avoid the negative effects of the synthetic samples caused by the mismatch of the class label spaces. Specifically, SCR penalizes the feature extractors by maximizing the similarity between variations of a synthetic sample, which is inspired by consistency regularization (Bachman et al., 2014; Xie et al., 2020; Sohn et al., 2020). Since SRC is independent of the relationship between training and foundation label spaces, it can leverage the valuable information contained in synthetic unlabeled data to train the model without negative effects. The training objective of MP-SSL is formalized as an alternating optimization problem of updating latent variables and updating training models through SSL with SCR.

To evaluate the effectiveness of MP-SSL, we conduct the experiments on multiple datasets by comparing MP-SSL with competitors, including P-SSL (Yamaguchi et al., 2022). We also compare MP-SSL with SSL methods using real unlabeled datasets. The results show that MP-SSL outperforms the real SSL methods when the labeled datasets are small. This suggests that synthetic samples can promote more effective learning than real unlabeled samples, especially in cases where the number of labels is extremely small. We believe this work will be the baseline for developing a new research area, generative semi-supervised learning.

Our contributions are summarized as follows.

- We propose a new problem setting of SSL called generative semi-supervised learning (gSSL), where the unlabeled samples are provided by generative foundation models instead of real unlabeled datasets.
- We introduce a training method for gSSL called MP-SSL, which finds optimal synthetic samples performing as unlabeled data through meta-optimizing latent variables and trains a classifier with a feature regularization with the synthetic samples.
- We confirm that MP-SSL can outperform simple baselines of the gSSL setting and outperform SSL methods with real unlabeled datasets in small amounts of labels.

2. Preliminary

2.1. Problem Setting

We consider a classification problem in which we train a neural network model $f_\theta : \mathcal{X} \to \mathcal{Y}$ on a labeled dataset $\mathcal{D} = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^N$, where $\mathcal{X}$ and $\mathcal{Y}$ are the input and output label spaces, respectively. In this setting, we can use a generative foundation model $G_\mathrm{F} : \mathcal{Z}_\mathrm{F} \times \mathcal{Y}_\mathrm{F} \to \mathcal{X}_\mathrm{F}$, where $\mathcal{Z}_\mathrm{F}$ is the latent space, $\mathcal{Y}_\mathrm{F}$ is the foundation label space, and $\mathcal{X}_\mathrm{F}$ is the output sample space. We assume that G_F is pre-trained on a large-scale dataset (e.g., ImageNet) and the output sample space $\mathcal{X}_\mathrm{F}$ contains a subset $\mathcal{X}'$ related to $\mathcal{X}$, i.e., $\mathcal{X}_\mathrm{F} \supset \mathcal{X}' \approx \mathcal{X}$. An input latent variable $z \in \mathcal{Z}_\mathrm{F}$ is sampled from a standard Gaussian distribution $\mathcal{N}(0, I)$. f_θ is defined by a composition of a feature extractor g_ψ and a classifier h_ω, i.e., $f_\theta = h_\omega \circ g_\psi$ and $\theta = [\psi, \omega]$. To validate f_θ, we can use a small validation dataset $\mathcal{D}_\mathrm{val} = \{(x^i_\mathrm{val}, y^i_\mathrm{val}) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{N_\mathrm{val}}$, which has no intersection with $\mathcal{D}$ (i.e., $\mathcal{D} \cap \mathcal{D}_\mathrm{val} = \emptyset$).

2.2. Semi-supervised Learning

Given a labeled dataset $\mathcal{D}$ and an unlabeled dataset $\mathcal{D}_\mathrm{u} = \{x^i \in \mathcal{X}\}_{i=1}^{N_\mathrm{u}}$, SSL to train f_θ is formulated as the following minimization problem.

$$\min_\theta \mathcal{L}(\theta) + \lambda_\mathrm{u} \mathcal{L}_\mathrm{u}(\theta), \tag{1}$$

$$\mathcal{L}(\theta) = \frac{1}{N} \sum\nolimits_{(x,y)\in\mathcal{D}} \ell(f_\theta(x), y) \tag{2}$$

$$\mathcal{L}_\mathrm{u}(\theta) = \frac{1}{N_\mathrm{u}} \sum\nolimits_{x_\mathrm{u}\in\mathcal{D}_\mathrm{u}} \ell_\mathrm{u}(f_\theta(x_\mathrm{u})) \tag{3}$$

where ℓ is a supervised loss for a labeled sample (e.g., cross-entropy loss), ℓ_u is an unsupervised loss for an unlabeled sample x_u, and λ_u is a hyperparameter for balancing $\mathcal{L}$ and $\mathcal{L}_\mathrm{u}$. SSL assumes a large amount of unlabeled data (i.e., $N \ll N_\mathrm{u}$). This assumption has long been justified on the premise that the difficulty of the dataset creation is centered on labeling, and the collection of unlabeled data can be easily done (Chapelle et al., 2006). However, unlabeled data are often unavailable due to privacy concerns. Starting with the EU's GDPR, privacy protection legislation has been developed globally, and creating large-scale datasets requires satisfying the privacy policy under the law. This paper explores an alternative SSL approach without collecting large-scale unlabeled datasets.

2.3. Generative Semi-supervised Learning

Generative semi-supervised learning (gSSL) is a variant of SSL where $\mathcal{D}_\mathrm{u}$ is prohibited from being accessed and the unlabeled data x_u is provided by a generative foundation model G_F by

$$x_\mathrm{u} = G_\mathrm{F}(z, \hat{y}_\mathrm{F}), \tag{4}$$

where $\hat{y}_\mathrm{F}$ is an estimated foundation label produced by gSSL algorithm. The gSSL algorithms have been rarely studied except for a prior work by Yamaguchi et al. (2022). In a transfer learning setting where the target and source architectures are not consistent, Yamaguchi et al. (2022) have proposed a method called pseudo semi-supervised learning

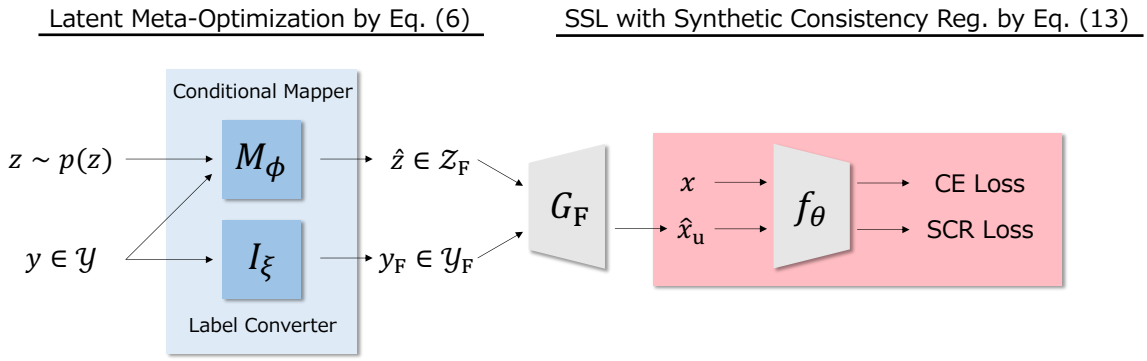


Figure 2: Overview of MP-SSL. We first generate a transformed latent variable $\hat{z}$ and a pseudo foundation label y_F through conditional mapper M_ϕ and label converter I_ξ. Then, we produce a pseudo unsupervised sample $\hat{x}_\mathrm{u} = G_\mathrm{F}(\hat{z}, \hat{y}_\mathrm{F})$ for semi-supervised learning (SSL) of f_θ. To find the optimal $\hat{z}$ and y_F, we update M_ϕ and I_ξ by latent meta-optimization (LMO, Eq. (6)). In the training of f_θ, we use the loss of synthetic consistency regularization (SCR, Eq. (13)) instead of existing SSL loss terms.

(P-SSL). Although P-SSL is focused on transfer learning, we consider it a simple baseline of gSSL. P-SSL trains f_θ by using Eq. (1) and estimates a foundation label $\hat{y}_\mathrm{F}$ as

$$\hat{y}_\mathrm{F} = f_{\theta_\mathrm{F}}(x), \tag{5}$$

where f_{θ_F} is a classifier pre-trained on a foundation dataset (e.g., an ImageNet pre-trained classifier). That is, P-SSL interprets the training sample x as the conditional sample of an interpolated class in $\mathcal{Y}_\mathrm{F}$ through the output of f_{θ_F}. This assumes the existence of f_{θ_F} and $y \in \mathcal{Y}$ can be semantically approximated by the soft foundation labels, i.e., $y_i \in \mathcal{Y} \approx f_{\theta_\mathrm{F}}(x_i) \in \mathcal{Y}_\mathrm{F}$. However, the synthetic samples by Eq. (5) do not always contribute to the performance of f_θ because the above assumption does not necessarily hold, and the synthetic samples are not directly optimized to improve f_θ. In fact, Yamaguchi et al. (2022) have reported that the performance gain by P-SSL is limited when the training datasets are not well approximated by Eq. (5). To stably improve the performance of f_θ, we present a meta-learning based SSL approach, which does not require the label assumptions.

3. Proposed Method

In this section, we describe our proposed method called MP-SSL. MP-SSL is composed of (i) latent meta-optimization (LMO) and (ii) synthetic consistency regularization (SCR). LMO finds synthetic samples performing as unlabeled data in SSL through meta-optimizing input latent variables and foundation class labels. SCR penalizes a feature extractor by maximizing the similarity between variations of a synthetic sample. MP-SSL alternately updates the parameters for sampling synthetic unlabeled data by LMO and training model f_θ by SRC. The overview of MP-SSL is illustrated in Figure 2.

3.1. Latent Meta-Optimization (LMO)

The goal of LMO is to find a synthetic sample that approximates unlabeled data and contributes to the performance of f_θ through SSL. To extract unlabeled samples from G_F, we optimize the parameters ϕ and ξ that generate the latent variables $\hat{z} \in \mathcal{Z}_\mathrm{F}$ and foundation class label $\hat{y}_\mathrm{F} \in \mathcal{Y}_\mathrm{F}$, respectively. That is, we search for an optimal pair of $(\hat{z}, \hat{y}_\mathrm{F})$ through this optimization process. In conditional generative models, the latent variables control overall characteristics without class categories (e.g., size of object and style), and the class labels determine the category of the synthetic samples (Odena et al., 2017; Brock et al., 2019). Searching $(\hat{z}, \hat{y}_\mathrm{F})$ can be more reasonable than directly optimizing whole parameters of G_F on $\mathcal{D}$ because the latter suffers from overfitting and the low-performance of f_θ due to the low-quality samples (Karras et al., 2020). For the optimization, we use specialized architectures called *conditional mapper* $M_\phi : \mathcal{Z}_\mathrm{F} \times \mathcal{Y} \to \mathcal{Z}_\mathrm{F}$ and *label converter* $I_\xi : \mathcal{Y} \to \mathcal{Y}_\mathrm{F}$. Through optimizing M_ϕ and I_ξ, we seek a synthetic sample $\hat{x}_\mathrm{u} = G_\Phi(M_\phi(z_\mathrm{F}, y), I(y))$. To this end, we formalize the optimization problem of LMO as follows.

$$\min_{\phi,\xi} \mathcal{L}_\mathrm{val}(\theta^*) + \lambda_\mathrm{gap}\mathcal{L}_\mathrm{gap}(\phi, \xi) \tag{6}$$

$$\mathcal{L}_\mathrm{val}(\theta^*) = \mathbb{E}_{(x_\mathrm{val}, y_\mathrm{val}) \in \mathcal{D}_\mathrm{val}} \ell(f_{\theta^*}(x_\mathrm{val}), y_\mathrm{val}) \tag{7}$$

$$\mathcal{L}_\mathrm{gap}(\phi, \xi) = \mathbb{E}_{x \in \mathcal{D}} \| g_\psi(x) - g_\psi(\hat{x}_\mathrm{u} = G_\mathrm{F}(M_\phi(z, y), I_\xi(y))) \|_2^2 \tag{8}$$

$$\text{s.t.} \quad \theta^* = \arg\min_\theta \mathcal{L}(\theta) + \lambda \mathcal{L}_\mathrm{u}(\theta, \phi, \xi), \tag{9}$$

where $\mathcal{L}_\mathrm{val}$ is for seeking samples to improve f_θ and $\mathcal{L}_\mathrm{gap}$ is for satisfying that $\hat{x}_\mathrm{u}$ approximates training data x as unlabeled samples. This meta-optimization problem can be solved by stochastic gradient descent by extending the prior meta-learning method such as MAML (Finn et al., 2017). In the rest of this subsection, we describe the design of the conditional mapper and label converter.

Conditional Mapper M_ϕ. The role of M_ϕ is to find optimal latent variables producing useful unlabeled samples for SSL through G_F. Our idea is to transform the concatenation input latent variable z and training class label y into a new latent variable $\hat{z}$. This is based on an expectation that partitioning the problem for each class will make searching latent variables easier; we confirm that using y yields more performance gain in Sec. 4.5.2. M_ϕ outputs the estimated latent variable $\hat{z}$ by

$$\hat{z} = M_\phi(z, y) = \mathrm{MLP}_\phi(\mathrm{Concat}(z, \mathrm{EMB}_\phi(y))), \tag{10}$$

where $\mathrm{EMB}_\phi : \mathcal{Y} \to \mathbb{R}^{d_\mathcal{Y}}$ is an embedding layer for y, $\mathrm{Concat}(\cdot)$ is a concatenation operation of two vectors, and $\mathrm{MLP}_\phi : \mathbb{R}^{d_{\mathcal{Z}_\mathrm{F}} + d_\mathcal{Y}} \to \mathcal{Z}_\mathrm{F} = \mathbb{R}^{d_{\mathcal{Z}_\mathrm{F}}}$ is a multi-layer perception yielding a new latent variable.

Label Converter I_ξ. I_ξ estimates a foundation label $\hat{y}_\mathrm{F}$ corresponding to a training class label y. To estimate a foundation label, a prior work (Yamaguchi et al., 2022) utilizes a pre-trained classifier on foundation datasets. This approach is simple, but the pre-trained classifiers are not necessarily given, and the estimation of foundation soft labels depends on the performance of the pre-trained classifiers. Thus, if high-performance pre-trained classifiers are unavailable, it is hard to estimate a foundation label correctly. Instead of the

Algorithm 1 MP-SSL

Require: Training dataset $\mathcal{D}$, validation dataset $\mathcal{D}_{\text{val}}$ classifier f_θ, generative foundation model G_{F}, conditional mapper M_ϕ, label converter I_ξ, training batchsize B, validation batchsize B_{val}, step size η and ξ, hyperparameter λ

Ensure: Trained classifier f_θ

while not converged **do**
  $\{(x^i, y^i)\}_{i=1}^{B} \sim \mathcal{D}$
  $\{z^i\}_{i=1}^{B} \sim \mathcal{N}(0, I)$
  // Updating ϕ and ξ by LMO
  $\{(x^i_{\text{val}}, y^i_{\text{val}})\}_{i=1}^{B_{\text{val}}} \sim \mathcal{D}$
  $\{\hat{x}^i_{\text{u}}\}_{i=1}^{B} = \{G_{\text{F}}(M_\phi(z^i, y^i), I_\xi(y^i)\}_{i=1}^{B}$
  $\theta' \leftarrow \theta - \eta\nabla_\theta(\frac{1}{B}\sum_{i=1}^{B}\ell(f_\theta(x^i), y^i) + \frac{\lambda}{B}\sum_{i=1}^{B}\ell_{\text{SCR}}(\hat{x}^i_{\text{u}}; \psi))$
  $\phi \leftarrow \phi - \xi\nabla_\phi(\frac{1}{B_{\text{val}}}\ell(f_{\theta'}(x_{\text{val}}), y_{\text{val}}) + \|\frac{1}{B}\sum_{i=1}^{B} f_\theta(x^i) - \frac{1}{B}\sum_{i=1}^{B} f_\theta(\hat{x}^i_{\text{u}})\|_2^2)$
  // Updating θ with SCR
  $\{\hat{x}^i_{\text{u}}\}_{i=1}^{B} = \{G_\Phi(F_\phi(z^i), y^i_{\text{p}})\}_{i=1}^{B}$
  $\theta \leftarrow \theta - \eta\nabla_\theta(\frac{1}{B}\sum_{i=1}^{B}\ell(f_\theta(x^i), y^i) + \frac{\lambda}{B}\sum_{i=1}^{B}\ell_{\text{SCR}}(\hat{x}^i_{\text{u}}; \psi))$
end while

pre-trained classifiers, we utilize the Gumbel-softmax (Jang et al., 2017) trick for sampling $\hat{y}_{\text{F}}$ through the parameter ξ updated by LMO:

$$\hat{y}_{\text{F}} = \arg\max_i \; I_\xi(y), \tag{11}$$

$$I_\xi(y)[i] = \frac{\exp\left((\log(\text{EMB}_\xi[i]) + \mathbf{g}[i])/\tau\right)}{\sum_{j=1}^{|\mathcal{Y}_{\text{F}}|}\exp\left((\log(\text{EMB}_\xi[j]) + \mathbf{g}[j])/\tau\right)}, \tag{12}$$

where $\text{EMB}_\xi : \mathcal{Y} \to \mathbb{R}^{d_\mathcal{Y}}$ is an embedding layer for y, $\mathbf{g}[i] = -\log(-\log(u_i \sim \text{Uniform}(0, 1)))$, τ is a temperature parameter. This formulation has several advantages: (a) it can be trained by backpropagation since it is fully differentiable, (b) the output $\hat{x}_{\text{u}}$ is expected to be unbiased due to randomness given by $\mathbf{g}$, and (c) the number of foundation classes of interest can be adjustable according to the training data by the temperature parameters. We confirm these advantages through comparison to the other variants of I_ξ in Sec. 4.5.3.

3.2. Synthetic Consistency Regularization

Although synthetic samples generated from G_{F} through LMO can contain useful information for training f_θ, it is hard to expect that they are exactly categorized to the training space $\mathcal{Y}$ because the training and foundation label spaces are not the same, i.e., $\mathcal{Y} \neq \mathcal{Y}_{\text{F}}$. Therefore, training with the synthetic samples via unsupervised losses using pseudo training labels in $\mathcal{Y}$ (e.g., FixMatch (Sohn et al., 2020)) might confuse f_θ due to the label space mismatch. To avoid the negative effect and maximize the gain from the synthetic samples, we introduce a simple unsupervised loss called synthetic consistency regularization (SCR). In contrast to existing pseudo-label based SSL methods, SCR is computed on the feature extractor g_ψ of f_θ. That is, we regularize g_ψ by synthetic samples instead of the classifier head h_ω. To regularize g_ψ, we design SCR based on consistency regularization (Bachman et al., 2014; Xie et al., 2020; Sohn et al., 2020), which minimizes the gap between the outputs of two variants of samples that are transformed by different data augmentations. Concretely, we

formalize the loss function of SCR as follows.

$$\ell_{\mathrm{SCR}}(\hat{x}_{\mathrm{u}};\psi) \quad = \quad 1 - \frac{g_\psi(T_{\mathrm{w}}(\hat{x}_{\mathrm{u}})) \cdot g_\psi(T_{\mathrm{s}}(\hat{x}_{\mathrm{u}}))}{\|g_\psi(T_{\mathrm{w}}(\hat{x}_{\mathrm{u}}))\|\|g_\psi(T_{\mathrm{s}}(\hat{x}_{\mathrm{u}}))\|}, \tag{13}$$

where $T_{\mathrm{w}}(\cdot)$ and $T_{\mathrm{s}}(\cdot)$ are a weak augmentation (e.g., flip and crop) and a strong augmentation (e.g., RandAugment (Cubuk et al., 2020)). As the measurement of the gap, we choose cosine distance; we empirically found that this formulation achieves the best results when comparing with L2, L1, and smooth L1 distance as shown in Sec. 4.5.4. By applying SCR to g_ψ, we expect that g_ψ learns robust feature representations that are useful for classifying real samples by h_ω.

Finally, we show the overall procedure of MP-SSL using LMO and SCR in Algorithm 1.

4. Experiments

This section evaluates our MP-SSL through experiments on multiple image classification datasets. We mainly aim to answer three research questions with the experiments: (1) Can MP-SSL improve the baselines without real unlabeled datasets? (2) What can training models learn through MP-SSL? (3) Is the MP-SSL design reasonable? We compare MP-SSL with baselines with synthetic samples, e.g., P-SSL (Yamaguchi et al., 2022), and baselines with real samples e.g., FreeMatch (Wang et al., 2023) in Sec. 4.2 and 4.3. Furthermore, we provide a detailed analysis of MP-SSL, such as the visualization of synthetic samples (Sec. 4.4) and detailed ablation studies of MP-SSL (Sec. 4.5).

4.1. Setting

Baselines. We compare our method with the following baselines in the gSSL setting. **Base Model**: training f_θ with only $\mathcal{D}$. **Naïve gSSL**: training f_θ with $\mathcal{D}$ and G_{F}, where a synthetic sample $\hat{x_{\mathrm{u}}}$ is generated from uniformly sampled z and y_{F}, then we train f_θ by an existing SSL method with the real and synthetic samples. **P-SSL** (Yamaguchi et al., 2022): training f_θ with $\mathcal{D}$ and G_{F} with sampling y_{F} by Eq. (5) and existing SSL methods updating h_ω. We also test SSL methods using a real unlabeled dataset $\mathcal{D}_{\mathrm{u}}$ to assess the practicality of the gSSL setting; We refer this setting to oracle SSL because they can access $\mathcal{D}_{\mathrm{u}}$ that is prohibited in gSSL. As the oracle SSL methods, We used three representative SSL methods: UDA (Xie et al., 2020), FixMatch (Sohn et al., 2020), and FreeMatch (Wang et al., 2023).

Datasets. We used six image datasets for classification tasks: Cars (Krause et al., 2013), Aircraft (Maji et al., 2013), Birds (Welinder et al., 2010), DTD (Cimpoi et al., 2014), Flowers (Nilsback and Zisserman, 2008), and Pets (Parkhi et al., 2012). To evaluate both generative and oracle SSL settings at the same time, we randomly split them into $\mathcal{D}$ and $\mathcal{D}_{\mathrm{u}}$ (50 : 50, by default), and discarded $\mathcal{D}_{\mathrm{u}}$ in gSSL and used in oracle SSL. Furthermore, to evaluate the effect of dataset size, we varied the size of labeled datasets of Cars by $\{10, 25, 50, 100\}\%$ in volume. Note that we used all of the rest of the unlabeled samples as $\mathcal{D}_{\mathrm{u}}$ in this setting. After creating $\mathcal{D}$, we randomly split $\mathcal{D}$ into 9 : 1 and used the former as $\mathcal{D}$ and the latter as $\mathcal{D}_{\mathrm{val}}$ in the training.

Table 1: Performance comparison of ResNet-18 classifiers on multiple datasets (Top-1 Acc. (%)). Underlined scores are the best of the oracle SSL setting (i.e., using real unlabeled datasets), and **Bolded scores** are the best among the methods of the generative SSL (gSSL) setting (i.e., using foundation generative models).

Method / Dataset	Aircraft	Birds	Cars	DTD	Flower	Pets
Base Model	$44.05^{\pm.59}$	$60.74^{\pm.29}$	$71.62^{\pm.30}$	$61.56^{\pm.56}$	$88.14^{\pm.18}$	$84.44^{\pm.48}$
Oracle SSL ($\mathcal{D} + \mathcal{D}_\mathrm{u}$)						
UDA (Xie et al., 2020)	$44.65^{\pm.38}$	$60.22^{\pm.03}$	$60.22^{\pm.03}$	$70.90^{\pm.58}$	$61.90^{\pm.10}$	$\underline{87.72}^{\pm.31}$
FixMatch (Sohn et al., 2020)	$47.89^{\pm.38}$	$60.58^{\pm.84}$	$80.98^{\pm.36}$	$61.31^{\pm.11}$	$\underline{90.08}^{\pm.48}$	$81.73^{\pm.39}$
FreeMatch (Wang et al., 2023)	$\underline{49.55}^{\pm.33}$	$\underline{66.09}^{\pm.16}$	$\underline{82.73}^{\pm.41}$	$\underline{63.83}^{\pm.49}$	$90.07^{\pm.27}$	$86.61^{\pm.40}$
Generative SSL ($\mathcal{D} + G_\mathrm{F}$)						
Naïve gSSL (FreeMatch)	$46.83^{\pm.34}$	$60.95^{\pm.29}$	$73.67^{\pm.67}$	$59.41^{\pm.17}$	$86.41^{\pm.25}$	$83.66^{\pm.69}$
P-SSL (Yamaguchi et al., 2022)	$45.43^{\pm.24}$	$60.54^{\pm.25}$	$72.45^{\pm.30}$	$60.82^{\pm.61}$	$88.20^{\pm.15}$	$84.84^{\pm.41}$
MP-SSL (Ours)	$\mathbf{49.48}^{\pm.25}$	$\mathbf{62.86}^{\pm.23}$	$\mathbf{76.33}^{\pm.31}$	$\mathbf{62.34}^{\pm.46}$	$\mathbf{88.44}^{\pm.51}$	$\mathbf{85.43}^{\pm.09}$

Architectures. We used ResNet-18 (He et al., 2016) as f_θ and BigGAN for 256×256 images (Brock et al., 2019) as G_F. M_ϕ was composed of a three-layer perceptron with a leaky-ReLU activation function. We used the ImageNet pre-trained weights of ResNet-18 distributed by PyTorch.[2] For BigGAN, we used the ImageNet pre-trained weights provided by Brock et al. (2019). Note that we used the same G_F in the baselines and our method.

Training. We trained f_θ by the Nesterov momentum SGD for 200 epochs with a momentum of 0.9 and an initial learning rate of 0.01; we decayed the learning rate by 0.1 at 60, 120, and 160 epochs. We trained M_ϕ and I_ξ by the Adam optimizer for 200 epochs with a learning rate of 1.0×10^{-4}. We used mini-batch sizes of 64. The input samples were resized into a resolution of 224×224; $\hat{x}_\mathrm{u}$ was resized by differentiable transformations. For synthetic samples from G_F in MP-SSL, the weak transformation T_w was horizontal flip and random crop, and the strong transformation T_s was RandAugment (Cubuk et al., 2020) by following Xie et al. (2020); it was implemented with differentiable transformations provided in Kornia (Riba et al., 2020). We determined the hyperparameter λ by grid search among $[0.1, 1.0]$ with a step size of 0.1 for each method by $\mathcal{D}_\mathrm{val}$. We used λ_gap of 10. For the hyperparameters of oracle SSL methods, we followed the default settings of the original papers (Xie et al., 2020; Sohn et al., 2020; Wang et al., 2023). We selected the final model by checking the validation accuracy for each epoch. We ran the experiments three times on a 24-core Intel Xeon CPU with an NVIDIA A100 GPU with 40GB VRAM and recorded average test accuracies with standard deviations evaluated on the final models.

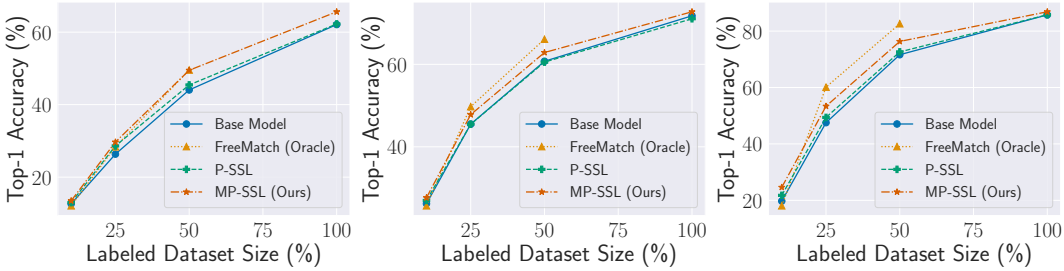
4.2. Evaluation on Multiple Datasets

First, we evaluate our MP-SSL's performance by comparing it with the baseline methods of gSSL and oracle SSL on various training datasets. Table 1 shows the results on six datasets. Note that we did not use the unlabeled dataset $\mathcal{D}_\mathrm{u}$ in the gSSL setting. Our MP-SSL achieved the best results among the gSSL methods with a large margin (up to 3pp). While P-SSL degraded the base model on DTD due to the mismatch between training and foundation label spaces (Yamaguchi et al., 2022), our MP-SSL succeeded in improving it.

2. https://github.com/pytorch/vision

Table 2: Performance comparison of ResNet-18 classifiers on the reduced Cars datasets (Top-1 Acc. (%)). Underlined scores are the best of the oracle SSL setting (i.e., using real unlabeled datasets), and **Bolded scores** are the best among the methods of the gSSL setting (i.e., using foundation generative models).

Method / Labeled Dataset Size	10%	25%	50%	100%
Base Model	$19.74^{\pm.15}$	$47.54^{\pm.67}$	$71.62^{\pm.30}$	$85.75^{\pm.08}$
Oracle SSL ($\mathcal{D} + \mathcal{D}_\mathrm{u}$)				
UDA (Xie et al., 2020)	$19.36^{\pm.44}$	$47.95^{\pm.30}$	$72.76^{\pm.53}$	N/A
FixMatch (Sohn et al., 2020)	$\underline{20.98}^{\pm.99}$	$\underline{63.58}^{\pm.64}$	$\underline{83.94}^{\pm.65}$	N/A
FreeMatch (Wang et al., 2023)	$18.07^{\pm.03}$	$60.13^{\pm.61}$	$82.60^{\pm.28}$	N/A
Generative SSL ($\mathcal{D} + G_\mathrm{F}$)				
Naïve gSSL (FreeMatch)	$20.11^{\pm.03}$	$49.33^{\pm.54}$	$72.91^{\pm.38}$	$81.68^{\pm.18}$
P-SSL (Yamaguchi et al., 2022)	$20.34^{\pm.42}$	$48.27^{\pm.48}$	$72.62^{\pm.33}$	$85.78^{\pm.23}$
MP-SSL (Ours)	$\mathbf{23.82}^{\pm.55}$	$\mathbf{53.37}^{\pm.56}$	$\mathbf{76.33}^{\pm.31}$	$\mathbf{86.84}^{\pm.10}$



(a) Aircraft (b) Bird (c) Cars

Figure 3: Performance Comparisons in Small Labeled Dataset (ResNet-18)

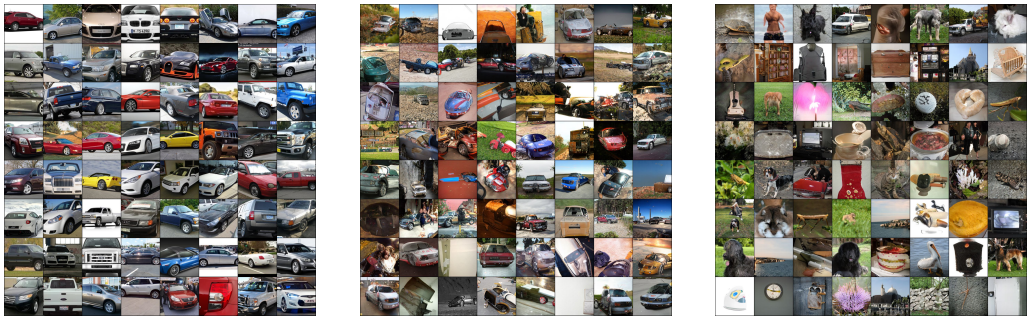
This indicates that MP-SSL is not sensitive to the label space mismatch and stably improves classifiers in various settings. Furthermore, on the Aircraft and DTD datasets, MP-SSL is competitive with the oracle SSL methods. This suggests that MP-SSL and gSSL have the potential to approximate the oracle SSL methods in terms of the final model accuracy.

4.3. Evaluation by Varying Dataset Size

We evaluate MP-SSL by varying the size of training labeled datasets. We used all of the remaining unlabeled samples as $\mathcal{D}_\mathrm{u}$ for the oracle SSL methods and did not use $\mathcal{D}_\mathrm{u}$ for the gSSL methods. Table 2 shows that our MP-SSL achieves the best results in the gSSL setting for all dataset sizes. More interestingly, MP-SSL significantly outperformed the best result of the oracle SSL methods when the labeled dataset is extremely small (i.e., 10% $\leq$ 1,000 samples). This trend is consistent with multiple datasets, as shown in Fig. 3. These results suggest that the synthetic samples from G_F are more valuable than real unlabeled samples for improving classification performance when the labeled datasets are quite small.

4.4. Analysis of Synthetic Samples

We examine what the classifier is learning through MP-SSL. To this end, we visualize the synthetic samples generated by MP-SSL and compare them to real and synthetic samples



(a) Real (b) P-SSL (c) MP-SSL (Ours)

Figure 4: Real and Synthetic Samples in Training (Cars)

Table 3: Analysis of LMO

Pattern	Cars Test Acc.(%)
Base Model	$71.62^{\pm.30}$
MP-SSL w/o LMO	$74.34^{\pm.01}$
MP-SSL w/o $\mathcal{L}_{\text{gap}}(\phi, \xi)$	$75.46^{\pm.22}$
MP-SSL w/o $\mathcal{L}_{\text{val}}(\theta^*)$	$75.63^{\pm.21}$
MP-SSL	$\mathbf{76.33}^{\pm.31}$

Table 4: Ablation Study of M_ϕ

Pattern	Cars Test Acc.(%)
Base Model	$71.62^{\pm.30}$
Unconditional M_ϕ	$75.55^{\pm.40}$
Conditional M_ϕ	$\mathbf{76.33}^{\pm.31}$

generated by P-SSL. Figure 4 shows the real and synthetic samples. Interestingly, we see that P-SSL produces more relative samples to real samples (Cars), whereas MP-SSL produces not so relative but diverse samples. Since the performance studies in Sec. 4.2 and 4.3 show that MP-SSL completely outperformed P-SSL, this visualization result is contrary to intuition. We consider that this can be caused by the unsupervised regularization of MP-SSL, which penalizes the feature extractor instead of the entire model. As defined in Eq. (6), LMO of MP-SSL optimizes the latent vectors through the backpropagation from the unsupervised loss, and thus, the synthetic samples generated from the latent vectors are not optimized to become similar to real samples in its label spaces. The results suggest that the regularization of feature extractors does not necessarily require perfect imitation of the training data, and the diversity of samples is more important.

4.5. Ablation Study

4.5.1. Meta-Learning and Gap Loss in LMO

We evaluate the effectiveness of LMO by decomposing the objective function defined in Eq. (6). Eq. (6) is composed of the meta-learning loss $\mathcal{L}_{\text{val}}(\theta^*)$ and the feature gap loss $\mathcal{L}_{\text{gap}}(\phi, \xi)$. Table 3 shows the impact of these components on accuracy by ablating them in MP-SSL. The row of MP-SSL w/o LMO denotes the test pattern of discarding LMO from MP-SSL, i.e., producing $\hat{x}_{\text{u}}$ by random sampling from G_{F}. From the results, we confirm that $\mathcal{L}_{\text{val}}(\theta^*)$ and $\mathcal{L}_{\text{gap}}(\phi, \xi)$ equally contribute to the test performance. In other words, the meta-learning loss and the feature gap loss have different effects on the synthetic samples and are complementary.

Table 5: Ablation Study of I_ξ

Output Module	Cars Test Acc.(%)
Soft Label by EMB_ξ	$75.55^{\pm.24}$
Soft Gumbel Softmax	$75.72^{\pm.41}$
Hard Gumbel Softmax ($\tau = 1.0 \times 10^{-1}$)	$75.85^{\pm.31}$
Hard Gumbel Softmax ($\tau = 1.0 \times 10^{-3}$)	$75.87^{\pm.33}$
Hard Gumbel Softmax ($\tau = 1.0 \times 10^{-5}$)	$\mathbf{76.33}^{\pm.31}$
Hard Gumbel Softmax ($\tau = 1.0 \times 10^{-7}$)	$76.02^{\pm.50}$

Table 6: Comparison of ℓ_u for MP-SSL

ℓ_u	Cars Test Acc.(%)
FreeMatch	$73.32^{\pm.40}$
L1 Distance	$73.80^{\pm.73}$
L2 Distance	$74.71^{\pm.86}$
Smooth L1 Distance	$74.67^{\pm.60}$
SCR (Eq. (13))	$\mathbf{76.33}^{\pm.31}$

4.5.2. Conditional Mapper

We assess the design validity of conditional mapper $M_\phi(z, y)$. In Eq. (10), we define M_ϕ to be conditioned by a training class label y. To confirm the effectiveness of using labels, we tested unconditional mapper $M_\phi(z)$, which is created by discarding the components for labels from $M_\phi(z, y)$. Table 4 summarises the results. MP-SSL with a conditional mapper significantly outperformed one with an unconditional mapper. Therefore, we can say that using conditional labels for transforming a latent variable z helps boost models' performance.

4.5.3. Label Converter

In Sec. 3.1, we design label converter I_ξ composed of the Gumbel softmax module as Eq. (11). This section provides the ablation study to evaluate the design choice. We varied the implementation of I_ξ with (a) soft label by embedding layer, i.e., $\hat{y}_\text{F} = \text{EMB}_\xi$, (b) soft Gumbel softmax, i.e., $\hat{y}_\text{F} = I_\xi$. Furthermore, we varied the hyperparameter τ in Eq. (11). Table 5 shows the results. Using the Gumbel softmax with hard label output brings better test accuracy. This indicates that using the soft label output might not be appropriate for the unsupervised regularization loss since it results in ambiguous and low-quality output as in P-SSL, which uses soft labels for generating synthetic samples (Figure 4b).

4.5.4. Synthetic Consistency Regularization

We lastly provide an ablation study of SCR defined by a cosine distance form as Eq. (13). We tested four variants of ℓ_u in MP-SSL: (a) FreeMatch (Wang et al., 2023) that updates the entire model f_θ including the classifier head h_ω, (b) L1 distance, i.e., $|g_\psi(T_\text{W}(\hat{x}_\text{u})) - g_\psi(T_\text{S}(\hat{x}_\text{u}))|$, (c) L2 distance, i.e., $\|g_\psi(T_\text{W}(\hat{x}_\text{u})) - g_\psi(T_\text{S}(\hat{x}_\text{u}))\|_2^2$, (d) Smooth L1 distance (Girshick, 2015). We list the results in Table 6. First, we see that our SCR loss significantly outperforms the FreeMatch loss. This means that the consistency regularization on the feature spaces is quite effective for gSSL, as we expected in Sec. 3.2. Second, among the variants of SCR, the cosine distance based loss function achieved the best results. We conjecture that losses that directly minimize differences between feature vectors, such as L1 and L2 distance, involve the L1 and L2 norm of the feature vector. Therefore, the norm of the feature vectors during training is relatively smaller, which hurts the norm of the loss gradients of classification tasks (Hariharan and Girshick, 2017).

5. Related Work

Semi-supervised Learning. Semi-supervised Learning (SSL) is a paradigm that trains a supervised model with labeled and unlabeled samples by simultaneously minimizing supervised and unsupervised loss. Historically, various SSL algorithms have been proposed for deep learning such as entropy minimization (Grandvalet and Bengio, 2005), pseudo-label (Lee et al., 2013), and consistency regularization (Bachman et al., 2014; Sajjadi et al., 2016; Laine and Aila, 2016). UDA (Xie et al., 2020) and FixMatch (Sohn et al., 2020), which combine ideas of pseudo-label and consistency regularization, have achieved remarkable performance. More recent methods such as FreeMatch (Wang et al., 2023) improve UDA and FixMatch to adaptively control the confidence threshold of acceptance of the pseudo labels for preventing error accumulation and overfitting. These SSL algorithms assume that many unlabeled data are provided because unlabeled samples can be more easily obtained than labeled samples with human annotations. However, we point out that even unlabelled data is becoming more difficult in today's increasingly privacy-conscious world. This paper opens up a new SSL paradigm that makes unlabelled data unnecessary by leveraging pre-trained generative foundation models.

Leveraging Generative Models for Training Discriminative Models. In the context of data augmentation and transfer learning, several studies have applied the expressive power of conditional generative models to boost the performance of discriminative models, e.g., classifiers. Zhu et al. (2018), Yamaguchi et al. (2020), Yamaguchi et al. (2023), and He et al. (2023) have exploited the generated images from conditional GANs and diffusion models for data augmentation and representation learning, and Sankaranarayanan et al. (2018) have introduced conditional GANs for domain adaptation setting to learn feature spaces of source and target domains jointly. Li et al. (2020) have implemented an unsupervised domain adaptation technique with conditional GANs in a setting of no accessing source datasets. More similar to our work, Yamaguchi et al. (2022) have proposed a transfer learning method called P-SSL using pre-trained generative foundation models in semi-supervised learning. However, we note that P-SSL and our method differ three-fold: (a) problem setting, (b) assumptions of data and label spaces, and (c) optimization methods. For (a), the problem setting of our method is focused on SSL, whereas P-SSL is for transfer learning, where the neural architectures of source and target classifiers are different. For (b), our method assumes the generative foundation model G_{F} covers the training data space $\mathcal{X}$. In contrast, P-SSL assumes the label space of G_{F} covers the training label space i.e., $\mathcal{Y} \subset \mathcal{Y}_{\mathrm{F}}$; the latter is more strict and thus the performance might degrade when it does not hold (Yamaguchi et al., 2022). For (c), we directly optimize the latent variables of G_{F} to find optimal unlabeled samples for SSL, whereas P-SSL just samples related synthetic samples via similarity in the label spaces through source pre-trained classifier. These differences produce the performance improvements of our method in SSL, as shown in Sec. 4.

6. Conclusion

This paper presents a new semi-supervised learning (SSL) problem setting called generative SSL, where real unlabeled datasets are unavailable, where a generative foundation model is given as the source of unlabeled data. This setting is important because we are often

restricted from obtaining real unlabeled data due to privacy concerns. To solve this problem, we propose a training method called MP-SSL, which consists of latent meta-optimization (LMO) and synthetic consistency regularization (SCR). We experimentally demonstrate that our MP-SSL outperforms existing baselines and can potentially replace real unlabeled datasets with generative foundation models. One of the limitations of this work is the dependency on the existence of foundation generative models, but this limitation will be relaxed because the foundation model trend is rapidly developing for various modalities in the community. Important future steps are to speed up or avoid the computation of meta-learning in LMO and to extend our method to diffusion models, which produce synthetic samples with higher fidelity but require higher computational costs for sampling than GANs.

References

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in neural information processing systems*, 2014.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-supervised Learning*. MIT Press, 2006.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 2017.

Ross Girshick. Fast r-cnn. In *International Conference on Computer Vision*, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, 2014.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2005.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE international conference on computer vision*, pages 3018–3027, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? 2023.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representation*, 2017.

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

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition*, Sydney, Australia, 2013.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2016.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Zinan Lin, Vyas Sekar, and Giulia Fanti. On the privacy properties of gan-generated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 1522–1530. PMLR, 2021.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv*, 2013.

M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, 2017.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, 2016.

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

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

Boris van Breugel, Zhaozhi Qian, and Mihaela van der Schaar. Synthetic data, real errors: how (not) to publish and use synthetic data. In *International Conference on Machine Learning*, 2023.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations*, 2023.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020.

Shin'ya Yamaguchi, Sekitoshi Kanai, and Takeharu Eda. Effective data augmentation with multi-domain learning gans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Shin'ya Yamaguchi, Sekitoshi Kanai, Atsutoshi Kumagai, Daiki Chijiwa, and Hisashi Kashima. Transfer learning with pre-trained conditional generative models. *arXiv preprint arXiv:2204.12833*, 2022.

Shin'ya Yamaguchi, Sekitoshi Kanai, Atsutoshi Kumagai, Daiki Chijiwa, and Hisashi Kashima. Regularizing neural networks with meta-learning generative models. In *Advances in Neural Information Processing Systems*, 2023.

Yezi Zhu, Marc Aoun, Marcel Krijn, and Joaquin Vanschoren. Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. In *British Machine Vision Conference*, 2018.