Collaborative Consortium of Foundation Models for Open-World Few-Shot Learning

Shuai Shao[1], **Yu Bai**[1,2], **Yan Wang**[3], **Baodi Liu**[2*], **Bin Liu**[1*]
[1]Zhejiang Lab
[2]China University of Petroleum (East China)
[3]Beihang University
{shaoshuai0914, wangyan9509, thu.liubaodi, bliu.81}@gmail.com, baiyu_upc@163.com

Abstract

Open-World Few-Shot Learning (OFSL) is a crucial research field dedicated to accurately identifying target samples in scenarios where data is limited and labels are unreliable. This research holds significant practical implications and is highly relevant to real-world applications. Recently, the advancements in foundation models like CLIP and DINO have showcased their robust representation capabilities even in resource-constrained settings with scarce data. This realization has brought about a transformative shift in focus, moving away from "building models from scratch" towards "effectively harnessing the potential of foundation models to extract pertinent prior knowledge suitable for OFSL and utilizing it sensibly". Motivated by this perspective, we introduce the **C**ollaborative **Co**nsortium of F**o**undation M**o**dels (**CO**$_3$), which leverages CLIP, DINO, GPT-3, and DALL-E to collectively address the OFSL problem. CO$_3$ comprises four key blocks: (1) the Label Correction Block (LC-Block) corrects unreliable labels, (2) the Data Augmentation Block (DA-Block) enhances available data, (3) the Feature Extraction Block (FE-Block) extracts multi-modal features, and (4) the Text-guided Fusion Adapter (TeFu-Adapter) integrates multiple features while mitigating the impact of noisy labels through semantic constraints. Only the adapter's parameters are adjustable, while the others remain frozen. Through collaboration among these foundation models, CO$_3$ effectively unlocks their potential and unifies their capabilities to achieve state-of-the-art performance on multiple benchmark datasets. https://github.com/The-Shuai/CO3.

Introduction

Few-shot learning (FSL) is a valuable area of research that enable identification in data-deficient and resource-limited scenarios and has made significant progress (Zhang et al. 2023c,a; Shao et al. 2021b; Guo et al. 2023; Shao et al. 2021a; Zhang et al. 2023b). However, methods that exhibit exceptional performance in research settings may struggle when applied to real-world situations. One critical factor is the overly idealized settings often employed in previous FSL research, which fail to adapt to the complexities of open-world scenarios. For instance, most studies assume that the label information of available data is cleaned, disregarding the presence of noise or errors commonly found in practical application scenarios.

Figure 1: An example of Open-World Few-Shot Learning (OFSL) as defined by (An et al. 2023). It portrays a scenario called 2-way 1-shot, where there are two classes, each containing only one available sample. However, these samples are accompanied by unreliable labels that may include noise. Noisy labels come from other categories that have appeared or the unseen class.

To tackle this challenge, (An et al. 2023) proposed Open-World Few-Shot Learning (OFSL) (see Fig. 1). OFSL is specifically designed to address the detrimental impact of noise in training data, which comes from both known and unseen categories. Compared to traditional weakly supervised learning (involves a small number of noisy labels alongside a large number of clean labels as guidance) and unsupervised learning (lacks labeled data entirely), OFSL faces even greater challenges due to its unique circumstances. Specifically, when there are only a few training samples available, especially just one sample per category, the negative impact of incorrect labeling on the model becomes significantly more detrimental than having no label at all. Therefore, finding effective solutions to the OFSL problem is of utmost importance. This involves developing techniques and algorithms that can robustly handle noisy labels and effectively utilize the limited available training samples.

Large-scale image recognition in the open world has been a prominent research area since 2015 (Bendale and Boult 2015), leading to significant advancements (Joseph et al. 2021; Wang, Ramanan, and Hebert 2019). However, the development of recognition tasks specifically for few-shot samples has only recently gained attention. Various strategies, such as metric learning and feature aggregation based approaches (An et al. 2023; Liang et al. 2022), have been proposed to enhance representation ability in the presence of

*Corresponding author.

noisy labels. In more recent times, there has been increasing interest in foundation models such as CLIP (Radford et al. 2021) and DINO (Caron et al. 2021). These models are pre-trained on large-scale datasets and possess powerful architectures, allowing them to retain strong representation capabilities even in scenarios with limited data and finite computational resources. This realization motivates us to shift the focus from "designing models from scratch" towards "harnessing the potential and expertise embedded within these pre-trained foundation models in a sensible manner to enhance OFSL performance". Compared to starting with a blank slate, foundation models undergo extensive validation and tuning, making them more robust against overfitting issues commonly encountered in OFSL, while also saving much time and computational resources.

In this paper, we propose the **C**ollaborative **Co**nsortium of F**o**undation M**o**dels (**CO**$_3$, see Fig. 2) for OFSL, leveraging the unique capabilities of four foundation models: CLIP (Radford et al. 2021), GPT-3 (Brown et al. 2020), DINO (Caron et al. 2021), and DALL-E (Ramesh et al. 2021). Each model contributes to the consortium's strength in enabling cross-modal comparisons, comprehensive description generation, feature extraction, and image generation. CO$_3$ comprises 4 distinct blocks: (1) **L**abel **C**orrection Block (**LC-Block**, see Fig. 3) utilizes CLIP, GPT-3, DINO, and DALL-E to achieve accurate label correction. (2) **D**ata **A**ugmentation Block (**DA-Block**) employs DALL-E to generate diverse samples based on the corrected labels, enriching the available training data. (3) **F**eature **E**xtraction Block (**FE-Block**) adopts CLIP and DINO to obtain three types of features: DINO-based image features, CLIP-based image features, and CLIP-based text features. (4) **Te**xt-guided and **Fu**sion Adapter (**TeFu-Adapter**, see Fig. 4) is specifically designed for the OFSL task. It fuses the aforementioned features and utilizes the semantic modality information of the samples to constrain the adapter's parameters. This approach effectively mitigates the impact of noise labels on the model. By incorporating these four blocks, CO$_3$ offers a comprehensive and robust solution to the challenges faced in OFSL.

Our main contributions are summarized as follows:

- We propose CO$_3$, which effectively leverages the potential and prior knowledge of foundation models to enhance OFSL performance.
- We introduce TeFu-Adapter, a specially designed mechanism that reduces the negative impact of noisy labels during the adapter updating stage.
- We extensively evaluate CO$_3$ on multiple benchmark datasets, demonstrating significant improvements compared to other state-of-the-art (SOTA) methods.

Related Work

Foundation Models Recently, research on foundation models is in full swing. Here, we introduce four models used in our paper: **GPT-3** (Brown et al. 2020) is a Transformer-based language model that captures statistical patterns and structures in language. It predicts the next word in a sentence to understand contextual relationships, enabling it to excel in tasks like machine translation and text completion. **CLIP** (Radford et al. 2021) is a versatile vision and language model that aligns image and text representations in a shared latent space. It uses contrastive learning to maximize agreement between matching pairs of image-text inputs while minimizing agreement between non-matching pairs. CLIP learns to understand concepts and relationships across different modalities by training on large datasets with diverse images and text. **DALL-E** (Ramesh et al. 2021) is a powerful generative model that synthesizes highly diverse and realistic images from textual descriptions. Through an encoder-decoder architecture, DALL-E encodes natural language descriptions into latent vectors, which are then used by the decoder to generate corresponding images. **DINO** (Caron et al. 2021) is an unsupervised learning framework consisting of teacher and student networks. The teacher network encodes data into high-dimensional representations, which the student network aligns with through contrastive learning. By leveraging the teacher-student framework and contrastive loss, DINO enhances unsupervised representation learning, applicable to tasks like classification and object detection.

Open-World Few-Shot Learning Since 2015, open-world object recognition (Bendale and Boult 2015, 2016) gained attention by detecting open sets and managing many samples. Now, researchers focus on solving recognition problems in scenarios with limited data, leading to OFSL (An et al. 2023; Willes et al. 2023). There's a focus on how noisy labels affect seen and unseen classes. OFSL, akin to weakly supervised learning but with few samples, differs from robust few-shot learning (Lu et al. 2021) by addressing label noise from both visible and unseen classes. In contrast to conventional approaches like metric learning (An et al. 2023), instance reweighting (Lu et al. 2021), and feature aggregation (Liang et al. 2022) commonly used to tackle this challenge, this paper introduces a pioneering alternative method that harnesses the power of foundation models to effectively overcome the hurdles associated with OFSL.

Foundation Solutions on Few-Shot Learning In the field of FSL, several notable works have emerged, harnessing the capabilities of foundation models to achieve remarkable progress. For examples, Tip-Adapter (Zhang et al. 2022) is a training-free method for enhancing CLIP's few-shot classification by constructing an adapter through a key-value cache model, updating the prior knowledge encoded in CLIP via feature retrieval; CaFo (Zhang et al. 2023c) leverages linguistic prompts from GPT, synthetic images from DALL-E, and a learnable cache model to blend predictions from CLIP and DINO, achieving advanced performance by harnessing the potential of various pre-trained methods. Moreover, many follow-up works (Guo et al. 2023; Zhou et al. 2022; Guo et al. 2023; Zhu et al. 2023; Roy et al. 2022; Cui et al. 2023; Rong et al. 2023; Palanisamy et al. 2023) also make significant contributions to the research community.

The research work that greatly inspired us is CaFo. However, there are three key differences between CaFo and CO$_3$: (1) CaFo mainly focuses on general situations and overlooks the open-world case, while our approach takes into account both scenarios; (2) The overall structure of our CO$_3$ differs significantly from CaFo's. (3) Moreover, unlike CaFo, which

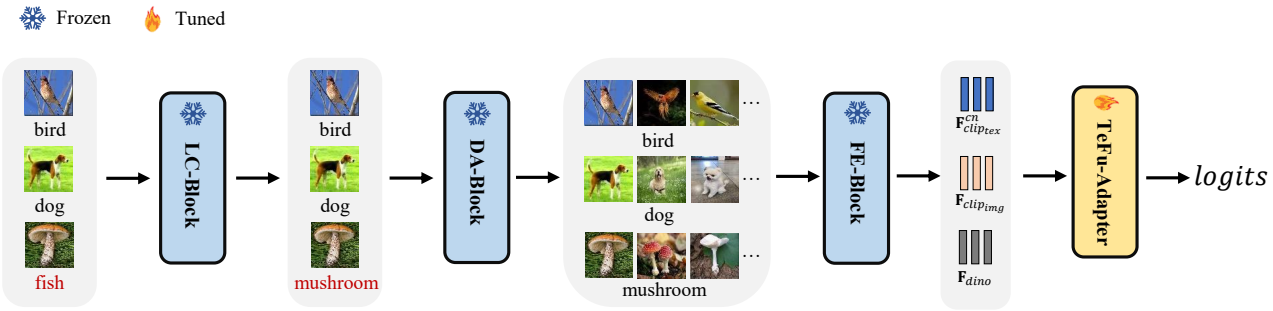

Figure 2: The flowchart of our Collaborative Consortium of Foundational Model (CO_3).

uses a mature Tip-Adapter for model adjustment, CO_3 incorporates a task-specific TeFu-Adapter designed to precisely meet the requirements of the task. This tailored adaptation enables our entire pipeline to operate more effectively.

Problem Setup

Few-Shot Learning In the standard FSL, the base set $\mathcal{D}_{base}$ contains a large amount of labeled data, which is used to pre-train a feature extraction model for downstream tasks. The novel set $\mathcal{D}_{novel}$ is the general term for all the data in the downstream tasks, and it is divided into support set $\mathcal{S}$ and query set $\mathcal{Q}$, where $\mathcal{S} \cap \mathcal{Q} = \emptyset$. $\mathcal{S} = \{(x_i, y_i, t_i)\}_{i=1}^{N\times K}$ contains a few labeled data, and the $\mathcal{Q} = \{(x_i, y_i, t_i)\}_{i=N\times K}^{N\times K+M}$ is the to-be-tested data, where x_i denotes the image sample, $y_i \in \mathcal{L}$ represents its label, $t_i \in \mathcal{T}$ is its category name; $\mathcal{L}$ and $\mathcal{T}$ are the label and category name sets; N is the number of classes in $\mathcal{S}$, K is the number of samples per class, they are usually called *N-way K-shot*, and M is the number of samples in $\mathcal{Q}$. FSL aims to use only a few support samples to accurately recognize the categories of query data.

Open-World Few-Shot Learning. Compared to the FSL, OFSL is a more difficult but more valuable task for practical applications. Its goal is to efficiently identify the query categories under the premise that the support labels are subject to random contamination and lack reliability. $\mathcal{S} = \{\mathcal{S}_{clean}, \mathcal{S}_{noise}\}_{i=1}^{N\times K}$, where $\mathcal{S}_{clean} = \{(x_i, y_i, t_i)\}$, $\mathcal{S}_{noise} = \{(x_i, y_j, t_j)\}$. $y_j \in \hat{\mathcal{L}}$ is the noise label, indicating that the i-th image belongs to class-i, but is erroneously labeled as class-j; $t_j \in \hat{\mathcal{T}}$ denotes the corresponding noise category name. $\hat{\mathcal{L}}$ and $\hat{\mathcal{T}}$ denote the noise label set and the category name set. The samples' noise labels come from other categories that have appeared in the support set, or the unseen class. In the training stage, we do not know which sample's label is the noise label.

Methodology

Overview Unlike conventional OFSL approaches that rely on additional base data to train a feature extractor, this paper takes a different approach by directly utilizing frozen foundation models (CLIP, DINO, DALL-E, GPT-3) and incorporating our specially designed adapter. The training framework for *3-way 1-shot* recognition is illustrated in Fig. 2. It involves the following steps:

- Feeding the support data into the Label Correction Block (LC-Block) to clean the data's noisy labels. The LC-Block consists of parameter-frozen foundation models, namely CLIP, DINO, DALL-E, and GPT-3.
- Sending the corrective data to the Data Augmentation Block (DA-Block) to get more abundant training data. This block is also frozen and uses the DALL-E model.
- Inputting the cleaned original data and augmented data to the Feature Extraction Block (FE-Block) to obtain three kinds of features, which are: (1) the image feature extracted by DINO encoder; (2) the image feature extracted by CLIP's image encoder; (3) and the text feature corresponding to the image's category name extracted by CLIP's text encoder. All the models are frozen.
- Fusing these features with our designed Text-guided and Fusion Adapter (TeFu-Adapter) and realizing the classification. The TeFu-Adapter is the only block that needs fine-tuning in the pipeline.

In the inference stage, we only need the FE-Block and TeFu-Adapter to achieve classification.

Label Correction Block We design a two-pronged LC-Block and illustrate its flowchart in Fig. 3.

The first branch combines DALL-E with DINO to achieve an initial label correction result, which involves three stages: (1) We input the category names of support data (may contain incorrect labels) into the DALL-E model, which generates new images based on these names. Then, we extract features from these generated images using DINO. By averaging these features, we obtain pseudo prototypes for each image class. (2) Simultaneously, we utilize DINO to directly extract features from the original support images. (3) We next calculate the similarity between these extracted features and the prototypes obtained earlier. This enables us to determine a coarse-grained label for each image based on their feature similarities. The process can be formulated as:

$$\hat{\mathbf{F}}^{p}_{dino} = Prototype\left(\mathcal{M}_{dino}\left(\mathcal{M}_{dalle}\left(\hat{\mathcal{T}}\right)\right)\right) \quad (1)$$

$$\hat{\mathbf{F}}_{dino} = \mathcal{M}_{dino}\left(\mathcal{X}\right) \quad (2)$$

$$\hat{\mathbf{U}}_1 = \left(\hat{\mathbf{F}}_{dino}\right)^T \hat{\mathbf{F}}^{p}_{dino} \quad (3)$$

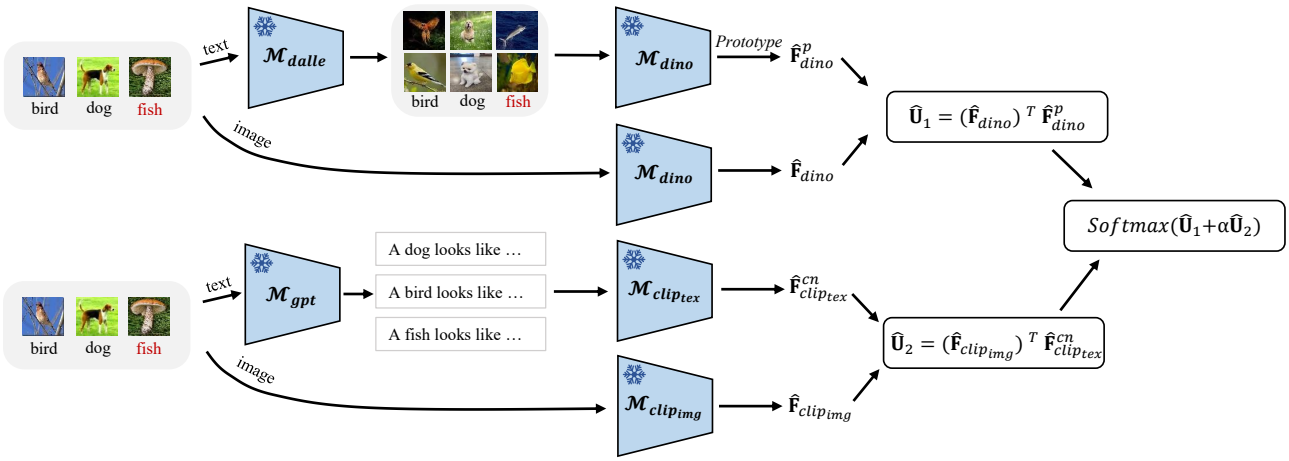

Figure 3: The flowchart of Label Correction Block (LC-Block). It comprises three steps: (1) Firstly, we generate new samples using DALL-E based on category names. Then, we use DINO to extract features from both the generated and original images. Next, we compute prototypes for each class using the generated data. Predictions are made by comparing the similarity between the original features and the prototypes. (2) In parallel, we use GPT-3 to capture descriptions of different categories. These descriptions are inputted into the CLIP text encoder to obtain encoded representations. Simultaneously, the original images are processed using the CLIP image encoder to extract distinctive features. Predictions are made by comparing the similarity between image features and the encoded category descriptions. (3) Labels are corrected based on the results from steps (1) and (2). All models remain frozen without further updates during the training process.

where $\mathcal{M}_{dalle}$ and $\mathcal{M}_{dino}$ denote the frozen DALL-E and DINO; $Prototype$ is the operator to compute the pseudo prototypes for all classes of generated images; $\mathcal{X}$ is the set of original images; $\hat{\mathbf{F}}^{p}_{dino} \in \mathbb{R}^{dim \times \hat{N}}$ is the prototype features, dim denotes the dimension, and $\hat{N}$ represents the length of the noise support label set; $\hat{\mathbf{F}}_{dino} \in \mathbb{R}^{dim \times NK}$ indicates the original image features; $\hat{\mathbf{U}}_1 \in \mathbb{R}^{NK \times \hat{N}}$ denotes the similarity matrix between the original images and the prototypes.

The second branch utilizes GPT-3 and CLIP for cross-modal comparisons, combining visual and textual information, which also has three components: (1) Initially, we use GPT-3 to generate comprehensive descriptions for the category names in the support set. Then, we employ CLIP's text encoder to encode these descriptions, obtaining informative text representations for each class. This step captures detailed textual information that accurately describes each category. (2) Next, we extract features from the original support images using CLIP's image encoder. This enables us to capture the visual characteristics and details of each sample. (3) Following, we calculate the similarity between the text features obtained from the class names and the image features derived from the original support set. By measuring the similarity between these two sets of features, we establish a meaningful correspondence between the textual descriptions and the visual content. This allows us to refine and validate the labels associated with the support set samples accurately. We formulated the process as:

$$\hat{\mathbf{F}}^{cn}_{clip_{tex}} = \mathcal{M}_{clip_{tex}}\left(\mathcal{M}_{gpt}\left(\hat{\mathcal{T}}\right)\right) \tag{4}$$

$$\hat{\mathbf{F}}_{clip_{img}} = \mathcal{M}_{clip_{img}}\left(\mathcal{X}\right) \tag{5}$$

$$\hat{\mathbf{U}}_2 = (\hat{\mathbf{F}}_{clip_{img}})^T \hat{\mathbf{F}}^{cn}_{clip_{tex}} \tag{6}$$

where $\mathcal{M}_{gpt}$, $\mathcal{M}_{clip_{tex}}$, and $\mathcal{M}_{clip_{img}}$ denote the frozen GPT-3, CLIP's text encoder and CLIP's image encoder; $\hat{\mathbf{F}}^{cn}_{clip_{tex}} \in \mathbb{R}^{dim \times \hat{N}}$ indicates the features of textual category names; $\hat{\mathbf{F}}_{clip_{img}} \in \mathbb{R}^{dim \times NK}$ indicates the original image features; $\hat{\mathbf{U}}_2 \in \mathbb{R}^{NK \times \hat{N}}$ denotes the similarity matrix between the original images and the class names.

Then we combine the results from both branches, considering their strengths and weaknesses. This integrated approach guarantees a dependable and comprehensive solution, resulting in more accurate and comprehensive results. The process can be summarized as follows:

$$\hat{p} = Softmax(\hat{\mathbf{U}}_1 + \alpha\hat{\mathbf{U}}_2) \tag{7}$$

$$\dot{\mathcal{L}}, \dot{\mathcal{T}} = Refinement(\hat{p}, \beta) \tag{8}$$

where $\hat{p}$ represents the probability that a sample belongs to a specific class; $\dot{\mathcal{L}}$ and $\dot{\mathcal{T}}$ denote the sets of corrective labels and category names, respectively. α and β are the hyper-parameters; $Refinement$ is a sampling process to correct labels, which is outlined below: (1) If the predicted result aligns with the given label, we consider the label to be correct. (2) In cases of inconsistent predictions and given labels, we assess the prediction probability. If it surpasses a threshold (β), we adopt the predicted label as the ground-truth label. (3) However, if the prediction probability falls below this threshold (β), we retain the original label as the ground-truth label. This approach ensures accurate label correction by considering the confidence of predictions while respecting the initial labeling information.

Data Augmentation Block. DA-Block tackles the limited training data challenge in FSL tasks by expanding and enriching the available data. Similar to LC-Block, DA-Block utilizes DALL-E to generate image information from category names. However, the key difference is that the category names provided as input in DA-Block have undergone label correction. To optimize efficiency, DA-Block only generates non-overlapping category images between sets $\dot{\mathcal{T}}$ and $\hat{\mathcal{T}}$. This eliminates redundancy and allows direct access to the remaining images from LC-Block, maximizing computational resource utilization. By incorporating DA-Block, we overcome data scarcity in FSL task by augmenting and diversifying support samples. This augmentation enhances the model's generalization ability and performance on query data. The augmented data set is denoted as $\mathcal{X}^{aug}$, which can be formulated as:

$$\mathcal{X}^{aug} = \mathcal{M}_{dalle}(\dot{\mathcal{T}}) \tag{9}$$

Feature Extraction Block. FE-Block is a crucial component responsible for extracting essential features from both the original cleaned data and augmented data. It generates three types of features: (1) The DINO encoder is used to extract image features, capturing visual characteristics and patterns for subsequent analysis and classification.; (2) The CLIP's image encoder is employed within the FE-Block to extract rich representations of the visual content in the images, enhancing a comprehensive understanding of their visual attributes; (3) FE-Block also utilizes CLIP's text encoder to extract text features from category names associated with the images. This process enhances semantic understanding and facilitates alignment with their respective classes. We formulate the process as:

$$\mathbf{F}^{cn}_{clip_{tex}} = \mathcal{M}_{clip_{tex}}\left(\mathcal{M}_{gpt}\left(\dot{\mathcal{T}}\right)\right) \tag{10}$$

$$\mathbf{F}_{clip_{img}} = \mathcal{M}_{clip_{img}}\left(\mathcal{X}, \mathcal{X}^{aug}\right) \tag{11}$$

$$\mathbf{F}_{dino} = \mathcal{M}_{dino}\left(\mathcal{X}, \mathcal{X}^{aug}\right) \tag{12}$$

where $\mathbf{F}^{cn}_{clip_{tex}} \in \mathbb{R}^{dim \times \dot{N}}$ indicates the features of corrective textual category names; $\dot{N}$ represents the length of the corrective support label set; $\mathbf{F}_{clip_{img}} \in \mathbb{R}^{dim \times (NK + \dot{N}K')}$ denotes the original and augmented CLIP features; K' denotes the augmented shots per class by DALL-E; $\mathbf{F}_{dino} \in \mathbb{R}^{dim \times (NK + \dot{N}K')}$ denotes the original and augmented DINO features. By combining these three types of extracted features, the FE-Block enables a holistic representation of both the visual and textual aspects of the data. These features serve as valuable inputs for subsequent classification.

Text-guided Fusion Adapter. In light of the preceding operations, we have successfully extracted three significant features from DINO and CLIP. The focus of this section is to maximize the utilization of this diverse information while mitigating the detrimental impact of label noise. To accomplish these goals, an innovative approach called TeFu-Adapter is introduced (see Fig. 4). TeFu-Adapter utilizes a simple multilayer perceptron (MLP) to fuse the visual features extracted from DINO and CLIP's image encoders.

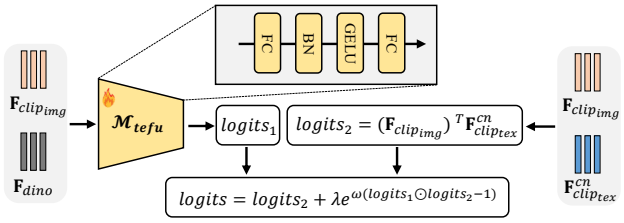

Figure 4: The flowchart of Text-guided Fusion Adapter (TeFu-Adapter).

The fusion process enables the computation of logits by directly incorporating the corrective labels. Leveraging this approach, the TeFu-Adapter ensures seamless integration of visual information from both encoders, facilitating precise calculations and predictions. We define this step as follows:

$$logits_1 = \mathcal{M}_{tefu}(\mathbf{F}_{clip_{img}}, \mathbf{F}_{dino}) \tag{13}$$

where $\mathcal{M}_{tefu}$ denotes the TeFu-Adapter, $logits_1 \in \mathbb{R}^{(NK + \dot{N}K') \times \dot{N}}$.

Furthermore, acknowledging the potential presence of errors in the corrective labels, the TeFu-Adapter incorporates the text encoding of each category to guide the calculation of image features and get the corresponding logits. By incorporating semantic information derived from the category names, the TeFu-Adapter diminishes reliance on potentially erroneous labels, enhancing the approach's robustness. We formulate this step as:

$$logits_2 = (\mathbf{F}_{clip_{img}})^T \mathbf{F}^{cn}_{clip_{tex}} \tag{14}$$

where $logits_2 \in \mathbb{R}^{(NK + \dot{N}K') \times \dot{N}}$.

Finally, the TeFu-Adapter merges the two sets of logits and calculates the cross-entropy loss. By skillfully leveraging the TeFu-Adapter to seamlessly integrate textual and visual information while mitigating label noise, this methodology bolsters the robustness and precision of the learning process. We formulate this step as:

$$logits = logits_2 + \lambda e^{\omega(logits_1 \odot logits_2 - 1)} \tag{15}$$

$$loss = CrossEntropy\left(Softmax\left(logits\right)\right) \tag{16}$$

where $logits \in \mathbb{R}^{(NK + \dot{N}K') \times \dot{N}}$; $\odot$ denotes the Hadamard inner product; λ and ω are the hyperparameters.

Experiments

Datasets We evaluate the performance of our methods on multiple well-known publicly available datasets: ImageNet (Deng et al. 2009), OxfordPets (Parkhi et al. 2012), Caltech101 (Fei-Fei, Fergus, and Perona 2004), Food101 (Bossard, Guillaumin, and Van Gool 2014), Sun397 (Xiao et al. 2010). We follow CaFo (Zhang et al. 2023c) and APE (Zhu et al. 2023) to train our models using 1, 2, 4, 8, and 16 labeled samples per class from the support set, and then test them on the entire query set. We introduce varying proportions of noisy labels to the support data in each dataset.

Methods	Time	Noisy Label Proportion										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
CoOp (Zhou et al. 2022)	45min	57.15	60.04	57.15	56.58	55.90	50.70	40.80	36.91	22.71	9.44	4.57
Tip-Adapter-F (Zhang et al. 2022)	1min	61.32	60.67	60.35	59.97	59.86	59.00	59.94	58.92	58.32	57.35	57.73
CLIP-Adapter (Gao et al. 2023)	2min	61.20	59.21	57.45	55.19	53.07	51.94	50.44	45.90	38.91	17.74	18.85
CALIP-FS (Guo et al. 2023)	20min	61.35	58.07	57.56	56.86	57.07	56.23	57.07	56.08	58.08	57.56	58.07
CaFo (Zhang et al. 2023c)	7min	**63.80**	61.53	60.16	59.99	59.78	58.70	58.64	57.98	58.52	58.82	59.03
APE-T (Zhu et al. 2023)	1min	62.50	58.43	57.00	51.02	54.04	52.90	51.72	51.53	51.17	50.80	50.20
CO_3 (Ours)	7min	63.07	**63.03**	**63.06**	**62.86**	**62.65**	**62.72**	**62.71**	**62.61**	**62.43**	**62.50**	**62.58**

Table 1: 1-shot accuracy (%) of methods on ImageNet. *Time* denotes the training time on one A100 GPU.

Implementation Our approach integrates GPT-3, DALL-E, CLIP, and DINO. GPT-3 is responsible for generating category descriptions, while DALL-E generates images for each category. We directly adopted the design of CaFo. CLIP and DINO serve as feature extractors. We use ResNet50 as the backbone for CLIP. The TeFu-Adapter, comprising two linear layers, is initialized using Kaiming initialization. We set the initial learning rate to 0.001 and employ AdamW as the optimizer, along with CosineAnnealingLR as the scheduler. During training, the data undergoes operations such as random cropping, random flipping, and normalization, with a batch size of 256. For testing, we use a batch size of 64. To introduce noise in the labels, incorrect labels are randomly assigned to support samples.

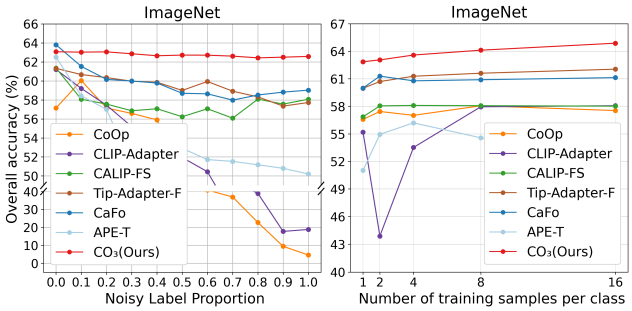

Figure 5: Performance (%) Comparison on ImageNet.

Performance on ImageNet We compare foundation model based methods utilizing frozen foundation models with added adapters for fine-tuning, including CoOp (Zhou et al. 2022), Tip-Adapter (Zhang et al. 2022), CLIP-Adapter (Gao et al. 2023), CALIP-FS (Guo et al. 2023), CaFo (Zhang et al. 2023c), and APE-T (Zhu et al. 2023).

Tab. 1 and Fig. 5(left) present the results under 1-shot conditions with varying proportions of noisy labels. Fig. 5(right) shows the results with a fixed noisy label ratio of 0.3 and varying numbers of available samples per class. Based on our observations, the following conclusions can be drawn: (1) CO_3 surpasses other SOTAs in the open-world setting, delivering outstanding performance even when the noise ratio reaches 100%. Furthermore, it maintains low computational costs, achieving an advantageous balance between performance and efficiency. (2) In the 1-shot setting with 0.3 noisy label proportion, CO_3 achieves a remarkable result of 62.86%, surpassing all comparison methods across 16-shots. (3) For most methods, an increase in available support data does not significantly enhance model performance due to the negative impact of noisy labels. In fact, many methods experience performance degradation. However, our method is robust to noise-induced variations. These findings demonstrate that our CO_3 effectively and stably addresses the OFSL problem.

Performance on Other Datasets To rigorously evaluate the robustness of our CO_3 across different scenarios, we conducted extensive testing on 10 additional datasets. The experimental results for OxfordPets, Caltech101, Food101, and Sun397 can be found in Fig. 6. Upon observing the results, our method consistently exhibits leading performance across multiple datasets in open-world cases, reaffirming its exceptional robustness. This impressive performance is attributed to two key factors: the collaborative utilization of diverse foundation models and the design of specific adapters. The consistent superiority of our method over alternative approaches highlights its distinct advantage in effectively tackling the OFSL challenge.

Ablation Study We conduct ablation studies and list the results in Tab 2,3 to assess the efficiency of different blocks.

(1) **LC-Block** serves a dual role in the pipeline. It can act as an auxiliary module to assist in achieving the final classification, or it can independently leverage the foundation models for classification purposes. From Tab. 2, we observe that directly using the LC-Block for classification yields unsatisfactory results (line ①). However, employing it as an auxiliary module and making decisions based on thresholds leads to improvements of at least 1.8% (lines ② and ⑥). These observations highlight the effectiveness of utilizing the LC-Block in a complementary manner within the pipeline.

(2) **DA-Block** plays a vital role in effectively expanding few-shot data by leveraging the prior knowledge of the DALL-E model. The results of lines ③ and ⑥ in Tab. 2 emphasize the significance of the DA-Block, as it can lead to a notable improvement of approximately 1.2% in OFSL task.

(3) **FE-Block** works in conjunction with the TeFu-Adapter, seamlessly integrating CLIP and DINO to collect two different types of features. By examining lines ④ and ⑥ in Tab. 2, it is evident that utilizing fusion features improves accuracy by 3.5% compared to solely relying on CLIP fea-

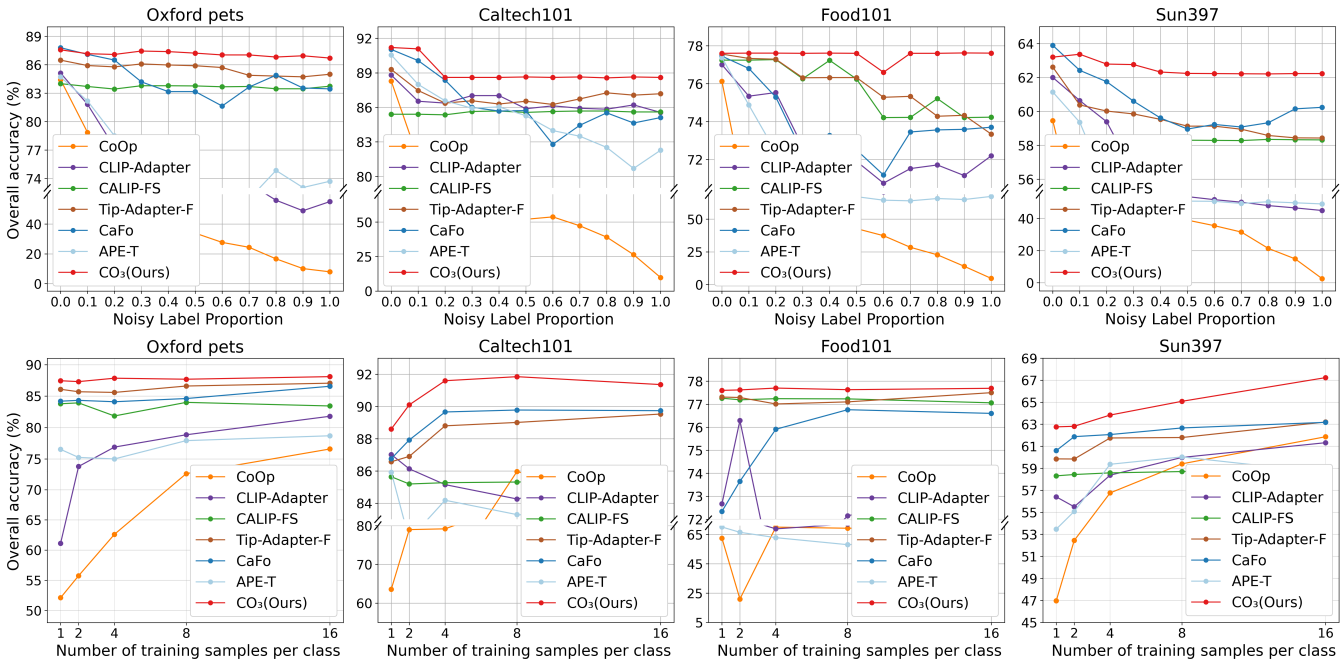

Figure 6: Performance (%) comparison on other datasets.

LC	DA	FE		TeFu		NLP
		CLIP	DINO	$logits_1$	$logits_2$	0.3
✓						61.14
	✓	✓	✓	✓	✓	61.02
✓		✓	✓	✓	✓	61.69
✓	✓	✓		✓		59.32
✓	✓		✓		✓	31.05
✓	✓	✓	✓	✓	✓	**62.86**

Table 2: Ablation study (%) of different blocks on ImageNet with 1-shot case. *NLP* is short for *Noisy Label Proportion*.

Adapters	NLP				
	0.1	0.3	0.5	0.7	0.9
w/o Adapter	20.62	15.17	15.56	13.17	15.92
Tip-Adapter	61.00	59.45	58.24	56.41	52.29
CLIP-Adapter	59.97	56.33	52.73	43.21	30.08
TeFu-Adapter	**63.03**	**62.86**	**62.72**	**62.61**	**62.50**

Table 3: Ablation study (%) of different adapters on ImageNet with 1-shot case. All comparison methods adopt our model architecture but utilize different final adapters. *NLP* is short for *Noisy Label Proportion*.

tures. Moreover, relying exclusively on DINO features (lines ⑤ and ⑥) results in a significant drop in accuracy to around 30%. These findings emphasize the importance of integrating both CLIP and DINO features through FE-Block.

(4) **TeFu-Adapter** has proven to be effective in facilitating multimodal fusion, as discussed in relation to the FE-Block above. To further demonstrate its superiority, we conduct a comparison with classic Tip-Adapter, CLIP-Adapter structures, and the absence of any adapters. Results are presented in Tab. 3. Significantly, the TeFu-Adapter outperforms other adapters consistently, achieving improvements ranging from 2% to 10% under various noise conditions. Additionally, the introduction of text guidance in the TeFu-Adapter mitigates the impact of noise, resulting in minimal fluctuations when noise is introduced. These observations highlight the notable advantages of the TeFu-Adapter, showcasing its ability to enhance performance and maintain stability even in the presence of noise.

Conclusion

To tackle the challenge posed by OFSL, we introduce CO_3, an innovative approach that leverages prior knowledge within foundation models. Extensive experiments on multiple datasets have demonstrated its efficacy. Looking ahead, our forthcoming endeavors will be concentrated on two pivotal areas: (1) We plan to expand the scope of OFSL by addressing a wider range of practical tasks beyond what has been studied in this paper. This will enable us to bridge the gap between research and real-world applications. (2) While acknowledging the foundation model's accomplishments, we are committed to delving into the underlying causes behind its occasional underperformance, thereby unlocking the untapped potential of the foundation model and bolstering its overall effectiveness.

Acknowledgments

This work was supported by Youth Foundation Project of Zhejiang Lab (No.K2023RC0AA01), Exploratory Research Project of Zhejiang Lab (No.2022RC0AN02), Major Basic Research Project in Shandong Province (No.ZR2023ZD32).

References

An, Y.; Xue, H.; Zhao, X.; and Wang, J. 2023. From instance to metric calibration: a unified framework for open-world few-shot learning. *TPAMI*, 9757–9773.

Bendale, A.; and Boult, T. 2015. Towards open world recognition. In *CVPR*, 1893–1902.

Bendale, A.; and Boult, T. E. 2016. Towards open set deep networks. In *CVPR*, 1563–1572.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *ECCV*, 446–461.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *NeurIPS*, volume 33, 1877–1901.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *ICCV*, 9650–9660.

Cui, Y.; Yu, Z.; Cai, R.; Wang, X.; Kot, A. C.; and Liu, L. 2023. Generalized few-shot continual learning with contrastive mixture of adapters. *arXiv preprint arXiv:2302.05936*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 178–178.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2023. Clip-adapter: Better vision-language models with feature adapters. *IJCV*.

Guo, Z.; Zhang, R.; Qiu, L.; Ma, X.; Miao, X.; He, X.; and Cui, B. 2023. Calip: Zero-shot enhancement of clip with parameter-free attention. In *AAAI*, volume 37, 746–754.

Joseph, K.; Khan, S.; Khan, F. S.; and Balasubramanian, V. N. 2021. Towards open world object detection. In *CVPR*, 5830–5840.

Liang, K. J.; Rangrej, S. B.; Petrovic, V.; and Hassner, T. 2022. Few-shot learning with noisy labels. In *CVPR*, 9089–9098.

Lu, J.; Jin, S.; Liang, J.; and Zhang, C. 2021. Robust few-shot learning for user-provided data. *TNNLS*, 32(4): 1433–1447.

Palanisamy, K.; Chao, Y.-W.; Du, X.; Xiang, Y.; et al. 2023. Proto-clip: Vision-language prototypical network for few-shot learning. *arXiv preprint arXiv:2307.03073*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *CVPR*, 3498–3505. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831.

Rong, J.; Chen, H.; Chen, T.; Ou, L.; Yu, X.; and Liu, Y. 2023. Retrieval-enhanced visual prompt learning for few-shot classification. *arXiv preprint arXiv:2306.02243*.

Roy, A.; Shah, A.; Shah, K.; Roy, A.; and Chellappa, R. 2022. DiffAlign: Few-shot learning using diffusion based synthesis and alignment. *arXiv preprint arXiv:2212.05404*.

Shao, S.; Xing, L.; Wang, Y.; Xu, R.; Zhao, C.; Wang, Y.; and Liu, B. 2021a. Mhfc: Multi-head feature collaboration for few-shot learning. In *ACMMM*, 4193–4201.

Shao, S.; Xing, L.; Xu, R.; Liu, W.; Wang, Y.; and Liu, B. 2021b. Mdfm: Multi-decision fusing model for few-shot learning. *TCSVT*, 32(8): 5151–5162.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2019. Meta-learning to detect rare objects. In *ICCV*, 9925–9934.

Willes, J.; Harrison, J.; Harakeh, A.; Finn, C.; Pavone, M.; and Waslander, S. 2023. Bayesian embeddings for few-shot open world recognition. *TPAMI*.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492.

Zhang, J.; Gao, L.; Hao, B.; Huang, H.; Song, J.; and Shen, H. 2023a. From Global to Local: Multi-scale Out-of-distribution Detection. *TIP*.

Zhang, J.; Gao, L.; Luo, X.; Shen, H.; and Song, J. 2023b. DETA: Denoised Task Adaptation for Few-Shot Learning. In *ICCV*.

Zhang, R.; Fang, R.; Zhang, W.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2022. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *ECCV*.

Zhang, R.; Hu, X.; Li, B.; Huang, S.; Deng, H.; Qiao, Y.; Gao, P.; and Li, H. 2023c. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *CVPR*, 15211–15222.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to prompt for vision-language models. *IJCV*, 130(9): 2337–2348.

Zhu, X.; Zhang, R.; He, B.; Zhou, A.; Wang, D.; Zhao, B.; and Gao, P. 2023. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *ICCV*.