

DeIL : Direct-and-Inverse CLIP for Open-World Few-Shot Learning

Shuai Shao¹, Yu Bai², Yan Wang³, Baodi Liu^{2*}, Yicong Zhou⁴

¹Zhejiang Lab,²China University of Petroleum (East China)

³Beihang University,⁴University of Macau

fshaoshuai0914, wangyan9509, thu.liubaodi, g@gmail.com

baiyu_upc@163.com, yicongzhou@um.edu.mo

Abstract

Open-World Few-Shot Learning (OFSL) is a critical field of research, concentrating on the precise identification of target samples in environments with scarce data and unreliable labels, thus possessing substantial practical significance. Recently, the evolution of foundation models like CLIP has revealed their strong capacity for representation, even in settings with restricted resources and data. This development has led to a significant shift in focus, transitioning from the traditional method of “building models from scratch” to a strategy centered on “efficiently utilizing the capabilities of foundation models to extract relevant prior knowledge tailored for OFSL and apply it judiciously”. Amidst this backdrop, we unveil the Direct-and-Inverse CLIP (DeIL), an innovative method leveraging our proposed “Direct-and-Inverse” concept to activate CLIP-based methods for addressing OFSL. This concept transforms conventional single-step classification into a nuanced two-stage process: initially filtering out less probable categories, followed by accurately determining the specific category of samples. DeIL comprises two key components: a pre-trainer (frozen) for data denoising, and an adapter (tunable) for achieving precise final classification. In experiments, DeIL achieves SOTA performance on 11 datasets.

<https://github.com/The-Shuai/DeIL>

1. Introduction

Few-shot learning (FSL) [12, 33, 35, 40] has progressed substantially in data-limited research yet confronts formidable practical challenges, mainly due to its overly simplified conditions that frequently presume flawless labels while disregarding prevalent noise and errors. To overcome this, [1] introduced Open-World Few-Shot Learning (OFSL), an extension of FSL aimed at enabling efficient identification even in the presence of noisy labels originating

Figure 1. An example to introduce the concept of Direct-and-Inverse. It is a 4-way classification task. The traditional method directly categorizes the data (Upper). Our method, however, splits this into a two-step process (Lower). The first phase filters out the less probable classes, simplifying the task from 4-way to 3-way classification. The second phase then precisely identifies the category of samples. This two-step approach streamlines the decision-making process by narrowing down choices, consequently decreasing the likelihood of misclassification and enhancing overall accuracy. We evaluate the concept's efficiency in Sec. 5.4.

ing from both known and unknown categories within the training data.

Compared to traditional weakly supervised learning with mixed noisy and clean labels, and unsupervised learning without any labels, OFSL encounters greater challenges due to its unique circumstances. Specifically, with limited training samples, especially only one per category, the negative impact of incorrect labeling on the model is more severe than having no labels at all. Recent advances in OFSL have been made through techniques like metric learning [1] and feature aggregation [20], yet these methods still face hurdles, particularly when dealing with a high prevalence of noisy labels. This underscores the urgent need to devise more robust methods to surmount these challenges.

Recently, foundation models [5, 6, 26] have become increasingly influential. These models, pre-trained on expansive datasets, boast robust architectures that offer strong

* Corresponding author.

Figure 2. The flowchart of our Direct-and-Inverse CLIP (DeIL). The complete procedure encompasses three distinct steps: (1) The 3-way 1-shot support data initially undergoes processing by the DeIL-Pretrainer, aimed at correcting noisy labels. (2) The redefined data is directed through DALL-E, which facilitates the generation of extra training data, thereby broadening its diversity. (3) Both the corrected and augmented data are subsequently fed into the DeIL-Adapter for executing the classification task. Only the DeIL-Adapter is tunable.

representational capabilities, even in scenarios with limited data and computational resources. This has prompted a strategic shift: instead of building models from scratch, there is a growing emphasis on leveraging the extensive potential and expertise of these pre-trained foundation models to address OFSL. Their advantages stem from extensive validation and optimization, enhancing their robustness against prevalent overfitting challenges in OFSL while also saving considerable time and computational resources.

Amidst this backdrop, we present the Direct-and-Inverse CLIP (DeIL), leveraging our “Direct-and-Inverse” concept (see Fig. 1) to activate CLIP-based methods for tackling OFSL. This concept transforms conventional classification tasks (reliant on a singular direct prediction step) into a more nuanced two-phase process. The initial phase effectively filters out the less probable categories, streamlining the decision-making process. Subsequently, the second phase accurately pinpoints the specific category of the samples. This two-phase approach enhances decision-making by methodically narrowing down choices, thereby lowering misclassification risks and boosting accuracy. We employ the remarkable CLIP [26] as the direct model for identifying likely categories, and its derivative, CLIPN [31], as the inverse model for ruling out unlikely categories.

DeIL is composed of two principal components and an auxiliary module (see Fig. 2), thoughtfully designed to address the complex OFSL: (1) Frozen DeIL-pretrainer leverages the Direct-and-Inverse concept to first pinpoint and then rectify noisy labels. (2) Tunable DeIL-Adapter employs the same concept, integrating classification and contrastive learning loss to enable precise label prediction. (3) Frozen DALL-E serves to augment data, enriching the diversity of samples. The synergy of these elements culminates in DeIL being a well-rounded and robust solution, adept at addressing the varied demands of OFSL.

Our main contributions are summarized as follows:

- We present DeIL, a method that ingeniously applies the Direct-and-Inverse concept to capitalize on the intrinsic capabilities and prior knowledge of CLIP-based methods, substantially boosting the performance of OFSL.
- We unveil the DeIL-Pretrainer and DeIL-Adapter, tailored mechanisms for OFSL to strategically minimize the adverse effects of noisy labels.
- Our thorough evaluations of DeIL on 11 benchmark datasets highlight its significant advancement and superiority over current state-of-the-art (SOTA) methods.

2. Related Work

Foundation Models Recently, research on foundation models is in full swing. Here, we introduce three models used in our paper. CLIP [26] is a multi-faceted model that bridges vision and language by aligning image and text representations within a shared latent space. It employs contrastive learning to enhance the correlation between compatible image-text pairs while reducing it for mismatched pairs. By being trained on extensive datasets comprising varied images and texts, CLIP acquires an understanding of concepts and relationships spanning different modalities. CLIPN [31] is a derivative model stemming from CLIP, mirrors its predecessor in terms of structure and training approaches. Its distinctive feature, however, lies in a specialized functionality. By modifying the prompt template, CLIPN alters the dynamic between image and text, offering insights into the likelihood that a certain sample falls outside a specific class. DALL-E [27] stands out as a potent model for generating a wide range of realistic images based on text descriptions. It operates on an encoder-decoder framework, where the encoder converts text descriptions into latent vectors. These vectors are subsequently utilized by the decoder to create the relevant images.

Open-World Few-Shot Learning Beginning in 2015, open-world object recognition [2, 3, 34] has emerged as an increasingly popular area of research. The initial focus was on managing extensive samples in open-world environments and effectively detecting open sets. However, more recent studies have shifted to addressing recognition issues in open scenarios with sparse data. This shift has brought about a heightened focus on the impact of noisy labels from both visible and invisible classes on results, leading to the development of OFSL. OFSL can be seen as a specialized form of weakly supervised learning under situations with limited samples available. It differs from robust few-shot learning [21], which is primarily concerned with label noise in visible classes, OFSL also accounts for noise from unseen classes. Departing from traditional solutions such as metric learning [1], instance reweighting [21], and feature aggregation [20], our paper introduces a groundbreaking method. Our method capitalizes on the direct-and-inverse concept to effectively engage foundation models, offering a new solution to the inherent challenges of OFSL.

Foundation Solutions on Few-Shot Learning In FSL, foundation models have driven significant progress, with key examples being CLIP [26] stands out as a robust model initially designed for zero-shot learning, and its utility has been extended effectively into FSL. CLIP [14] is a parameter-free method that enhances the zero-shot performance of CLIP by enabling visual and textual representations to interact through an attention mechanism, and its parametric solution achieves even higher accuracy in few-shot scenarios. CLIP-Adapter [13] is a re-tuning method for CLIP by applying lightweight residual-style adapters. APE [40] is an advanced method that reuses CLIP's adaptive priors, substantially boosting downstream task performance. It smartly navigates through class disparities and decouples domain-specific knowledge, delivering exceptional accuracy while maintaining computational efficiency. CaFo [38] combines GPT's linguistic prompts [5], DALL-E's synthetic images [27], and a learnable cache model. This integration enhances predictions by amalgamating outputs from both CLIP and DINO [6], leveraging a variety of pre-trained methods for superior performance. Additionally, numerous subsequent researches [8, 24, 28, 29, 36, 37, 39] have made substantial contributions, enriching the breadth and depth of the research community.

3. Problem Setup

Few-Shot Learning In the standard FSL, the base set, a large collection of labeled data, is used to pre-train a feature extraction model for downstream tasks. The novel set, containing all data for downstream tasks, consists of two key components: the support set S (and the query

set Q), with no overlap between them ($S \cap Q = \emptyset$). $S = \{f(x_i; y_i; t_i)g_{i=1}^{C \times K}\}$ contains a few labeled data points, each with an image $x_i \in \mathbb{R}^{2 \times 2 \times X}$, label $y_i \in \mathbb{Y}$, and category-name $t_i \in \mathbb{T}$. X , Y and T are the image, label and category-name sets. C is the number of classes, and K denotes the number of samples per class, known as K -shot. On the other hand, $Q = \{f(x_i; y_i; t_i)g_{i=1}^{C \times K + N_{qry}}\}$ is the to-be-tested data with N_{qry} samples. FSL's main goal is to classify Q 's categories using only a few support samples. Open-World Few-Shot Learning OFSL is more intricate yet valuable for practical applications than FSL, aiming to accurately identify query data categories under the premise that the support labels are subject to random contamination and lack reliability. In OFSL, the support set consists of clean and noisy samples, $S = \{S_{clean}; S_{noise}\}g_{i=1}^{C \times K}$, where $S_{clean} = \{f(x_i; y_i; t_i)g_{i=1}^{C \times K}\}$ and $S_{noise} = \{f(x_i; \hat{y}_i; \hat{t}_i)g_{i=1}^{C \times K}\}$. $\hat{y}_i \in \mathbb{Y}$ signifies a noisy label, indicating that the image x_i belongs to class y_i but is mislabeled as class \hat{y}_i . Meanwhile, $\hat{t}_i \in \mathbb{T}$ refers to the noisy category name. \mathbb{Y} and \mathbb{T} are collections of noisy labels and category names. The noise sources might be other categories in the support set or unseen classes, and it's uncertain during training which samples are affected by noise.

4. Methodology

4.1. Overview

Departing from traditional OFSL methodologies that typically depend on additional base data for improving feature extraction, our paper presents a forward-thinking approach. We introduce DeIL, an inventive method that employs the Direct-and-Inverse concept to effectively harness CLIP-based foundation models for OFSL. The training framework is depicted in Fig. 2, including 3 steps:

- Feeding the support data into the DeIL-Pretrainer (Fig. 3) to correct noisy labels. The DeIL-Pretrainer includes two frozen-parameter foundation models: CLIPN, which uses inverse reasoning to detect mislabeled data, and CLIP, which employs direct reasoning for label correction.
- Forwarding the rectified support samples to the frozen DALL-E model to further enrich the training data. Inputting both the rectified original data and the augmented data into our tailor-made DeIL-Adapter (Fig. 4) for the final classification. Initially, employing the direct concept, we combine CLIP and the adapter for label prediction and compute the classification loss. Subsequently, we bring in contrastive loss, choosing negative samples through the inverse concept and positive samples based on prior label predictions. Lastly, we update the adapter by integrating these two losses.

During the inference stage, only the classification phase within the DeIL-Adapter is required.

Figure 3. The flowchart of DeIL-Pretrainer. (1) Inverse-phase(Left): We feed all category text information into CLIPN's text encoder and all image data into its image encoder to extract the respective features. We then calculate the similarity between these features to gauge the probability of a sample not belonging to a specific category. This aids in pinpointing samples with noisy labels by comparing them against the provided labels. (2) Direct-phase(Right): All category information, alongside the identified noisy images, are fed into CLIP's text and image encoders. By assessing the similarity between the obtained features, we can effectively correct the noisy labels.

4.2. DeIL-Pretrainer for Label Correction

The DeIL-Pretrainer (see Fig. 3) is our developed concatenated label correction module, primarily consisting of two key components. It is a plug-and-play freeze module.

Noisy Label Identification (1) We employ an inverse prompt template to meticulously craft descriptions for the category names in the support set. The template can be structured as sentences like "photo without [CLASSNAME]". We denote the inverse prompt for noisy category names as $\text{Template}_{\text{inv}}(\hat{\mathbf{T}})$.

(2) Following this, we harness frozen CLIPN's text encoder to process these descriptions, yielding informative text representations for each class. This critical step captures intricate inverse textual information that accurately characterizes each category. Simultaneously, we extract features from the original support images using frozen CLIPN's image encoder, allowing us to capture the visual attributes corresponding to the inverse textual information. We formulate the processes as:

$$\hat{\mathbf{F}}_{\text{clipn_txt}} = \mathbf{M}_{\text{clipn_txt}} \text{Template}_{\text{inv}}(\hat{\mathbf{T}}) \quad (1)$$

$$\hat{\mathbf{F}}_{\text{clipn_img}} = \mathbf{M}_{\text{clipn_img}}(\mathbf{X}) \quad (2)$$

where $\mathbf{M}_{\text{clipn_txt}}$ and $\mathbf{M}_{\text{clipn_img}}$ denote the frozen CLIPN's text and image encoders, $\mathbf{X} \in \mathbb{R}^{CK \times \dim}$ is the collection of original images, $\hat{\mathbf{F}}_{\text{clipn_txt}} \in \mathbb{R}^{\hat{\mathbf{C}} \times \dim}$ signifies the features of textual category names, \dim denotes the dimension, and $\hat{\mathbf{C}}$ is the number of the noisy support label set, $\mathbf{F}_{\text{clipn_img}} \in \mathbb{R}^{\dim \times CK}$ indicates the original image features.

(3) Subsequently, we gauge the similarity between the text features derived from the class names and the image features obtained from the original support images by:

$$\hat{\mathbf{L}}_{\text{inv}} = (\hat{\mathbf{F}}_{\text{clipn_img}})^T \hat{\mathbf{F}}_{\text{clipn_txt}} \quad (3)$$

where $\hat{\mathbf{L}}_{\text{inv}} \in \mathbb{R}^{CK \times \hat{\mathbf{C}}}$ is the inverse logits, denotes the similarity matrix between the original images and the class

names, the element $\hat{L}_{\text{inv}}(i; c)$ represents the probability that the i -th sample does not belong to the c -th class.

(4) Afterward, we derive a mask by comparing the provided label with the corresponding value in $\hat{\mathbf{L}}_{\text{inv}}$. If the value exceeds the threshold, it is categorized as a noisy label. The Mask is defined as:

$$\text{Mask}(i; c) = \begin{cases} \text{noisy label} & \text{if } y_i = c \text{ and } \hat{L}_{\text{inv}}(i; c) \\ \text{ignore} & \text{otherwise} \end{cases} \quad (4)$$

(5) Finally, we identify the noisy images by:

$$\mathbf{X}_{\text{nsy}} = \text{Filter}(\mathbf{X}; \text{Mask}) \quad (5)$$

where $\mathbf{X}_{\text{nsy}} \in \mathbb{R}^{N_{\text{nsy}} \times \dim}$ represents the set of predicted noisy images, with N_{nsy} as their count, and Filter denotes the operation for selecting images with noisy labels.

Noisy Label Correction (1) We use a direct prompt template to generate elaborate descriptions for the category names in the support set. This template can take the form of sentences like "A photo of [CLASSNAME]", and we define it as $\text{Template}_{\text{dir}}(\hat{\mathbf{T}})$.

(2) Next, we utilize the text encoder from the frozen CLIP to produce precise and informative text representations for each class. Simultaneously, we extract features from the noisy images using the image encoder from the same frozen CLIP, enabling us to capture the visual characteristics that align with the textual information.

$$\hat{\mathbf{F}}_{\text{clip_txt}} = \mathbf{M}_{\text{clip_txt}} \text{Template}_{\text{dir}}(\hat{\mathbf{T}}) \quad (6)$$

$$\hat{\mathbf{F}}_{\text{clip_img}}^{\text{nsy}} = \mathbf{M}_{\text{clip_img}}(\mathbf{X}_{\text{nsy}}) \quad (7)$$

where $\mathbf{M}_{\text{clip_txt}}$ and $\mathbf{M}_{\text{clip_img}}$ denote the frozen CLIP's text and image encoders, $\hat{\mathbf{F}}_{\text{clip_txt}} \in \mathbb{R}^{\hat{\mathbf{C}} \times \dim}$ signifies the features of textual category names, $\hat{\mathbf{F}}_{\text{clip_img}}^{\text{nsy}} \in \mathbb{R}^{\dim \times N_{\text{nsy}}}$ indicates the features of predicted noisy images.

(3) Subsequently, we assess the similarity between the feature representations of noisy images and the textual features of class names in the support data. This similarity measurement establishes a meaningful correspondence between the textual descriptions and the visual content, enabling us to accurately refine and validate the labels associated with the support set samples.

$$\hat{L}_{dir} = (\hat{F}_{clipn img}^{nsy})^T \hat{F}_{clip txt} \quad (8)$$

where $\hat{L}_{dir} \in \mathbb{R}^{N_{nsy} \times C}$, the element $\hat{L}_{dir}(i; c)$ represents the probability that the i -th sample belongs to the c -th class.

(4) We finally use the forecasted labels to replace the original, erroneous ones:

$$Y; T = \text{Refinement}(\hat{Y}; \hat{T}; \hat{L}_{dir}) \quad (9)$$

where Y and T denote the sets of corrective labels and category names, respectively. Refinement represents a sampling process to correct labels.

4.3. DALL-E for Data Augmentation

We utilize the frozen DALL-E model to expand the data for OFSL, generating images based on corrected category names. This approach, by augmenting and diversifying support samples, addresses data scarcity in FSL, improving the model's generalization and performance on query data. The augmented data $X_{aug} \in \mathbb{R}^{N_{aug}}$ can be defined as:

$$X_{aug} = M_{dalle}(T) \quad (10)$$

where M_{dalle} denotes the frozen DALL-E model.

4.4. DeLL-Adapter for Classification

We develop a two-pronged method for final classification (see Fig. 4). Initially, we merge CLIP with the adapter for direct reasoning, obtaining predicted labels and classification loss. We then introduce contrastive learning, using CLIP's inverse reasoning to select negative samples and determining anchor and positive samples from previous label predictions for contrastive loss calculation. This technique successfully integrates the Direct-and-Inverse concept into the trainable adapter, significantly enhancing its performance and suitability for the inference stage.

Label Prediction We input both the corrected original data and the augmented data into the CLIP model to extract their direct-features. Following this, we calculate their similarity and denote the logits as L_{dir}^1 .

$$F_{clip txt} = M_{clip txt} \text{Template}_{dir}(T) \quad (11)$$

$$F_{clip img} = M_{clip img}(X; X_{aug}) \quad (12)$$

$$L_{dir}^1 = (F_{clip img})^T F_{clip txt} \quad (13)$$

Figure 4. The flowchart of DeLL-Adapter. Firstly, in the Direct-phase (Upper), we calculate the classification loss and determine the anchor and positive samples. Then, in the Inverse-phase (Lower), we select negative samples for calculating the contrastive loss. Finally, we update the adapter based on these two losses.

where $F_{clip txt} \in \mathbb{R}^{dim \times C}$ indicates the CLIP features of corrective textual category names, C represents the length of the corrective support label set, $F_{clip img} \in \mathbb{R}^{dim \times (CK + N_{aug})}$ denotes the original and augmented image features extracted by CLIP.

Next, the image features are fed into an adapter structure, which is constructed with a simple multi-layer perceptron (MLP). The output logits are represented as L_{dir}^2 .

$$L_{dir}^2 = M_{adapter}(F_{clip img}) \quad (14)$$

where $L_{dir}^2 \in \mathbb{R}^{(CK + N_{aug}) \times C}$.

Subsequently, we refer to [38] to merge the two sets of logits as the direct-logits. The purpose of this fusion is to retain a healthy skepticism regarding the corrected labels and so it's necessary to consider the zero-shot classification capability of CLIP as an essential reference in this process.

$$L_{dir} = L_{dir}^1 + e^{-(L_{dir}^1 - L_{dir}^2)} \quad (15)$$

where $L_{dir} \in \mathbb{R}^{(CK + N_{aug}) \times C}$; \odot denotes the Hadamard product; and α, β are the hyperparameters.

Finally, we compute the classification loss and predict the labels by:

$$\text{loss}_{cls} = \text{CrossEntropy}(\text{Softmax}(L_{dir})) \quad (16)$$

$$\text{label}_{pred} = \text{argmax}(\text{Softmax}(L_{dir})) \quad (17)$$

Contrastive Learning To further reduce the impact of noisy labels, we have developed a contrastive learning method that operates independently of given labels. Within each mini-batch, we define anchor samples and positive samples based on the predicted logits. We select the top N_{anc} samples with the highest confidence in each category as anchor samples, while the samples with confidence ranging from N_{anc} to N_{pos} are designated as positive samples.

$$X_{anc}^c = f(x_i; y_i = c; \text{argsort}(L_{dir}^c[: N_{anc}]))g \quad (18)$$

$$X_{pos}^c = f(x_i; y_i = c; \text{argsort}(L_{dir}^c[N_{anc} : N_{pos}]))g \quad (19)$$

where X_{anc}^c and X_{pos}^c denote the anchor and positive samples of the c -th class within each mini-batch, respectively.

Following this, we use CLIPN for inverse reasoning to identify negative samples not belonging to the anchor class, which can be defined as:

$$F_{clipn_txt} = M_{clipn_txt} \cdot \text{Template}_{inv}(T) \quad (20)$$

$$F_{clipn_img} = M_{clipn_img}(X; X_{aug}) \quad (21)$$

$$L_{inv} = (F_{clipn_img})^T F_{clipn_txt} \quad (22)$$

$$X_{neg}^c = f(x_i; y_i \neq c; \text{argsort}(L_{inv}^c[: N_{neg}]))g \quad (23)$$

where $F_{clipn_txt} \in \mathbb{R}^{2 \times \text{dim}_C}$ indicates the CLIPN features of corrective textual category names, $F_{clipn_img} \in \mathbb{R}^{2 \times (\text{dim}_C + N_{aug})}$ denotes the original and augmented image features extracted by CLIPN, $L_{inv} \in \mathbb{R}^{(CK + N_{aug}) \times C}$ is the inverse logits, X_{neg}^c denotes the negative samples that do not belong to the c -th class within each mini-batch, N_{neg} is the number of negative samples.

Then, InfoNCE [17] loss is defined as:

$$\text{loss}_{nce} = \frac{1}{N} \sum_i \log \frac{e^{h(x_i; X_{pos})}}{\sum_j e^{h(x_i; X_{pos}(j))} + \sum_j e^{h(x_i; X_{neg}(j))}} \quad (24)$$

where $X_{pos}(i)$ and $X_{neg}(i)$ represents the positive and negative samples of anchor h ; \cdot denotes the operation to compute the cosine similarity; τ is the temperature.

Loss Function and Inference The total loss is defined as:

$$\text{loss} = \text{loss}_{cls} + \text{loss}_{nce} \quad (25)$$

where τ is the hyperparameter. In the inference stage, we can accomplish classification by exclusively relying on Eq. (17), without the necessity of considering the contrastive learning branch.

5. Experiments

5.1. Settings

Datasets Our method is tested across 11 renowned and publicly accessible datasets: ImageNet [9], OxfordPets [25], Caltech101 [11], DTD [7], Food101 [4], Sun397 [32], UCF101 [30], EuroSAT [16], FGVC [22], Flower102 [23], and StanfordCars [19]. We adopted the setting used in CaFo [38] and APE [40], training our models with 1, 2, 4, 8, and 16 labeled samples per class from the support set, and then evaluating them on the complete query set. To mimic real-world conditions, we introduced different levels of noisy labels into the support data for each dataset.

Comparison Methods We compare foundation model based methods leveraging frozen foundation models with adapters for fine-tuning, including CLIP [26], CoOp [39], Tip-Adapter [37], CLIP-Adapter [13], CALIP-FS [14], CaFo [38], and APE-T [40].

Implementation Our methodology adeptly integrates the capabilities of CLIP [26], CLIPN [31], and DALL-E [27]. CLIP and CLIPN are utilized as feature extractors for extracting direct and inverse features, respectively. We use ResNet-50 [15] as the backbone for CLIP, and ViT-B-16 [10] as the backbone for CLIPN. DALL-E is key in creating category-specific images, adhering to the design ethos of CaFo [38]. Within the DeLL-Adapter, the MLP includes two linear layers, initialized through the Kaiming initialization. We begin with a learning rate of 0.001, using AdamW [18] for optimization and CosineAnnealingLR as our learning rate scheduler. Our training process encompasses data augmentation methods like random cropping, flipping, and normalization, executed with a batch size of 256 across 40 epochs. During testing, the batch size is adjusted to 64.

5.2. Performance

On ImageNet Tab. 1 and Fig. 5 (left) display results under 1-shot conditions with varied noisy label proportions, and Fig. 5 (right) shows outcomes with a consistent noisy label ratio of 0:3 and assorted sample sizes per class. These results yield several key insights: (1) In open-world environments, DeLL stands out among SOTA methods, achieving exceptional performance, even with a noise ratio as high as 100%. Moreover, it strikes a balance between computational efficiency and high-level performance. (2) In a testing environment with a 0:3 noise ratio, DeLL notably achieves a success rate of 22.28% in the 1-shot setting, exceeding the performance of comparable methods in settings up to 10 shots. (3) Most methods fail to show notable performance gains with increasing of support data due to the detrimental effects of noisy labels, but DeLL stands out for its exceptional ability to resist such disruptions. These results substantiate DeLL as a highly reliable and effective approach, adeptly addressing the inherent challenges in OFSL.

On Other Datasets To thoroughly assess the resilience of our DeLL model in various conditions, we undertook comprehensive evaluations using 10 additional datasets. The outcomes for datasets like OxfordPets, Caltech101, DTD, Food101, and Sun397 are depicted in Fig. 6, while the results for the other datasets are detailed in the Supplementary

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
Linear-probe CLIP (ICML'21) [26]	10min	22.29	19.25	16.51	13.98	11.48	9.49	7.30	5.28	3.43	1.61	0.08
CoOp (ICCV'22) [39]	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 (ECCV'22) [37]	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 (ICCV'23) [13]	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 (AAAI'23) [14]	20min	61.35	58.07	57.56	56.86	57.07	56.23	57.07	56.08	58.08	57.56	58.07
CaFo (CVPR'23) [38]	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 (ICCV'23) [40]	1min	62.50	58.43	57.00	51.02	54.04	52.90	51.72	51.53	51.17	50.80	50.20
DeLL (Ours)	12min	62.62	62.37	62.16	62.28	62.37	62.42	62.33	62.27	62.27	62.11	62.02

Table 1. 1-shot accuracy (%) of methods on ImageNet. DeLL indicates Foundation Model. Time is the training time on one A100 GPU.

Adapters	Noisy Label				
	0.1	0.3	0.5	0.7	0.9
w/o Adapter (Linear-probe)	21.30	17.45	15.32	10.66	15.97
Tip-Adapter	60.88	59.51	60.37	59.29	57.13
CLIP-Adapter	59.60	56.73	53.44	49.17	33.64
DeLL-Adapter	62.37	62.28	62.42	62.27	62.11

Table 3. Ablation study (%) of different adapters on ImageNet with 1-shot case. All comparison methods adopt our model architecture but utilize different adapters.

Figure 5. Comparison performance (%) on ImageNet. The left column denotes the results under 1-shot setting with varying proportions of noisy labels. The right column denotes the results under a fixed noisy label ratio of 0.3 on different few-shot settings. Automatically displayed in Tab. 2, 3.

DALL-E	DeLL-Pretrainer		DeLL-Adapter		Noisy Label	
	ID	CRC	CST	CLS	0.3	0.7
↖	X	X			61.44	61.44
-	X		X	X	61.68	61.82
Ⓢ	X	X	X	X	61.82	60.57
-	X	X		X	61.95	62.22
◦		X	X	X	61.70	61.52
±	X	X	X	X	62.28	62.27

Table 2. Ablation study (%) of different blocks on ImageNet with 1-shot case. ID, CRC, CST, and CLS are short for Identification, Correction, Contrastive Learning, and Classification.

Material. Analysis of these results demonstrates that our method uniformly outperforms others in open-world scenarios across a range of datasets, which underscores its remarkable robustness. This superior performance is largely attributed to the synergistic integration of diverse foundational models and the strategic development of specialized adapters. The consistent outperformance of our method when compared to others emphasizes its unique effectiveness in addressing the challenges of OFSL.

5.3. Ablation Study

We carry out detailed ablation studies to ascertain the efficiency of different components, and the findings are systematically displayed in Tab. 2, 3.

DeLL-Pretrainer The DeLL-Pretrainer, crucial for rectifying noisy labels, markedly enhances the efficacy of our pipeline. As illustrated in Tab. 2, integrating the DeLL-Pretrainer as a supportive component results in a significant performance boost of at least 0.5% (comparison between lines ↖ and ±). This underscores its utility when used in tandem with other elements of the pipeline. Additionally, the DeLL-Pretrainer uniquely integrates an inverse concept, forming an identification block that complements the correction process. The analysis of lines ↖ and ± reveals this block's vital contribution, accounting for at least a 0.5% increase in performance. Moreover, while the DeLL-Pretrainer is capable of acting as an independent classification block, its sole use for this purpose, as shown in line ↖, leads to suboptimal results. This highlights the effectiveness of using the DeLL-Pretrainer in conjunction with other components in the pipeline for optimal performance.

DeLL-Adapter The DeLL-Adapter, expertly tailored for OFSL, combines essential classification with supportive contrastive learning. The latter's effectiveness is clearly illustrated in Tab. 2 (lines ↖ and ±), where it shows performance improvements of around 0.1%. To further validate DeLL-Adapter's superiority, we compared it with established frameworks like Tip-Adapter and CLIP-Adapter, and also with adapter-free setups. These comparison results are detailed in Tab. 3. We observe that the DeLL-Adapter

Figure 6. Comparison performance (%) on other datasets. The upper column presents the results under 1-shot conditions with different noisy label proportions, while the lower column presents the results with a fixed noisy label proportion of 0.3 on varying few-shot settings.

consistently outperforms other adapters, delivering impressive performance gains of 5% across various noise scenarios. These findings emphasize the remarkable attributes of the DeIL-Adapter, demonstrating its capacity to enhance performance and maintain stability under noisy conditions.

DALL-E DALL-E plays a crucial role in augmenting the diversity of samples. A comparative analysis of life and \pm reveals that integrating DALL-E results in a notable improvement of approximately 6% in OFSL tasks.

5.4. Efficiency of Direct and Inverse Concept

The foundational idea of this study is to evolve traditional direct classification prediction into a dual-phase prediction process, harnessing the novel Direct-and-Inverse concept. Both the DeIL-Pretrainer and DeIL-Adapter are built on this concept, with each being assessed individually. The experiments of the DeIL-Pretrainer and DeIL-Adapter are detailed in Tab. 4 and Supplementary Material, respectively. Observing the Tab. 4: The Direct Correction is the method that labels are identified as noisy and subject to correction if the predicted probability exceeds 0.85 and does not align with the initially provided label, and the Direct-and-Inverse Correction follows Sec. 4.2. We conclude that the strategy of first identifying incorrect instances using the inverse concept, followed by label correction, is markedly more effective than direct correction alone, and robustly validates the effectiveness of the Direct-and-Inverse concept in refining classification methodologies.

6. Conclusion

To address the challenges of OFSL, we introduce DeIL, an advanced method utilizing the Direct-and-Inverse concept

Methods	Noisy Label				
	0.1	0.3	0.5	0.7	0.9
w/o Correction	100	300	500	700	900
Direct Correction	95	247	379	529	681
Direct-and-Inverse Correction	65	172	271	372	471

Table 4. The impact of label correction methods on ImageNet. The value denotes the number of 1-shot noisy data. Lower is better.

to activate prior knowledge within CLIP-based models. Our rigorous testing across 11 datasets validates DeIL’s effectiveness. Looking ahead, our efforts will be channeled into two key areas: (1) Broadening the application of OFSL to cover a more diverse range of practical tasks, going beyond the current research boundaries. This initiative is aimed at forging a more robust connection between academic research and tangible real-world implementations. (2) While acknowledging the strengths of foundation models, we intend to investigate the reasons behind their occasional shortcomings. This exploration is geared towards maximizing the untapped capabilities of these models, thereby enhancing their overall utility and impact.

Acknowledgments. This work was supported by the Science and Technology Development Fund, Macau SAR (No. 0049/2022/A1); the University of Macau (No. MYRG2022-00072-FST); the Major Basic Research Project in Shandong Province (No.ZR2023ZD32); the China Postdoctoral Science Foundation (No. 2023M743266); Zhejiang Provincial Postdoctoral Excellence Program (No. ZJ2023067). We thank GPT for polishing our paper.

References

- [1] Yuexuan An, Hui Xue, Xingyu Zhao, and Jing Wang. From instance to metric calibration: a unified framework for open-world few-shot learning. *IEEE TPAMI*, pages 9757–9773, 2023. 1, 3
- [2] Abhijit Bendale and Terrance Boulton. Towards open world recognition. In *CVPR*, pages 1893–1902, 2015. 3
- [3] Abhijit Bendale and Terrance E Boulton. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016. 3
- [4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014. 6
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, pages 1877–1901, 2020. 1, 3
- [6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hérnánsgou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 3
- [7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 6
- [8] Yawen Cui, Zitong Yu, Rizhao Cai, Xun Wang, Alex C Kot, and Li Liu. Generalized few-shot continual learning with contrastive mixture of adapters. *arXiv preprint arXiv:2302.05936*, 2023. 3
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 6
- [11] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR*, pages 178–178. IEEE, 2004. 6
- [12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017. 1
- [13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *IJCV*, 2023. 3, 6, 7
- [14] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. Calip: Zero-shot enhancement of clip with parameter-free attention. *AAAI*, pages 746–754, 2023. 3, 6, 7
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016. 6
- [16] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *STAR5*, 12(7):2217–2226, 2019. 6
- [17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, pages 18661–18673, 2020. 6
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 6
- [20] Kevin J Liang, Samrudhdi B Rangrej, Vladan Petrovic, and Tal Hassner. Few-shot learning with noisy labels. *ICPR*, pages 9089–9098, 2022. 1, 3
- [21] Jiang Lu, Sheng Jin, Jian Liang, and Changshui Zhang. Robust few-shot learning for user-provided data. *IEEE TNNLS*, 32(4):1433–1447, 2021. 3
- [22] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6
- [23] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008. 6
- [24] Kamalesh Palanisamy, Yu-Wei Chao, Xinya Du, Yu Xiang, et al. Proto-clip: Vision-language prototypical network for few-shot learning. *arXiv preprint arXiv:2307.03073*, 2023. 3
- [25] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. *CVPR*, pages 3498–3505, 2012. 6
- [26] 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 *ICML*, pages 8748–8763, 2021. 1, 2, 3, 6, 7
- [27] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ICML*, pages 8821–8831, 2021. 2, 3, 6
- [28] Jintao Rong, Hao Chen, Tianxiao Chen, Linlin Ou, Xinyi Yu, and Yifan Liu. Retrieval-enhanced visual prompt learning for few-shot classification. *arXiv preprint arXiv:2306.02243*, 2023. 3
- [29] Aniket Roy, Anshul Shah, Ketul Shah, Anirban Roy, and Rama Chellappa. Diffalign: Few-shot learning using diffusion based synthesis and alignment. *arXiv preprint arXiv:2212.05404*, 2022. 3
- [30] Khuram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6
- [31] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *ICCV*, pages 1802–1812, 2023. 2, 6
- [32] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene

recognition from abbey to zoo. *ICVPR* pages 3485–3492, 2010. [6](#)

- [33] Ji Zhang, Jingkuan Song, Lianli Gao, Ye Liu, and Heng Tao Shen. Progressive meta-learning with curriculum. *ICCV*, 32(9):5916–5930, 2022. [1](#)
- [34] Ji Zhang, Lianli Gao, Bingguang Hao, Hao Huang, Jingkuan Song, and Hengtao Shen. From global to local: multi-scale out-of-distribution detection. *TIP*, 2023. [3](#)
- [35] Ji Zhang, Lianli Gao, Xu Luo, Hengtao Shen, and Jingkuan Song. Deta: Denoised task adaptation for few-shot learning. In *ICCV*, pages 11541–11551, 2023. [1](#)
- [36] Ji Zhang, Shihan Wu, Lianli Gao, Hengtao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. *arXiv preprint arXiv:2309.07439*, 2023. [3](#)
- [37] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *ICCV*, 2022. [3](#), [6](#), [7](#)
- [38] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. *ICVPR* pages 15211–15222, 2023. [3](#), [5](#), [6](#), [7](#)
- [39] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *ICCV*, 130(9):2337–2348, 2022. [3](#), [6](#), [7](#)
- [40] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *ICCV*, 2023. [1](#), [3](#), [6](#), [7](#)