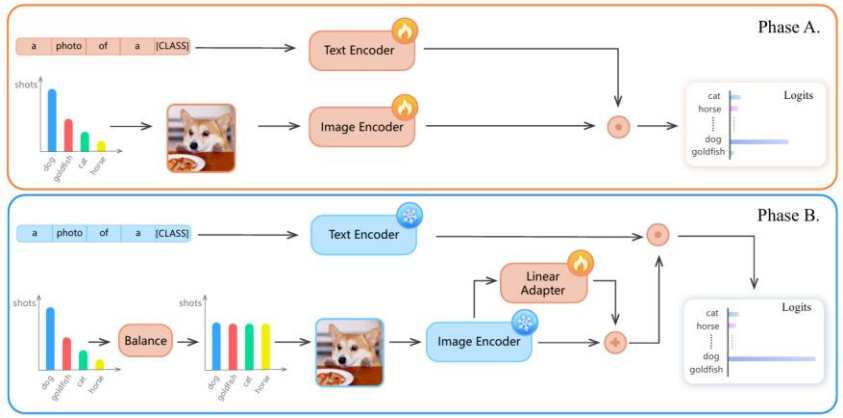
南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Long-Tail Learning with Foundation Model: Heavy Fine-Tuning Hurts

Jiang-Xin Shi [*1 2] Tong Wei [*3 4] Zhi Zhou [1] Jie-Jing Shao [1] Xin-Yan Han [1] Yu-Feng Li [1 2]

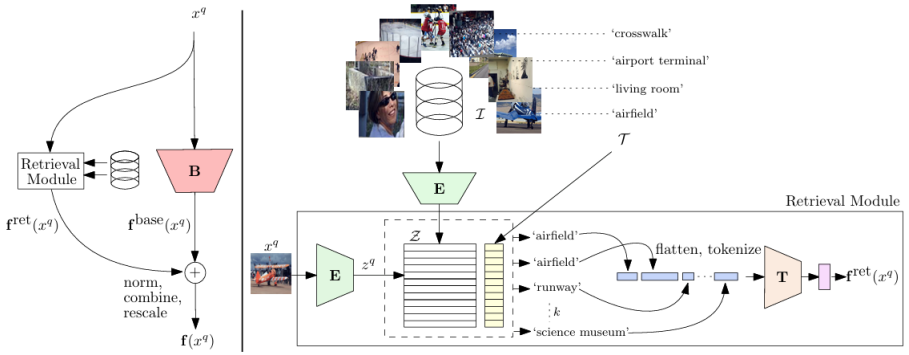
ICML 2024

Motivation



BALLAD first fully fine-tunes the foundation model, then freezes the backbone and optimizes a linear adapter on the re-sampled data.

[Arxiv 2021]A Simple Long-Tailed Recognition Baseline via Vision-Language Model



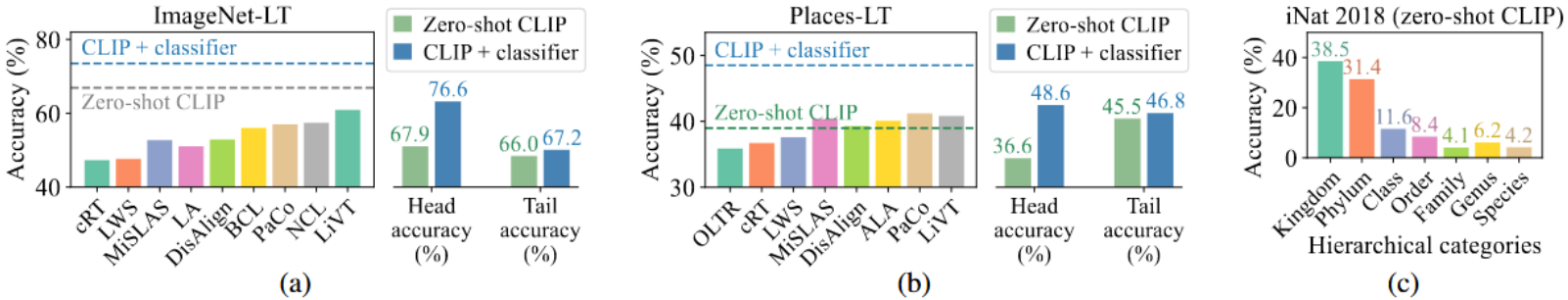
(a) The proposed RAC architecture (b) Retrieval module

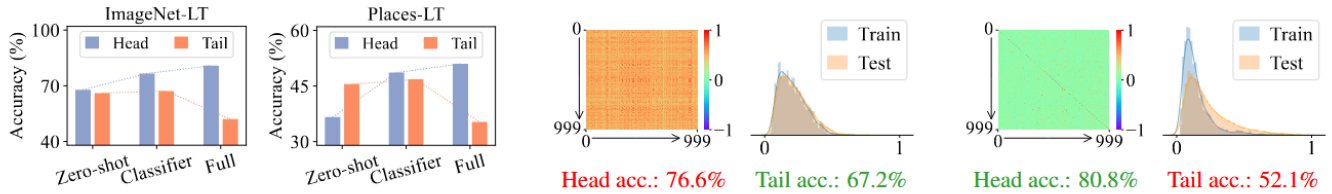
RAC jointly fine-tunes an encoder and trains a retrieval module to augment the input image with external datasets such as ImageNet-21K.

[CVPR 2022] Retrieval Augmented Classification for Long-Tail Visual Recognition

we reveal that heavy fine-tuning may lead to non-negligible performance deterioration on tail classes.

Motivation





(a) Performance of fine-tuning. (b) Classifier fine-tuning. (c) Full fine-tuning.

Figure 3: (a) Full fine-tuning improves head-class accuracy while decreasing tail-class accuracy, even if we optimize the balanced LA loss. (b-c) Inter-class feature similarities (heatmaps) and intra-class distributions from tail classes (histograms) on ImageNet-LT. Classifier fine-tuning limits head-class performance due to unoptimized inter-class similarities. Full fine-tuning optimizes inter-class similarities but leads to inconsistent distribution between train and test data on tail classes.

Indicates that the performance deterioration of full fine-tuning is attributed to the inconsistent class-conditional distributions among the tail classes.

Previous works such as LA (Menon et al., 2021) assume that the class-conditional distribution is consistent between the source and target domains

we reveal that heavy fine-tuning may lead to non-negligible performance deterioration on tail classes.

Fully fine-tuning yields more discriminative representations (low similarity)
However, it also distorts the intra-class distributions.

Motivation

$$\mathcal{L}_{\mathrm{LA}}(\boldsymbol{x}, y=j) = -\log \mathrm{P}_s(y=j \mid \boldsymbol{x}) = -\log \frac{\mathrm{P}_t(y=j \mid \boldsymbol{x}) \cdot \mathrm{P}_s(y=j) \cdot \zeta_{s-t}(j)}{\sum_{k\in[K]} \mathrm{P}_t(y=k \mid \boldsymbol{x}) \cdot \mathrm{P}_s(y=k) \cdot \zeta_{s-t}(k)}$$
$$= -\log \frac{\exp\left(z_j + \log \mathrm{P}_s(y=j) + \log \zeta_{s-t}(j)\right)}{\sum_{k\in[K]} \exp\left(z_k + \log \mathrm{P}_s(y=k) + \log \zeta_{s-t}(k)\right)} \tag{13}$$

where $\zeta_{s-t}(k)$ denotes $\dfrac{\mathrm{P}_s(\boldsymbol{x} \mid y=k)}{\mathrm{P}_t(\boldsymbol{x} \mid y=k)}$.

Indicates that the performance deterioration of full fine-tuning is attributed to the inconsistent class-conditional distributions among the tail classes.

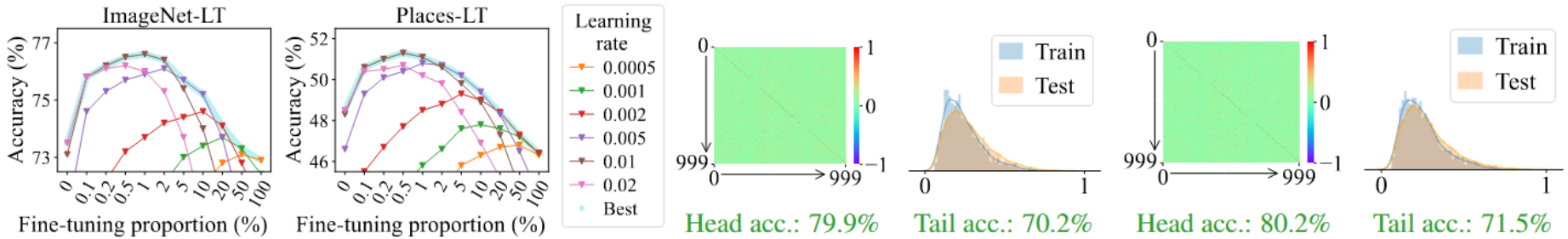
Previous works such as LA (Menon et al., 2021) assume that the class-conditional distribution is consistent between the source and target domains

Full fine-tuning also tends to encounter severe overfitting on long-tail datasets, particularly on the tail classes.

(a) ImageNet-LT.

Methods	Overall			Head			Medium			Tail		
	train	test	Δ	train	test	Δ	train	test	Δ	train	test	Δ
Full fine-tuning (*best lr*)	91.6	72.9	18.7	92.3	80.8	11.5	88.5	72.4	16.1	72.8	52.1	20.7
Full fine-tuning (*lr equal to LIFT*)	91.6	61.7	29.9	91.8	70.3	21.5	91.3	60.0	21.3	86.0	43.4	42.6
Classifier (*best lr*)	86.1	73.5	12.6	83.2	76.6	6.6	86.9	72.8	14.1	91.7	67.2	24.5
Classifier (*lr equal to LIFT*)	82.8	73.1	9.7	82.6	77.0	5.6	83.8	73.1	10.7	79.3	61.6	17.7
LIFT (Ours)	87.1	77.0	10.1	86.8	80.2	6.6	87.9	76.1	11.8	88.9	71.5	17.4

Method/Lightweight Fine-Tuning Helps



(a) Fine-tuning with varying proportions. (b) Arbitrary lightweight fine-tuning. (c) Structured lightweight fine-tuning.

Figure 4: (a) Fine-tuning a small proportion of all parameters (*e.g.*, 0.1%-2%) yields superior performance. As the proportion increases, performance deteriorates even when we search for the best learning rate. (b-c) Inter-class feature similarities (heatmaps) and intra-class distributions from tail classes (histograms) on ImageNet-LT. Both arbitrary and structured lightweight fine-tuning perform well in optimizing inter-class similarities and preserving intra-class distributions.

$$\boldsymbol{X}\boldsymbol{W} \rightarrow \boldsymbol{X}(\boldsymbol{W} \circ \boldsymbol{M}) + \underbrace{\boldsymbol{X}(\boldsymbol{W} \circ (1 - \boldsymbol{M}))}_{\text{gradient detached}} \qquad (4)$$

Arbitrary: the optimized parameters are selected arbitrarily
Structured: task-specific parameters (LoRA, Adapter)

Method

Semantic-Aware Initialization:
compute prompt features (num_classes, dim), which are then employed to initialize the classifier weights

Test-Time Ensembling:

$$\boldsymbol{z} = \log \mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{M} \sum_{i=1}^{M} \log \mathrm{P}(\boldsymbol{y} \mid \alpha_i(\boldsymbol{x})) \qquad (6)$$

Experiment

Table 1: Comparison with state-of-the-art methods on ImageNet-LT.

Methods	Backbone	Learnable Params.	#Epochs	Overall	Head	Medium	Tail
Training from scratch							
cRT (Kang et al., 2020)	ResNet-50	23.51M	90+10	47.3	58.8	44.0	26.1
LWS (Kang et al., 2020)	ResNet-50	23.51M	90+10	47.7	57.1	45.2	29.3
MiSLAS (Zhong et al., 2021)	ResNet-50	23.51M	180+10	52.7	62.9	50.7	34.3
LA (Menon et al., 2021)	ResNet-50	23.51M	90	51.1	-	-	-
DisAlign (Zhang et al., 2021)	ResNet-50	23.51M	90	52.9	61.3	52.2	31.4
BCL (Zhu et al., 2022)	ResNet-50	23.51M	100	56.0	-	-	-
PaCo (Cui et al., 2021)	ResNet-50	23.51M	400	57.0	-	-	-
NCL (Li et al., 2022a)	ResNet-50	23.51M	400	57.4	-	-	-
LiVT (Xu et al., 2023)	ViT-B/16	85.80M	100	60.9	73.6	56.4	41.0
Fine-tuning foundation model							
BALLAD (Ma et al., 2021)	ViT-B/16	149.62M	50+10	75.7	79.1	74.5	69.8
Decoder (Wang et al., 2024)	ViT-B/16	21.26M	∼18	73.2	-	-	-
LIFT (Ours)	ViT-B/16	**0.62M**	**10**	77.0	80.2	76.1	71.5
LIFT w/ TTE (Ours)	ViT-B/16	**0.62M**	**10**	**78.3**	**81.3**	**77.4**	**73.4**
Fine-tuning with extra data							
VL-LTR (Tian et al., 2022)	ViT-B/16	149.62M	100	77.2	84.5	74.6	59.3
GML (Suh & Seo, 2023)	ViT-B/16	149.62M	100	78.0	-	-	-

Table 2: Comparison with state-of-the-art methods on Places-LT.

Methods	Backbone	Learnable Params.	#Epochs	Overall	Head	Medium	Tail
Training from scratch (with an ImageNet-1K pre-trained backbone)							
OLTR (Liu et al., 2019)	ResNet-152	58.14M	30	35.9	44.7	37.0	25.3
cRT (Kang et al., 2020)	ResNet-152	58.14M	90+10	36.7	42.0	37.6	24.9
LWS (Kang et al., 2020)	ResNet-152	58.14M	90+10	37.6	40.6	39.1	28.6
MiSLAS (Zhong et al., 2021)	ResNet-152	58.14M	90+10	40.4	39.6	43.3	36.1
DisAlign (Zhang et al., 2021)	ResNet-152	58.14M	30	39.3	40.4	42.4	30.1
ALA (Zhao et al., 2022)	ResNet-152	58.14M	30	40.1	43.9	40.1	32.9
PaCo (Cui et al., 2021)	ResNet-152	58.14M	30	41.2	36.1	47.9	35.3
LiVT (Xu et al., 2023)	ViT-B/16	85.80M	100	40.8	48.1	40.6	27.5
Fine-tuning foundation model							
BALLAD (Ma et al., 2021)	ViT-B/16	149.62M	50+10	49.5	49.3	50.2	48.4
Decoder (Wang et al., 2024)	ViT-B/16	21.26M	∼34	46.8	-	-	-
LPT (Dong et al., 2023)	ViT-B/16	1.01M	40+40	50.1	49.3	52.3	46.9
LIFT (Ours)	ViT-B/16	**0.18M**	**10**	51.5	51.3	52.2	50.5
LIFT w/ TTE (Ours)	ViT-B/16	**0.18M**	**10**	**52.2**	**51.7**	**53.1**	**50.9**
Fine-tuning with extra data							
VL-LTR (Tian et al., 2022)	ViT-B/16	149.62M	100	50.1	54.2	48.5	42.0
RAC (Long et al., 2022)	ViT-B/16	85.80M	30	47.2	48.7	48.3	41.8

Table 3: Comparison with state-of-the-art methods on iNaturalist 2018.

Methods	Backbone	Learnable Params.	#Epochs	Overall	Head	Medium	Tail
Training from scratch							
cRT (Kang et al., 2020)	ResNet-50	23.51M	90+10	65.2	69.0	66.0	63.2
LWS (Kang et al., 2020)	ResNet-50	23.51M	90+10	65.9	65.0	66.3	65.5
MiSLAS (Zhong et al., 2021)	ResNet-50	23.51M	200+30	71.6	73.2	72.4	70.4
DiVE (He et al., 2021)	ResNet-50	23.51M	90	69.1	70.6	70.0	67.6
DisAlign (Zhang et al., 2021)	ResNet-50	23.51M	90	69.5	61.6	70.8	69.9
ALA (Zhao et al., 2022)	ResNet-50	23.51M	90	70.7	71.3	70.8	70.4
RIDE (Wang et al., 2021c)	ResNet-50	23.51M	100	72.6	70.9	72.4	73.1
RIDE+CR (Ma et al., 2023)	ResNet-50	23.51M	200	73.5	71.0	73.8	74.3
RIDE+OTmix (Gao et al., 2023)	ResNet-50	23.51M	210	73.0	71.3	72.8	73.8
BCL (Zhu et al., 2022)	ResNet-50	23.51M	100	71.8	-	-	-
PaCo (Cui et al., 2021)	ResNet-50	23.51M	400	73.2	70.4	72.8	73.6
NCL (Li et al., 2022a)	ResNet-50	23.51M	400	74.2	72.0	74.9	73.8
GML (Suh & Seo, 2023)	ResNet-50	23.51M	400	74.5	-	-	-
LiVT (Xu et al., 2023)	ViT-B/16	85.80M	100	76.1	**78.9**	76.5	74.8
Fine-tuning foundation model							
Decoder (Wang et al., 2024)	ViT-B/16	21.26M	**∼5**	59.2	-	-	-
LPT (Dong et al., 2023)	ViT-B/16	**1.01M**	80+80	76.1	-	-	79.3
LIFT (Ours)	ViT-B/16	4.75M	20	79.1	72.4	79.0	81.1
LIFT w/ TTE (Ours)	ViT-B/16	4.75M	20	**80.4**	74.0	**80.3**	**82.2**
Fine-tuning with extra data							
VL-LTR (Tian et al., 2022)	ViT-B/16	149.62M	100	76.8	-	-	-
RAC (Long et al., 2022)	ViT-B/16	85.80M	20	80.2	75.9	80.5	81.1

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Efficient and Long-Tailed Generalization for Pre-trained Vision-Language Model

Jiang-Xin Shi*
National Key Laboratory for Novel Software Technology
School of Artificial Intelligence
Nanjing University, China
shijx@lamda.nju.edu.cn

Chi Zhang*
National Key Laboratory for Novel Software Technology
Nanjing University, China
chi-zhang@smail.nju.edu.cn

Tong Wei
School of Computer Science and Engineering
Key Laboratory of Computer Network and Information Integration of Ministry of Education
Southeast University, China
weit@seu.edu.cn

Yu-Feng Li†
National Key Laboratory for Novel Software Technology
School of Artificial Intelligence
Nanjing University, China
liyf@lamda.nju.edu.cn

KDD 2024

Motivation

In down-stream task:

1. data may exhibit long-tailed data
2. There might be emerging tasks with new classes that contain no samples at all.

Train set: base classes (long-tailed)
Test set: base classes + new classes

Table 2: Empirical study results for zero-shot CLIP and visual prototypes over 11 datasets, using ViT-B/16 as the visual encoder. The visual prototypes are obtained by calculating the mean value of 16-shot features for each class and used subsequently to calculate cosine similarity with image features to get the classification scores.

	CAL.	OP.	SC.	Flw.	Food.	FA.	SUN.	DTD.	ES.	UCF.	IN.	Avg Results.
Zero-shot CLIP	89.3	**88.9**	65.6	70.4	**89.2**	27.1	65.2	46.0	54.1	69.8	**68.6**	66.7
Visual Prototypes	**93.4**	80.2	**71.7**	**95.9**	81.4	**41.3**	**69.8**	**64.2**	**75.2**	**78.1**	61.7	**73.9**
Δ	+4.1	-8.7	+6.1	+25.5	-7.8	+14.2	+4.6	+18.2	+21.1	+8.3	-6.9	+7.2

For instance, on the FGVCAircraft [21] dataset, the class names are different numeral versions such as '737-200' and '737-300', which hardly contain any useful information; or on the UCF101 [34] dataset, the image samples consist of frames from a video and do not precisely match the prompt templates such as 'a photo of a {class}'.

Method

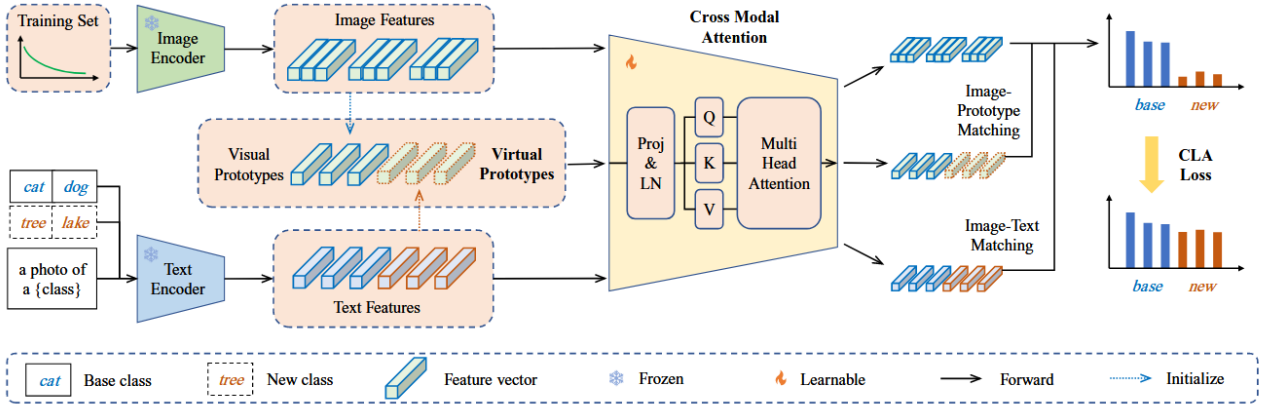


Figure 2: An overview of the proposed framework.

New classes has no samples in training stage: we introduce learnable virtual prototypes for new classes to hold the place of missing visual prototypes.

$$x', V', T' = \text{Attn}([P_I(x), P_I(V), P_T(T)]) \tag{5}$$

$$p(y = i \mid x) = \frac{\exp(\cos(P_I(x), P_T(T_i))/\tau_t)}{\sum_{j=1}^{N} \exp(\cos(P_I(x), P_T(T_j))/\tau_t)} \tag{4}$$

$$p_V(y = i \mid x) = \frac{\exp(\cos(x', V_i')/\tau_v)}{\sum_{j=1}^{K} \exp(\cos(x', V_j')/\tau_v)} \tag{6}$$

$$p_T(y = i \mid x) = \frac{\exp(\cos(x', T_i')/\tau_t)}{\sum_{j=1}^{K} \exp(\cos(x', T_j')/\tau_t)} \tag{7}$$

$$\mathcal{L} = \mathcal{L}_{cla}(z_P, y) + \mathcal{L}_{cla}(z_V, y) + \mathcal{L}_{cla}(z_T, y) \tag{9}$$

$$\mathcal{L}_{cla}(z, y = j) = -\log \frac{\exp(z_j + \log p(y = j))}{\sum_{k=1}^{K} \exp(z_k + \log p(y = k))} \tag{3}$$

Method

Table 9: Ablation on different loss functions. Δ indicates the difference in performance for the same method trained with CE loss or LA loss. Our method is the least sensitive to the change in loss function.

	Base	New	Harmonic Mean
CoOp + LA Loss	80.26	61.94	69.92
CoOp + CE Loss	76.12	61.19	67.84
Δ	-4.14%	-0.75%	-2.08%
CoCoOp + LA Loss	77.91	71.05	74.32
CoCoOp + CE Loss	72.05	65.36	68.54
Δ	-5.86%	-5.69%	-5.78%
Candle (Ours)	80.38	76.14	78.20
Candle w/ CE Loss	78.07	75.36	76.69
Δ	-2.31%	-0.78%	-1.51%

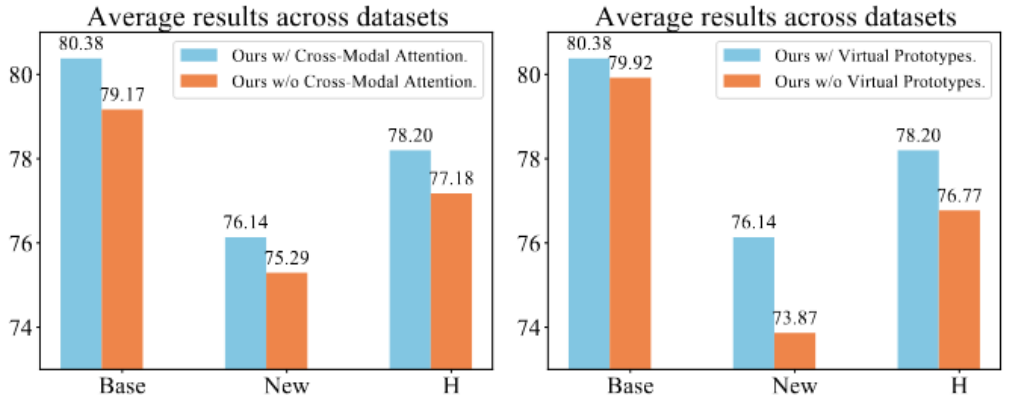


Figure 5: Ablation studies on cross-modal attention (left) and virtual prototypes (right). The experiment is conducted on the imbalanced base-to-new generalization task with an imbalance ratio of 50.

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Vision-Language Models are Strong Noisy Label Detectors

Tong Wei[1,2,3] **Hao-Tian Li**[1,2] **Chun-Shu Li**[1,2] **Jiang-Xin Shi**[3,4]
Yu-Feng Li[3,4] **Min-Ling Zhang**[1,2]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]National Key Laboratory for Novel Software Technology, Nanjing University, China
[4]School of Artificial Intelligence, Nanjing University, China
{weit,liht}@seu.edu.cn

NIPS 2024

Related works

Robust learning: the problem of maintaining a balance between robustness and accuracy
Clean sample selection: they inevitably overlook the significance of clean hard samples with a large loss

Introduction

Find the most effective methods for CLIP adaptation:

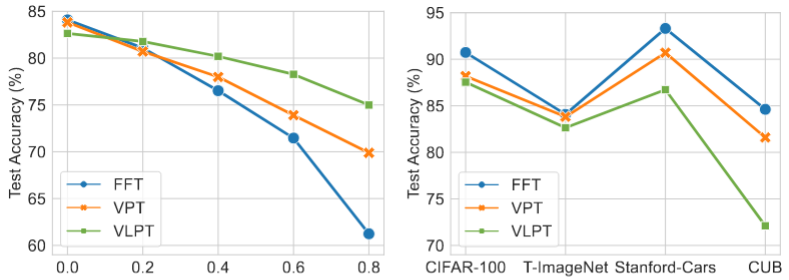
1. FFT (fully fine tuning) updates the entire model parameters
2. VPT (Vision-Prompt tuning) fixes pretrained model parameters and prepends a small extra learnable parameters to the visual encoder during fine-tuning
3. VLPT (Vision-Language Prompt Tuning) which integrates both visual and textual learnable prompts into the fixed pre-trained model for fine-tuning

For FFT and VPT, we learn an additional linear classifier, while VLPT directly utilizes the learned textual prompts for classification.

VPT benefits representation learning in the presence of massive noisy labels: As only a small set of parameters is introduced, VPT efficiently retains the generalization ability from image-text pre-training while enhancing classification performance on downstream tasks

Textual classifier is robust to noisy labels
The improvement in performance across diverse noise ratios further affirms the robustness of learnable textual prompts in mitigating the impact of label noise for model adaptation

FFT enhances visual recognition on clean datasets.
FFT improved performance by leveraging its substantial capacity to incorporate task-specific representations. VLPT exhibits the worst performance on clean datasets. This is primarily due to the implicit regularization of pre-trained textual information when tuning the context of textual prompts.



(a) Noisy Tiny-ImageNet (b) Clean Datasets

Figure 1: Comparison of different fine-tuning methods under (a) various ratios of noisy labels and (b) clean datasets.

Method

In the first phase, DEFT learns dual textual prompts to separate clean and noisy samples while adapting the visual encoder utilizing PEFT methods.
In the second phase, DEFT re-adapts the pre-trained model using FFT, leveraging the curated clean samples to further boost visual recognition performance.

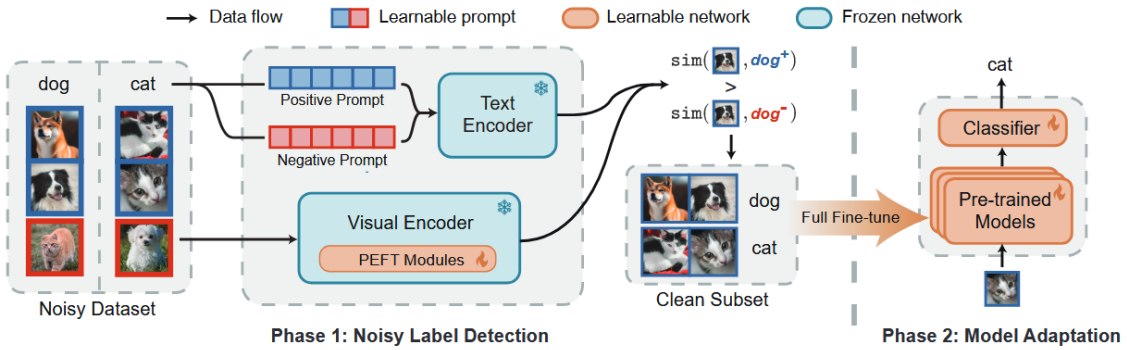


Figure 2: Illustration of the proposed DEFT framework. Left: We identify noisy labels with learnable dual textual prompts and improve image-text alignment by optimizing PEFT modules. Right: Adapt pre-trained models using FFT on selected clean samples.

positive prompt: maximizing the similarity between the image features and their corresponding text features.
negative prompt: serves as a learnable sample-dependent similarity threshold to select clean data

$$\phi_i = \mathtt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^-)$$

$$\mathcal{D}^{\text{clean}} = \{(\boldsymbol{x}_i, y_i) \mid \mathtt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^+) > \phi_i \text{ and } y_i = k\}$$

surpasses the conventional loss-based approaches in two aspects:

1. data-driven thresholds thus eliminating the requirement for prior knowledge like noise ratio
2. the integration of text modality enhances its robustness to label noise, making it capable of identifying challenging hard noise

the primary dilemma lies in the optimization of positive and negative prompts using noisy downstream datasets.

Method

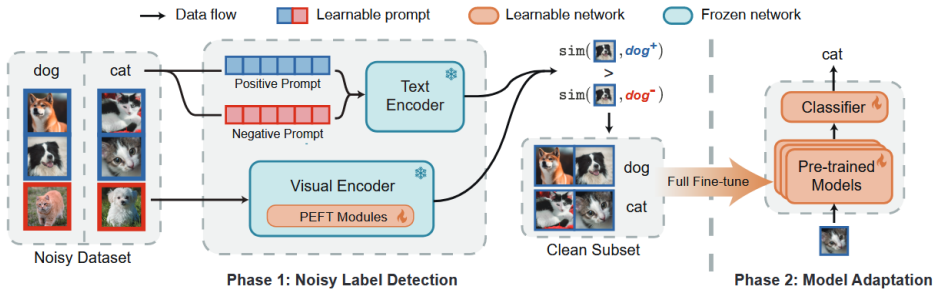
$$p_{ik}^{\text{clean}} = \frac{\exp(\texttt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^+)/\tau)}{\exp(\texttt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^+)/\tau) + \exp(\texttt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^-)/\tau)} \tag{6}$$

$$\mathcal{L}_{dp} = \frac{1}{N}\sum_{i=1}^{N} \ell_{nll}(\boldsymbol{p}_i^{\text{clean}}, \hat{y}) + \ell_{nll}(1 - \boldsymbol{p}_i^{\text{clean}}, \bar{y})$$

$$p(y = k \mid \boldsymbol{x}) = \frac{\exp(\texttt{sim}(\boldsymbol{I}, \boldsymbol{T}_k)/\tau)}{\sum_{k=1}^{K} \exp(\texttt{sim}(\boldsymbol{I}, \boldsymbol{T}_k)/\tau)},$$

we designate each image with a randomly picked complementary label $\bar{y}$ to form negative samples.

$$\mathcal{L}_{sim} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\texttt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_i^+)/\tau)}{\sum_{k=1}^{K} \exp(\texttt{sim}(\boldsymbol{I}_i, \boldsymbol{T}_k^+)/\tau)} \tag{8}$$



Experiment

Method	*Sym.* 0.2		*Sym.* 0.4		*Sym.* 0.6		*Ins.* 0.2		*Ins.* 0.3		*Ins.* 0.4	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
CIFAR-100												
Label-match	99.83	63.62	99.61	63.85	99.31	63.52	99.93	63.65	99.85	63.72	99.81	63.69
Small-loss	97.24	96.79	95.68	94.49	92.93	90.68	95.20	95.46	94.00	92.53	90.33	89.85
DEFT (ours)	**99.51**	**97.77**	**98.75**	**97.91**	**97.04**	**97.27**	**98.47**	**97.88**	**96.32**	**97.63**	**94.08**	**95.28**
Δ	↑ 2.27	↑ 0.98	↑ 3.07	↑ 3.42	↑ 4.11	↑ 6.59	↑ 3.27	↑ 2.42	↑ 2.32	↑ 5.10	↑ 3.75	↑ 5.43
Tiny-ImageNet												
Label-match	99.92	60.81	99.83	60.79	99.50	60.66	99.91	60.58	99.84	60.53	99.76	60.47
Small-loss	97.25	**96.93**	95.33	94.48	92.63	90.89	94.74	95.17	93.66	92.35	90.41	89.71
DEFT (ours)	**99.50**	96.00	**98.78**	**95.97**	**97.21**	**95.44**	**99.21**	**96.21**	**97.80**	**95.80**	**95.45**	**95.77**
Δ	↑ 2.25	↓ 0.93	↑ 3.45	↑ 1.49	↑ 4.58	↑ 4.55	↑ 4.47	↑ 1.04	↑ 4.14	↑ 3.45	↑ 5.04	↑ 6.06
Stanford-Cars												
Label-match	99.97	60.34	99.86	60.27	99.70	60.71	99.85	60.34	99.82	60.32	99.80	60.25
Small-loss	96.92	96.56	93.71	93.21	89.46	87.79	96.94	97.78	96.72	95.96	95.25	94.48
DEFT (ours)	**98.72**	**99.56**	**98.98**	**98.56**	**98.58**	**95.62**	**99.02**	**99.09**	**98.96**	**98.15**	**98.75**	**97.71**
Δ	↑ 1.80	↑ 3.00	↑ 5.27	↑ 5.35	↑ 9.12	↑ 7.83	↑ 2.08	↑ 1.31	↑ 2.24	↑ 2.19	↑ 3.50	↑ 3.23
CUB-200-2011												
Label-match	99.92	53.26	99.74	53.13	99.46	53.02	99.96	53.39	99.96	53.32	99.74	53.69
Small-loss	96.74	96.32	93.69	92.84	84.10	82.01	96.91	97.33	96.49	95.59	93.98	93.96
DEFT (ours)	**99.04**	**97.01**	**96.76**	**95.60**	**93.88**	**96.43**	**99.15**	**97.45**	**97.93**	**96.85**	**96.03**	**97.11**
Δ	↑ 2.30	↑ 0.69	↑ 3.07	↑ 2.76	↑ 9.78	↑ 14.42	↑ 2.24	↑ 0.12	↑ 1.44	↑ 1.26	↑ 2.05	↑ 3.15

Table 1: On each dataset, we compare the Precision (%) and Recall (%) of DEFT with CLIP label-match and small-loss to evaluate the clean sample selection performance. Δ is the difference between the performance of DEFT and small-loss.

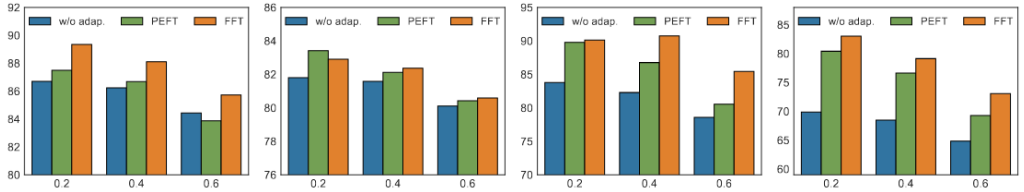
Experiment

	Method	*Sym.* 0.2	*Sym.* 0.4	*Sym.* 0.6	*Ins.* 0.2	*Ins.* 0.3	*Ins.* 0.4
				CIFAR-100			
FFT	CE	86.71 / 86.70	84.06 / 82.60	81.05 / 77.45	87.30 / 87.18	84.60 / 83.64	78.41 / 75.66
	ELR	86.53 / 86.53	83.66 / 83.66	78.34 / 78.34	86.61 / 86.61	85.89 / 85.89	**85.78 / 85.78**
	SCE	86.82 / 86.82	83.84 / 83.84	78.90 / 77.71	86.61 / 86.61	83.99 / 83.20	80.06 / 73.45
	GMM	88.49 / 88.49	87.21 / 87.21	85.22 / 85.20	88.44 / 88.44	87.95 / 87.95	82.14 / 82.11
DEFT	Ours	**89.38 / 89.35**	**88.17 / 88.11**	**85.81 / 85.72**	**89.38 / 89.35**	**88.68 / 88.68**	85.75 / 85.74
				Tiny-ImageNet			
FFT	CE	81.77 / 81.08	76.53 / 76.52	73.17 / 71.46	80.75 / 80.71	78.83 / 78.57	74.80 / 74.08
	ELR	79.40 / 79.40	77.13 / 77.13	73.74 / 73.74	79.98 / 79.98	77.13 / 77.13	73.74 / 73.74
	SCE	79.23 / 79.23	76.24 / 76.18	71.76 / 70.62	78.96 / 78.90	77.80 / 77.54	74.47 / 73.25
	GMM	81.91 / 81.88	80.37 / 80.37	43.47 / 43.47	81.84 / 81.79	81.26 / 81.26	79.01 / 79.01
DEFT	Ours	**82.91 / 82.91**	**82.48 / 82.37**	**80.60 / 80.59**	**83.37 / 83.33**	**82.69 / 82.65**	**80.52 / 80.49**
				Stanford-Cars			
FFT	CE	89.75 / 89.74	85.10 / 84.89	71.70 / 71.55	89.13 / 89.06	85.94 / 85.92	80.59 / 80.59
	ELR	86.61 / 86.61	76.98 / 76.98	61.58 / 61.58	84.40 / 84.40	83.11 / 83.11	75.97 / 75.84
	SCE	91.11 / 91.11	87.73 / 87.45	79.09 / 79.09	90.34 / 90.34	87.35 / 86.23	83.50 / 80.69
	GMM	90.10 / 90.08	83.14 / 83.10	56.90 / 56.90	88.15 / 88.10	85.39 / 85.33	78.76 / 78.72
DEFT	Ours	**92.13 / 92.12**	**90.75 / 90.75**	**85.72 / 85.45**	**92.19 / 92.15**	**90.77 / 90.77**	**89.74 / 89.68**
				CUB-200-2011			
FFT	CE	80.76 / 80.76	73.09 / 72.87	55.42 / 55.21	80.36 / 80.25	75.80 / 75.53	69.62 / 69.62
	ELR	77.70 / 77.70	68.26 / 68.26	50.17 / 49.88	78.32 / 78.32	73.16 / 73.08	63.57 / 63.34
	SCE	82.81 / 82.74	78.12 / 77.87	63.31 / 63.31	81.91 / 81.91	78.31 / 78.03	71.25 / 70.95
	GMM	75.79 / 75.73	64.39 / 64.38	42.84 / 42.84	75.73 / 75.65	69.95 / 69.95	56.13 / 55.80
DEFT	Ours	**83.05 / 83.03**	**79.24 / 79.13**	**73.08 / 73.08**	**82.53 / 82.50**	**81.39 / 81.39**	**79.34 / 79.24**

Table 2: Test accuracy (%) on synthetic datasets with *symmetric* and *instance-dependent* label noise.

Dataset	CE	ELR	SCE	GMM	RoLT	UNICON	LongReMix	ProMix	DEFT (Ours)
CIFAR-100N	72.41	72.83	72.52	76.06	75.91	77.68	73.94	75.97	**79.04**
Clothing1M	69.75	72.14	70.49	70.03	70.46	70.38	70.62	70.71	**72.44**
WebVision	84.64	79.32	82.88	84.88	84.12	84.56	84.96	84.44	**85.12**

Table 3: Test accuracy (%) on datasets with real-world label noise.



(a) CIFAR-100 (b) Tiny-ImageNet (c) Stanford-Cars (d) CUB-200-2011

Figure 3: Ablation studies. We report the test accuracy across varying noise ratios for the following variants: 1) **w/o adap.**: DEFT without the model adaptation phase, 2) **PEFT**: use PEFT for model adaptation phase, and 3) **FFT**: use FFT for model adaptation phase.

Architecture	CE	GCE	ELR	TURN	DEFT (Ours)
ResNet-50 [11]	66.02	66.19	66.19	66.31	**70.82**
MAE-ViT-B [10]	61.31	60.80	61.51	61.96	**65.23**
ViT-B/16 [5]	68.98	69.74	68.73	**70.28**	69.84
ConvNeXt-T [27]	68.80	68.92	68.52	69.53	**71.68**

Table 4: Test accuracy (%) using various pre-trained models on Clothing1M. Partial results are sourced from [1]. The best results across all methods are highlighted in bold, with the second-best results indicated by underscores.

Thank you