Curriculum Fine-tuning of Vision Foundation Model for Medical Image Classification Under Label Noise

Yeonguk Yu **Minhwan Ko** **Sungho Shin** **Kangmin Kim** **Kyoobin Lee***
Gwangju Institute of Science and Technology
{yeon_guk, mhko1998, hogili89, kgmin156}@gm.gist.ac.kr kyoobinlee@gist.ac.kr

Abstract

Deep neural networks have demonstrated remarkable performance in various vision tasks, but their success heavily depends on the quality of the training data. Noisy labels are a critical issue in medical datasets and can significantly degrade model performance. Previous clean sample selection methods have not utilized the well pre-trained features of vision foundation models (VFMs) and assumed that training begins from scratch. In this paper, we propose CUFIT, a curriculum fine-tuning paradigm of VFMs for medical image classification under label noise. Our method is motivated by the fact that linear probing of VFMs is relatively unaffected by noisy samples, as it does not update the feature extractor of the VFM, thus robustly classifying the training samples. Subsequently, curriculum fine-tuning of two adapters is conducted, starting with clean sample selection from the linear probing phase. Our experimental results demonstrate that CUFIT outperforms previous methods across various medical image benchmarks. Specifically, our method surpasses previous baselines by 5.0%, 2.1%, 4.6%, and 5.8% at a 40% noise rate on the HAM10000, APTOS-2019, BloodMnist, and OrgancMnist datasets, respectively. Furthermore, we provide extensive analyses to demonstrate the impact of our method on noisy label detection. For instance, our method shows higher label precision and recall compared to previous approaches. Our work highlights the potential of leveraging VFMs in medical image classification under challenging conditions of noisy labels.

1 Introduction

Deep neural networks have demonstrated remarkable performance across various tasks, including classification, detection, and segmentation [20, 16, 19, 57]. In medical imaging, these neural networks leverage large amounts of labeled data to train models capable of accurately detecting or classifying medical conditions from images such as dermatoscopes, X-rays, MRIs, and CT scans. However, in practical settings, data often contain noisy labels and it is well established that neural networks perform well only when the quality of training data is sufficiently high [42, 29, 5]. Noisy labels occur when the data annotations—the labels assigned to training images—are incorrect or inconsistent. This issue is particularly problematic in medical imaging, where annotating images is more complex compared to natural images [53, 25]. Consequently, improving the robustness of neural networks against noisy labels is a crucial area of research, directly affecting the effectiveness and reliability of medical imaging technologies.

A large number of algorithms have been developed to address the issue of performance degradation caused by noisy samples [42]. In particular, clean sample selection methods, such as MentorNet [24], Co-teaching [17], Co-teaching+ [56], JoCor [47], and CoDis [50], have demonstrated superior performance without requiring modifications to the model architecture or training loss. The core

*Corresponding author: Kyoobin Lee. Our code is available at github.com/gist-ailab/CUFIT.

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

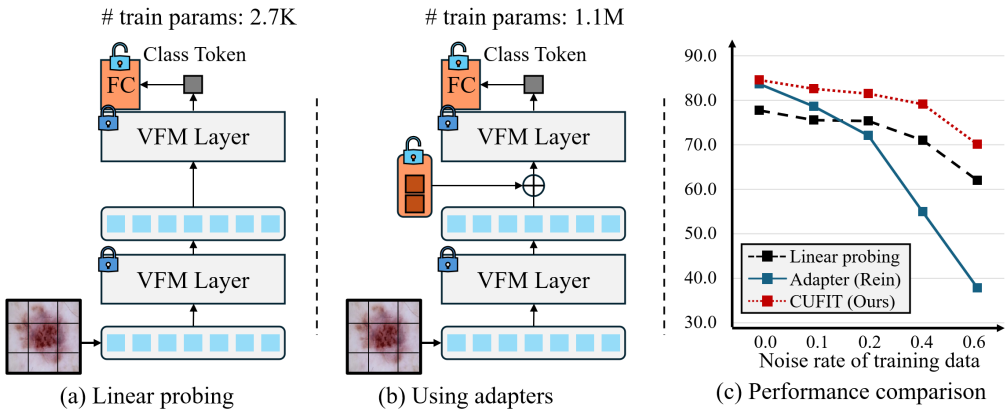


Figure 1: Illustration of linear probing (a) and adapter usage (b). Specifically, the weights of the foundation model are frozen, while the fully connected layer or adapter weights (shown in orange) are updated during the training phase. In (c), a performance comparison using a simulated noisy dataset (HAM10000) is presented. It demonstrates that linear probing is more robust to noisy labels compared to the adapter, whereas the adapter outperforms linear probing when there are no noisy labels.

principle behind these methods is that small-loss samples are likely to be clean, as they are easier to classify and the model memorizes them faster. Additionally, using two different but homogeneous networks to select small-loss samples for each other is more stable than relying on a single model for sample selection. These methods have shown outstanding performance using traditional neural network architectures based on convolutional layers. However, there are practical issues with these methods in two aspects: (i) it cannot be guaranteed that these methods will perform equally well with transformer-based architectures, which have recently gained significant attention, and (ii) their assumption that training starts from scratch is impractical, as it prevents the use of rich features from pre-trained models, which could be beneficial for filtering noisy labels.

Recently, large-scale vision foundation models (VFMs) with transformer-based architectures [13], such as CLIP [39], MAE [18], SAM [28], and DINOv2 [37], have gained attention for their performance and applicability across various tasks. The self-supervised training of VFMs on large-scale datasets enhances their robustness against various image corruptions and improves their generalization capabilities [10, 48, 38, 40]. The inherent robustness and rich features of VFMs can be beneficial for detecting noisy labels. For instance, linear probing of VFMs is relatively unaffected by noisy samples since it does not modify the VFM's feature extractor, preventing the memorization of noisy data, as shown in Figure 1. However, linear probing does not fully leverage the VFM's capabilities when there is a domain gap between the pretraining task of the VFM and the target task (e.g., pretraining on natural images versus medical image classification) [48]. To address this issue, some researchers have proposed using trainable fine-tuning adapters for VFMs [22, 23, 48, 9]. Yet, these adapters might degrade performance by memorizing noisy labels due to their trainable parameters involved in feature extraction. Therefore, we state our research question as follows: *How can we use the power of pre-trained vision foundation models for medical image classification in the presence of noisy labels?*

In this paper, we introduce a **Cu**rriculum **Fi**ne-**T**uning paradigm for vision foundation models in medical image classification under noisy labels, called **CUFIT**. CUFIT is a curriculum-learning framework designed to fine-tune VFMs with noisy medical datasets. The framework consists of three training modules: the Linear Probing Module (LPM), the Intermediate Adapter Module (IAM), and the Last Adapter Module (LAM). During the training stage, clean samples are selected based on an *agreement* criterion, where a sample is selected if its annotation matches the module's prediction. Specifically, the LPM is trained using all available samples, as linear probing is robust against noisy labels. Subsequently, the IAM is trained with the samples selected by the LPM, and the LAM is trained with the samples selected by the IAM. This inter-module curriculum training (i.e., LPM→IAM→LAM) is beneficial for increasing the number of clean samples available to train the LAM, considering that the LPM only selects a limited number of samples due to performance degradation caused by the domain gap. Consequently, CUFIT leverages the LAM for final predictions,

offering strong fine-tuning performance for medical image classification in the presence of noisy labels by utilizing the strengths of both linear probing and adapters, as illustrated in Figure 1-c. In summary, the main contributions of this paper are as follows:

- We introduce CUFIT, a simple yet effective fine-tuning paradigm for medical image classification using VFMs in the presence of noisy labels. This method leverages the robustness of linear probing and the generalization capability of fine-tuning adapters to handle noisy datasets during training stage.
- We conduct various experiments demonstrating that CUFIT significantly improves the robustness of VFMs against noisy labels in medical datasets. We show that CUFIT outperforms previous methods across several medical image benchmarks.
- We provide extensive analyses to enhance the understanding of our fine-tuning paradigm. Additionally, we validate our framework with various VFMs and adapter configurations.

2 Related Work

Vision foundation models. Vision transformers (ViTs) embed 2D images into 1D tokens and model their global correlations using the self-attention mechanism [13, 35, 43]. ViTs are known to be effective when large datasets are used, and the concept of vision foundation models is introduced. Several studies have developed pre-trained vision transformers based on self-supervised learning. For instance, contrastive language-image pre-training (CLIP) provides high-quality visual representations through contrastive learning with a large amount of image-text pairs [39]. Additionally, masked auto-encoder (MAE) offers high-capacity models that generalize well through self-supervised learning with masked auto-encoding, where the task is to reconstruct token patches from the given masked tokens [18]. Moreover, knowledge distillation with no labels (DINOv1) [7] proposed a teacher-student framework for self-training without annotations, resulting in a well-generalized ViT. More recently, DINOv2 introduced a self-training framework that combines masked autoencoding and teacher-student training based on carefully curated datasets [37].

Since large models and self-training in VFMs provide strong generalization capabilities for various tasks, parameter-efficient fine-tuning has gained attention. Parameter-efficient fine-tuning (PEFT) aims to adapt foundation models to new tasks by training only a few adapter parameters while keeping the model itself frozen. Notably, there is research that proposes low-rank adaptation (LoRA), which introduces trainable rank decomposition matrices into each layer of the transformer architecture in large language models [22]. For the VFMs, visual prompt tuning [23] proposed appending prompts to the input sequence of each transformer block, achieving excellent fine-tuning performance with minimal parameters. Similarly, Adaptformer [9] introduces a novel MLP block to replace the original one in transformer blocks, allowing for the use of both original and few trainable parameters. More recently, Wei *et al.* proposed Rein, which aims to adapt VFMs for semantic segmentation with domain generalization capabilities [48]. Also, Dutt *et al.* investigate PEFT algorithms across both convolutional and large transformer-based networks for medical image classification, demonstrating the effectiveness of PEFT, particularly in the low-data regimes common in medical imaging [14]. In this paper, we focus on fine-tuning VFMs for image classification in the presence of noisy labels using adapters. Rather than introducing a new adapter, we utilize an existing adapter within our training paradigm.

Learning with noisy label. Deep neural networks have demonstrated remarkable performance on large-scale datasets. However, it is well-known that neural networks can easily memorize noisy labeled samples, leading to degraded performance. Several studies have been conducted to explore robust supervised learning in the presence of noisy labels. These studies can be categorized into five approaches [42]: (i) robust architectures, (ii) robust regularization, (iii) robust loss functions [15, 33, 46, 59], (iv) loss adjustment [26, 31, 32], and (v) sample selection [24, 36, 17, 56, 50]. In this paper, we categorize our method as a sample selection method, which selects samples with clean labels from a noisy training dataset. While previous sample selection methods typically consider training from scratch, we focus on training starting from a pre-trained model, which is known to be more robust to noisy labels [21]. Additionally, research has explored using CLIP to enable robust training by leveraging its text-image matching capability on noisy datasets [49].

Various methods have been proposed for clean sample selection from noisy datasets. For example, MentorNet introduced the use of a teacher network to guide the student network to focus on clean labels [24]. Similarly, Decoupling proposed updating two networks by using only the samples with differing predictions between them [36]. Co-teaching also trains two networks simultaneously, updating them based on sample recommendations from each other [17]. Co-teaching+ [56] improved upon Co-teaching by introducing the "update by disagreement" strategy, where only the samples with differing predictions between the two networks are used. More recently, Xia *et al.* proposed CoDis, an extension of Co-teaching+ that employs an "update by discrepancy" strategy, selecting samples with high-discrepancy prediction probabilities between the two networks to utilize more samples [50]. These methods are based on the assumption that clean samples can be identified using certain criteria, and that network collaboration is more stable than self-selection, which may lead to error accumulation. In this paper, we develop our method based on same assumption, but we assume that we start the learning process from the pre-trained VFM.

3 Problem Setup

We consider a k-class classification task using a neural network. Let $\mathcal{X} \in \mathbb{R}^d$ denote the input space and $\mathcal{Y} \in \mathbb{R}^k$ represent the ground-truth label space. In a typical classification task, the neural network is trained to align the input space with the label space. To this end, a training dataset $D = \{(x_i, \hat{y}_i)\}_{i=1}^n$ is used for supervised learning with cross-entropy loss. In practice, a sample $(x_i, \hat{y}_i)$ is considered as a noisy labeled sample when human-annotated label $\hat{y}_i$ does not match the true label y_i. The objective of this paper is to develop a fine-tuning approach for VFMs that is robust to noise and capable of performing accurately on noisy datasets.

Given a pre-trained VFM with parameters θ_{VFM}, consisting of a sequence of layers (e.g., attention blocks in ViT [13]) $L_1, L_2, ..., L_M$, where M is the depth of θ_{VFM}, the learning objective for a classification problem can be formulated as:

$$\min_{\theta_t} \sum_{i=1}^{n} \mathcal{L}_{ce}(p(x_i|\theta_{\text{VFM}}, \theta_t), \hat{y}_i), \tag{1}$$

where θ_t and $\mathcal{L}_{ce}$ represent the parameters targeted for updating and the cross-entropy loss, respectively. Here, $p(\cdot|\theta)$ refers to the prediction for a given input using parameters θ. We refer to the training process as linear probing when θ_t is limited to the parameters of the linear layer θ_l. Additionally, we refer to the training process as full-tuning when θ_t includes θ_{VFM}, and as adapter tuning when it includes adapter parameters θ_a, which are not part of θ_{VFM}.

4 Method

In this section, we begin by describing the adapter method for fine-tuning the VFM in Section 4.1. Following this, in Section 4.2, we introduce our method, CUFIT, which utilizes three modules: a linear layer and two adapters, to combat noisy labels. The key idea behind CUFIT is to leverage the well-pre-trained features of the VFM without updating the feature extractor when handling corrupted samples. Subsequently, the adapters are trained using the samples selected in a curriculum-based training manner, as shown in Figure 2 (i.e., linear probing $\rightarrow$ intermediate adapter $\rightarrow$ last adapter). This approach helps increase the number of selected samples by reducing the domain gap between the pretraining task and the medical image task. It is important to note that our framework does not train the modules sequentially (i.e., where one module starts training only after another finishes); instead, it trains the modules simultaneously on the current batch, similar to multi-task training.

4.1 Learning with Adapter

We consider various adapters for fine-tuning VFMs on medical image datasets.In particular, adapters like Visual Prompt Tuning (VPT [23]), AdaptFormer [9], Low Rank Adaptation (LoRA [22]) and Rein [48] can be used. These methods have been shown to be efficient for various image and video tasks, even compared to full model training [9, 48]. Typically, when an adapter is used for fine-tuning, the parameters of the VFM are frozen and not included in the optimization process.

In this section, we briefly introduce how an adapter works. Note that our goal is not to propose a novel adapter but rather to present a training paradigm that can be applied to various adapters. For

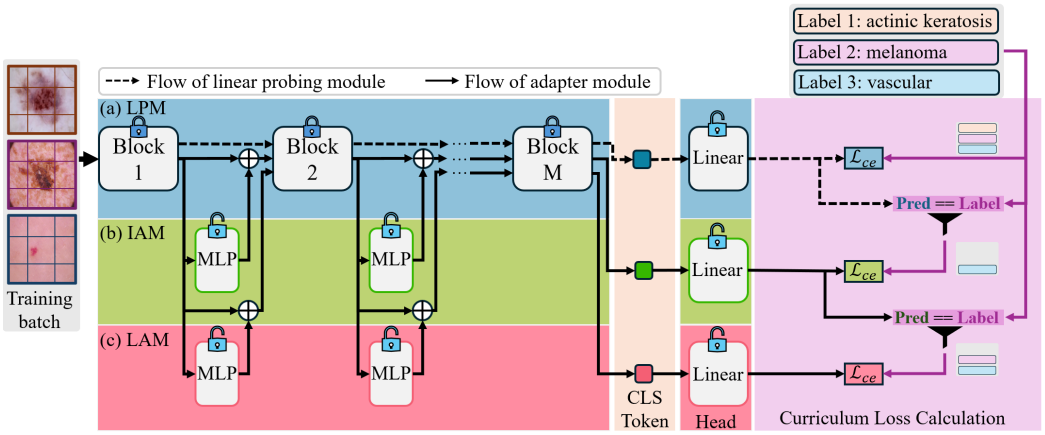


Figure 2: Illustration of our proposed training framework, CUFIT, which consists of a pre-trained VFM and three distinct modules: (a) the linear probing module (LPM), (b) the intermediate adapter module (IAM), and (c) the last adapter module (LAM). During the training stage, the LPM selects clean samples for the IAM based on the *agreement* criterion, and the IAM selects clean samples for the LPM. During the inference stage, only the LAM is used for prediction.

vision transformers (ViTs), the output of the attention block for the given input patches is calculated as follows:

$$x'_l = \text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}}V) + x_{l-1}, \tag{2}$$

where x_{l-1} is the output token of the previous block.Here, Q, K, and V refer to the query, key, and value vectors, respectively, which are derived from linear projection and LayerNorm [6] applied to x_{l-1}. The final output of the block, x_l, is then computed using LayerNorm and an MLP. Without using an adapter, this process is formulated as:

$$x_l = \text{MLP}(\text{LN}(x'_l)) + x'_l, \tag{3}$$

where x_l is the output token of the l-th block. When an adapter is used, the Eq 3 is replaced by the following:

$$x_l = \text{MLP}(\text{LN}(x'_l)) + x'_l + \text{Adapt}(x_{l-1}; \theta^a_{l-1}), \tag{4}$$

where $\text{Adapt}(\cdot; \theta^a_l)$ refers to the adapter function for the l-th layer, parameterized by θ^a_l. We consider this process to be an arbitrary function, as various adapters can be used.In the last block, the [CLS] token is passed to the following linear layer for final image classification.

4.2 Curriculum training of three different modules

We consider a pre-trained VFM, θ_{VFM}, with a single linear layer parameterized by $\theta_{\text{LPM}} \in \mathbb{R}^{c \times k}$, an intermediate adapter module parameterized by θ_{IAM}, and a last adapter module parameterized by θ_{LAM}, where c refers to the dimension of the class token (e.g., 384 dimensions for the ViT-small architecture). Then, we propose a curriculum training framework for these three modules, in which the LPM is trained with all samples from the given batch, while the adapter modules are trained with filtered samples selected by their corresponding module using the *agreement* criterion. The *agreement* criterion refers to a method where a sample is considered clean if the module's prediction matches the sample's annotation. The idea behind this criterion is that a robust classifier will correctly predict the sample under the assumption that clean labels are in the majority within a noisy class. Therefore, a sample is selected as clean if it meets the *agreement* criterion (e.g., a "dog" image with a "dog" annotation). Thus, we build the curriculum training framework based on the robustness of the LPM against noisy labels using the *agreement* criterion.

In particular, the linear probing module (LPM) is trained as follows:

$$\min_{\theta_{\text{LPM}}} \sum_{i=1}^{n} \mathcal{L}_{ce}(p(x_i|\theta_{\text{VFM}}, \theta_{\text{LPM}}), \hat{y}_i), \tag{5}$$

which directly represents supervised learning using the given images and corresponding labels. Here, $p(x_i|\theta_{\text{VFM}}, \theta_{\text{LPM}})$ refers to the output of the network using θ_{VFM} and θ_{LPM} for the given image x_i. During the training stage, the intermediate adapter module (IAM) is trained as follows:

$$\min_{\theta_{\text{IAM}}} \sum_{i=1}^{n} \mathcal{L}_{ce}(p(x_i|\theta_{\text{VFM}}, \theta_{\text{IAM}}), \hat{y}_i) \mathbb{1}\{\arg\max p(x_i|\theta_{\text{VFM}}, \theta_{\text{LPM}}) = \hat{y}_i\}, \tag{6}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. This simple modification using the indicator function ensures that the adapter module is trained only on selected samples chosen by the linear layer. Finally, the last adapter module (LAM) is trained as follows:

$$\min_{\theta_{\text{LAM}}} \sum_{i=1}^{n} \mathcal{L}_{ce}(p(x_i|\theta_{\text{VFM}}, \theta_{\text{LAM}}), \hat{y}_i) \mathbb{1}\{\arg\max p(x_i|\theta_{\text{VFM}}, \theta_{\text{IAM}}) = \hat{y}_i\}. \tag{7}$$

The Eq 7 is equivalent to Eq 6, but it uses LAM and IAM instead of IAM and LPM, respectively. This simple yet effective sample selection strategy is well-suited for fine-tuning the VFM on noisy image datasets. Notably, it does not require any hyperparameters like the estimated noise rate, which are commonly needed in previous works [17, 47, 50], where they assume the noise rate is known in order to select small-loss samples (e.g., selecting 60% of samples in a batch for a known noise rate of 40%). After training is completed, only the last adapter module is used to predict the given test image.

5 Experiments

5.1 Settings

Datasets. We evaluate our approach on four simulated noisy label medical multi-class image classification benchmarks: HAM10000 [44], APTOS-2019 [11], BloodMnist [3], and OrgancMnist [52]. Additionally, we conduct an evaluation on a real-world noisy label benchmark, Kaggle-EyePACS [2]. In particular, the detail of datasets are as follows:

- HAM10000 [44]: This dataset contains 10,015 dermatoscopic images for skin lesion classification, with each image classified into one of seven possible disease categories. We use all the images for training, and the evaluation is conducted using the 1,512 test images provided by the ISIC 2018 challenge [1].
- APTOS-2019 [11]: This dataset consists of 3,662 retina images taken with fundus photography under various imaging conditions. Each image is rated for the severity of diabetic retinopathy (DR) on a scale from 0 to 4. We use 2,930 images for training and 366 images for evaluation.
- BloodMnist [3]: This dataset contains 17,092 images of individual cells, with each image annotated as one of eight possible cell types. We use 11,959 images for training and 3,421 images for evaluation.
- OrgancMnist [52]: This dataset includes 23,538 images that are center-sliced from the Hounsfield-Unit of 3D images in a coronal view. Each image is labeled as one of eleven body organs. We use 12,975 images for training and 8,216 images for evaluation.
- Kaggle-EyePACS [2]: This Kaggle competition dataset provides 35,126 retina images categorized into five DR severity grades for training, which are known to contain noisy labels [25]. Specifically, some DR category labels (e.g., mild DR labeled as moderate DR) are noisy, and some images considered normal may actually contain retinal diseases such as glaucoma or drusen, which are not included in the classification categories. It is estimated that there is approximately a 30%–40% label error in this dataset [25, 45]. We use the original 35,126 training images and their annotations for training, and all images from APTOS-2019 for evaluation. Additionally, we use the FGADR [60] dataset for further evaluation.

Baselines. We compare the performance of CUFIT with basic training paradigms: full training, linear probing, and fine-tuning with Rein [48]. Additionally, we evaluate our approach against other training-based methods, including Co-teaching [17], JoCor [47], and CoDis [50]. Like ours, these methods do not modify the training loss or architecture. Specifically, Co-teaching trains two networks simultaneously, with each network selecting small-loss samples from its peer's predictions to guide

Dataset	Noise rate	Method						
		Full-training	Linear probing	Rein	Co-teaching	JoCor	CoDis	CUFIT
HAM10000	0.1	66.5	75.6	78.6	81.5	81.1	81.9	**82.6**
	0.2	62.6	75.3	72.1	79.1	79.4	80.1	**81.5**
	0.4	56.1	71.0	54.9	74.3	73.9	74.1	**79.1**
	0.6	59.9	61.9	37.8	67.3	67.1	66.1	**70.1**
	Mean	61.3	71.0	60.8	75.5	75.4	75.5	**78.3**
APTOS-2019	0.1	66.8	79.2	82.5	82.8	**84.8**	83.2	84.2
	0.2	65.9	79.4	78.7	81.2	83.1	82.0	**84.2**
	0.4	69.9	79.5	77.2	79.5	76.0	79.5	**81.6**
	0.6	48.2	66.9	42.0	72.9	74.2	75.7	**76.3**
	Mean	62.7	76.3	68.9	79.1	79.5	80.1	**81.6**
BloodMnist	0.1	95.4	97.2	95.9	98.6	98.5	98.5	**99.0**
	0.2	93.9	96.7	89.0	97.6	97.3	97.2	**98.8**
	0.4	91.8	95.8	69.3	93.7	93.0	93.5	**98.3**
	0.6	87.9	90.3	45.6	88.7	87.3	88.0	**98.2**
	Mean	92.3	95.0	75.0	94.7	94.0	94.3	**98.6**
OrgancMnist	0.1	85.3	83.3	87.4	92.1	92.1	92.1	**93.7**
	0.2	79.9	82.9	82.0	90.9	91.9	90.7	**93.6**
	0.4	72.1	79.9	63.8	85.8	85.3	85.8	**91.6**
	0.6	64.5	72.2	43.1	82.8	82.6	81.9	**87.4**
	Mean	75.5	79.6	69.1	87.9	88.0	87.6	**91.6**

Table 1: Average test accuracy (%) on four simulated noisy datasets with different noise levels. The test accuracy is averaged over the last ten epochs. The best and second-best results in each case are highlighted in **bold** and underline, respectively.

Testset	Method						
	Full-training	Linear probing	Rein	Co-teaching	JoCor	CoDis	CUFIT
APTOS-2019	34.2	65.4	69.1	**70.9**	69.3	69.2	69.8
FGADR	14.3	46.4	48.8	44.9	53.1	53.0	**53.7**
Total	27.5	59.0	62.3	62.2	63.9	63.8	**64.4**

Table 2: Average test accuracy (%) on real-world noisy datasets (Kaggle-EyePACS for training). After the training is done, we evaluate the model on two datasets: APTOS-2019 and FGADR. The best result and second-best result in each case are highlighted in **bold** and underline, respectively.

learning. JoCor extends this idea by incorporating co-regularization to maximize agreement between the two networks. CoDis further refines this process by selecting samples that not only have small losses but also show high divergence between the two networks. It is important to note that we do not compare our proposed framework with state-of-the-art methods that modify the training loss (e.g., semi-supervised learning) or model architecture [51, 8]. In the experiments, we apply these methods to VFMs with adapters, as they do not require specific model architectures, and VFMs with adapters outperform the linear probing of VFMs (i.e., DINOv2 with the Rein adapter is used as the default setting for training with Co-teaching, JoCor, and CoDis for a fair comparison).

Implementation details. For the experiments, we use DINOv2 [37] with the ViT-small [13] backbone as our basic vision foundation model. Additionally, we use Rein [48] as the fine-tuning adapter, originally proposed for domain-generalized semantic segmentation of VFMs. In our setup, we utilize the class token from the block for classification, rather than the patch tokens from multiple blocks.

We use the PyTorch [4] codebase for our experiments. BloodMnist and OrgancMnist datasets are sourced from MedMnist [54, 55]. We use the ViT-small architecture and the Adam optimizer [27]. All training runs for 100 epochs with a batch size of 32. The initial learning rate is 0.001, which decays by a factor of 10 at epochs 50, 75, and 90. For full-parameter training, however, we start with an initial learning rate of 0.0001. For the simulated noisy label benchmarks, we generate symmetric noise [17] for evaluation, with noise rates set at 10%, 20%, 40%, and 60%.

5.2 Simulated noisy medical image classification benchmark

First, we evaluate our framework on simulated noisy label benchmarks using four medical datasets. The average classification test accuracy for each dataset is provided in Table 1. Our framework consistently outperforms previous baselines, demonstrating its effectiveness under noisy labels by leveraging the pre-trained features of DINOv2 and the Rein adapter. Notably, our framework proves

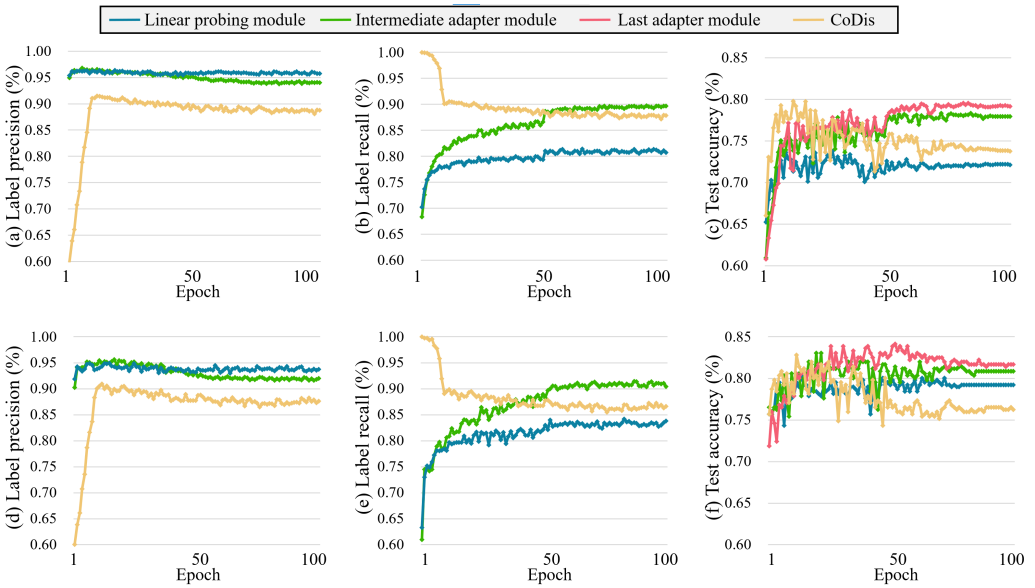


Figure 3: Illustration of label precision (a,d), label recall (b,e), and test accuracy (c,f) vs. epoch. The first row is for HAM10000 with 40% noise rate, and the second row is for APTOS-2019 with 40% noise rate.

to be more effective as the noise rate increases. For example, CUFIT achieves 0.85% relateively higher accuracy than CoDis on HAM10000 with a 10% noise rate, while the improvement rises to 3.7% at a 60% noise rate. This result indicates that the pre-trained features of the VFM are particularly useful for handling noisy labels in the given datasets.

5.3 Real-world noisy medical image classification benchmark

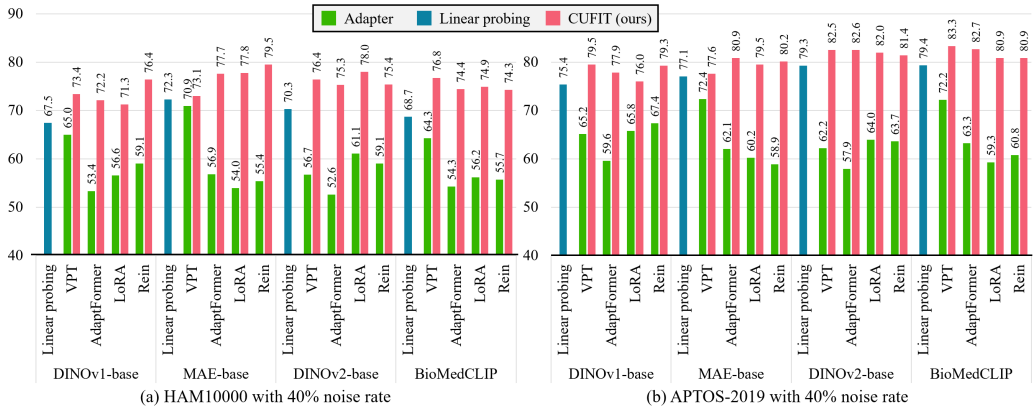
We train DINOv2 with Rein on the real-world benchmark, using the Kaggle-EyePACS dataset for training and the APTOS-2019 and FGADR datasets for testing. Given the highly imbalanced training set (e.g., approximately 73% of the samples are labeled as the normal class), we use weighted cross-entropy loss to train the model. Since previous sample selection methods require a noise rate hyperparameter, we employed the noise estimation method from [34], following Co-teaching [17].

In Table 2, we report the classification accuracy on the APTOS-2019 and FGADR datasets, as well as the overall accuracy across both datasets. Our method outperforms other baselines on the FGADR dataset and the combined dataset, while Co-teaching achieves the highest accuracy on the APTOS-2019 dataset. We believe this discrepancy is due to the distribution of normal class samples—approximately 50% in APTOS-2019 and about 5% in FGADR. Co-teaching performs well in classifying the normal class, whereas our method excels at classifying diseased samples. For example, our method achieves 53.9% macro-average test accuracy, while Co-teaching achieves 48.5% on the combined test set.

6 Discussion

6.1 How does CUFIT works?

So far, we have demonstrated through empirical results that our framework significantly improves the robustness of VFM fine-tuning against noisy labels. However, we have not yet discussed why our framework is effective in learning with noisy labels. In Figure 3, we present label precision, label recall, and test accuracy over the number of epochs to illustrate how our framework functions. In principle, higher label precision indicates fewer noisy samples in the selected data, while higher label recall indicates fewer clean samples in the unselected data. We have following three observations:



(a) HAM10000 with 40% noise rate (b) APTOS-2019 with 40% noise rate

Figure 4: Test accuracy of our method with various VFMs (DINOv1 [7], MAE [18], DINOv2 [37]) and adapters (VPT [23], AdaptFormer [9], Rein [48]). We use HAM10000 and APTOS-2019 with 40% noise rate for training.

Dataset	Noise Rate	Full-training		Linear probing		CoDis		CUFIT		
		ResNet	DINOv2	ResNet	DINOv2	ResNet	DINOv2	ResNet	ResNet+rein	DINOv2
HAM10000	0.2	73.1	66.5	71.1	75.6	74.9	80.1	77.7	79.9	82.6
	0.4	59.6	62.6	67.8	75.3	72.4	74.1	73.8	75.4	81.5
APTOS-2019	0.2	80.4	66.8	80.3	79.2	80.3	82.0	82.4	82.2	84.2
	0.4	64.7	65.9	73.4	79.4	78.1	79.5	80.5	82.2	84.2

Table 3: Average test accuracy on simulated noisy datasets (HAM10000 and APTOS-2019) using the ResNet50 architecture. Test accuracy is averaged over the last ten epochs.

- We find that CoDis exhibits lower label precision during training compared to LPM and IAM, suggesting that previous sample selection methods fail to effectively utilize the pre-trained features of VFMs, leading to lower test accuracy. These methods train the network on all training data without sample selection during the early stages of training (e.g., epochs 1 to 10 in their default setting). However, this approach may harm the feature extraction capability of VFMs and result in degraded performance.
- LPM consistently achieves the highest label precision across epochs but has the lowest label recall, indicating that it effectively prevents the memorization of noisy samples. However, because the feature extractor remains unchanged, its overall accuracy is limited, thus selecting only a small number of clean samples.
- IAM, on the other hand, achieves similar label precision but higher label recall by leveraging the adapter module, which contributes to the improved test accuracy of LAM. This suggests that by adapting the feature extractor through training the adapter on a few certain clean samples, IAM can be a more accurate module. It can then provide more clean samples to LAM, resulting in better overall performance.

6.2 Performance comparison across various VFMs and adapters

To validate the performance of CUFIT in various settings, we present experimental results using three VFMs and adapters in Figure 4. We use the same experimental setup (i.e., 100 epochs with the Adam optimizer) to train the network across all backbones and adapters. Specifically, we utilize four backbones, including DINOv1 [7], MAE [18], DINOv2 [37], and BioMedCLIP [58] and four adapters, including VPT [23], AdaptFormer [9], LoRA [22], and Rein [48]. BioMedClip is a CLIP-like model trained with the PMC-15M dataset, which contains 15 million biomedical image-text pairs collected from 4.4 million scientific articles. Our results demonstrate that our framework consistently helps build a robust classifier across different VFMs and adapters. For example, our framework achieves better performance compared to both adapter-based methods and linear probing. Additionally, we observe that linear probing consistently outperforms the adapter method in all cases, indicating that the performance of adapters can be degraded by noisy labels across various adapters.

Dataset	Noise rate	Method						
		Full-training	Linear probing	Rein	Co-teaching	JoCor	CoDis	CUFIT
CIFAR10	0.8	25.9	79.0	24.8	78.2	75.7	76.3	**83.9**
CIFAR100		6.3	59.6	25.6	66.7	64.2	63.7	**73.8**
ANIMAL10N	0.08*	74.5	89.1	88.0	92.2	91.9	91.7	**92.3**

Table 4: Average test accuracy on the natural image dataset with simulated noisy labels (CIFAR, symmetric noise at 80%) and real-world noisy labels (ANIMAL10N [41], which has an estimated noise ratio of 8%). The test accuracy is averaged over the last ten epochs. We use DINOv2 with Rein adapter for the experiment. **Bold** values the best result.

6.3 Performance on CNNs with adapters

We designed CUFIT for VFMs due to their strong pre-trained feature extraction capabilities, enabled by self-supervised training on large datasets. However, CNN-based architectures like ResNet [20] also utilize pre-trained weights instead of starting from scratch. Therefore, we validate CUFIT on ImageNet [12] pre-trained ResNet50, with and without the Rein adapter modified for ResNet. The experimental results are shown in Table 3. We observe that our method outperforms other training paradigms when using the ImageNet pre-trained ResNet architecture. Additionally, the Rein adapter for ResNet improves performance, demonstrating that using fewer trainable parameters with an adapter, compared to full training, is beneficial for combating noisy labels. Finally, we show that the more representative pre-trained features of DINOv2 outperform the ImageNet pre-trained features across all training methods.

6.4 Performance on noisy natural image classification benchmark

Since our framework is easily applicable not only to medical image classification but also to natural image classification, we present experimental results on the CIFAR [30] simulated noisy classification benchmarks and ANIMAL10N [41] real-world noisy classification benchmark in Table 4. We validate our framework under an extremely high noise rate setting (80%) for CIFAR benchmark, as it is intuitive that our framework performs well under low noise rates due to the feature extraction capabilities of VFM. As shown in Table 4, our framework outperforms other sample selection methods in natural image classification benchmarks as well. This demonstrates the effectiveness of our framework, highlighting that using well pre-trained VFMs is beneficial for detecting noisy labels in natural images, as expected.

7 Conclusion

This paper presents a curriculum fine-tuning paradigm called CUFIT, designed to robustly fine-tune vision foundation models (VFMs) for medical image classification. Our framework is based on the insight that linear probing of VFMs is robust to noisy labels, as it does not modify the feature extraction process. Building on this, CUFIT consists of three training modules: the linear probing module (LPM), the intermediate adapter module (IAM), and the last adapter module (LAM). These modules are trained simultaneously, with each selecting clean samples for the next module. Specifically, while the LPM is trained on all samples, the LPM and IAM select clean samples for the IAM and LAM, respectively (i.e., LPM$\rightarrow$IAM$\rightarrow$LAM). Experiments demonstrate that CUFIT significantly improves the performance of VFMs in the presence of noisy labels for medical image classification. Additionally, we provide extensive analyses to enhance the understanding of CUFIT. We hope our insights inspire future research to further explore the robustness of vision foundation models when learning with noisy labels for various medical imaging tasks.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II0951, Development of Uncertainty-Aware Agents Learning by Asking Questions, 90%) and Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No. 22-CM-GU-08, 10%).

References

[1] Isic 2018 challenge. https://challenge.isic-archive.com/landing/2018/.

[2] Kaggle diabetic retinopathy detection competition. https://www.kaggle.com/c/diabetic-retinopathy-detection.

[3] Andrea Acevedo, Anna Merino González, Edwin Santiago Alférez Baquero, Ángel Molina Borrás, Laura Boldú Nebot, and José Rodellar Benedé. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30(article 105474), 2020.

[4] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, April 2024.

[5] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Bingzhi Chen, Zhanhao Ye, Yishu Liu, Zheng Zhang, Jiahui Pan, Biqing Zeng, and Guangming Lu. Combating medical label noise via robust semi-supervised contrastive learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 562–572. Springer, 2023.

[9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[10] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023.

[11] Omar Dekhil, Ahmed Naglah, Mohamed Shaban, Mohammed Ghazal, Fatma Taher, and Ayman Elbaz. Deep learning based method for computer aided diagnosis of diabetic retinopathy. In *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–4. IEEE, 2019.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] 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.

[14] Raman Dutt, Linus Ericsson, Pedro Sanchez, Sotirios A Tsaftaris, and Timothy Hospedales. Parameter-efficient fine-tuning for medical image analysis: The missed opportunity. In *Medical Imaging with Deep Learning*.

[15] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[16] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[17] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] 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*, pages 770–778, 2016.

[21] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pages 2712–2721. PMLR, 2019.

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[24] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[25] Lie Ju, Xin Wang, Lin Wang, Dwarikanath Mahapatra, Xin Zhao, Quan Zhou, Tongliang Liu, and Zongyuan Ge. Improving medical images classification with label noise using dual-uncertainty estimation. *IEEE transactions on medical imaging*, 41(6):1533–1546, 2022.

[26] Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. Fine samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24137–24149, 2021.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[28] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[29] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 301–320. Springer, 2016.

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[31] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[32] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9485–9494, 2021.

[33] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

[34] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[35] 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*, pages 10012–10022, 2021.

[36] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *Advances in neural information processing systems*, 30, 2017.

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[38] Yu Qiao, Chaoning Zhang, Taegoo Kang, Donghun Kim, Shehbaz Tariq, Chenshuang Zhang, and Choong Seon Hong. Robustness of sam: Segment anything under corruptions and beyond. *arXiv preprint arXiv:2306.07713*, 2023.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[40] Poulami Sinhamahapatra, Franziska Schwaiger, Shirsha Bose, Huiyu Wang, Karsten Roscher, and Stephan Guennemann. Finding dino: A plug-and-play framework for unsupervised detection of out-of-distribution objects using prototypes. *arXiv preprint arXiv:2404.07664*, 2024.

[41] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5907–5915. PMLR, 09–15 Jun 2019.

[42] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

[43] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[44] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

[45] Xin Wang, Lie Ju, Xin Zhao, and Zongyuan Ge. Retinal abnormalities recognition using regional multitask learning. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*, pages 30–38. Springer, 2019.

[46] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330, 2019.

[47] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13726–13735, 2020.

[48] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger, fewer, & superior: Harnessing vision foundation models for domain generalized semantic segmentation. *arXiv preprint arXiv:2312.04265*, 2023.

[49] Cheng-En Wu, Yu Tian, Haichao Yu, Heng Wang, Pedro Morgado, Yu Hen Hu, and Linjie Yang. Why is prompt tuning for vision-language models robust to noisy labels? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15488–15497, 2023.

[50] Xiaobo Xia, Bo Han, Yibing Zhan, Jun Yu, Mingming Gong, Chen Gong, and Tongliang Liu. Combating noisy labels with sample selection by mining high-discrepancy examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1843, 2023.

[51] Xiaohan Xing, Zhen Chen, Zhifan Gao, and Yixuan Yuan. Gradient and feature conformity-steered medical image classification with noisy labels. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 75–84. Springer, 2023.

[52] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient multiple organ localization in ct image using 3d region proposal network. *IEEE transactions on medical imaging*, 38(8):1885–1898, 2019.

[53] Cheng Xue, Lequan Yu, Pengfei Chen, Qi Dou, and Pheng-Ann Heng. Robust medical image classification from noisy labeled data with global and local representation guided co-training. *IEEE transactions on medical imaging*, 41(6):1371–1382, 2022.

[54] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.

[55] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[56] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International conference on machine learning*, pages 7164–7173. PMLR, 2019.

[57] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.

[58] Sheng Zhang, Yanbo Xu, Naoto Usuyama, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Matthew Lungren, Tristan Naumann, and Hoifung Poon. Large-scale domain-specific pretraining for biomedical vision-language processing, 2023.

[59] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[60] Yi Zhou, Boyang Wang, Lei Huang, Shanshan Cui, and Ling Shao. A benchmark for studying diabetic retinopathy: segmentation, grading, and transferability. *IEEE Transactions on Medical Imaging*, 40(3):818–828, 2020.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We provide the main claims in the introduction and validate it in the experiments and discussion sections.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitation in the conclusion.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

Justification: We do not provide the theoretical result in the paper.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We provide the experimental setting of our paper. Also, we provide source code for experiments in supplementary.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Datasets used in this paper are available publicly. The code will be open-sourced upon publication.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We give the training and test details in the Experiment section.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: We do not report the statistical significance.

   Guidelines:

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

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

8. **Experiments Compute Resources**

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

   Answer: [No]

   Justification: Our experiments are relatively free from computational resources since it updates adapters of the model using small batch-size.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have used publicly opened code and data.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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