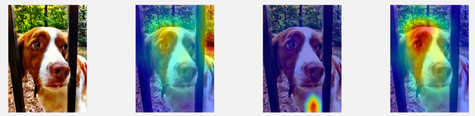
FD-Align: Feature Discrimination Alignment for Fine-tuning Pre-Trained Models in Few-Shot Learning

Kun Song, Huimin Ma†, Bochao Zou, Huishuai Zhang, Weiran Huang†

NEURAL INFORMATION PROCESSING SYSTEMS

Background & Motivation

CLIP shows remarkable performance in various visual tasks, but fully fine-tuning CLIP on limited data can lead to overfitting and poor out-of-distribution (OOD) generalization. After visualizing the attention of CLIP, it is found that fully fine-tuning make CLIP focus more on local features, which would undermine CLIP's robustness to spurious correlations and thus affect the OOD performance of CLIP. Our work focuses on ensuring the CLIP's robustness to spurious correlations during fine-tuning, thereby preserving the OOD generalization of CLIP.



(a) Original Image (b) CLIP (c) Fully Fine-Tuned CLIP (d) FD-Align

Method

Notation

- Pretrained visual encoder f_0
- Target visual encoder f_t
- Image x
- Prompt templates $(P_1, \cdots, P_M)$
- Pretrained text encoder g_0
- Proxy dataset $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$
- Label y
- Prompts $[P_1, y], \cdots, [P_M, y]$

Fine-tuning on Proxy Dataset

Using the feature of class name as prototypes of classes,

$$\mu_y^{\text{class}} := \frac{1}{M} \sum_{j=1}^{M} g_0([P_j, y]).$$

Calculating the cosine similarity between image feature and prototype and using the cross entropy loss as class loss,

$$\mathcal{L}_{\text{class}} = -\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \log \frac{\exp(s(f_t(x_i), \mu_{y_i}^{\text{class}}))}{\sum_{y \in \mathcal{Y}} \exp(s(f_t(x_i), \mu_y^{\text{class}}))},$$

Spurious Feature Constraint

In order to preserve the robustness to spurious correlation, we keep the spurious feature extracted by CLIP before and after fine-tuning consistent. Calculating the mean of the features of each prompt template P_i over all classes as prototypes of the spurious features,

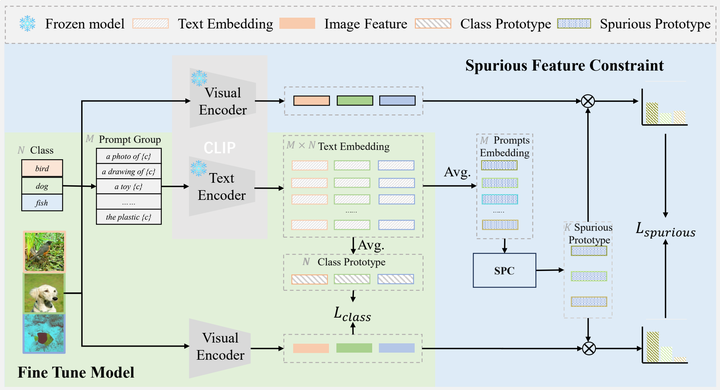
$$\mu_{P_j}^{\text{spurious}} := \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} g_0([P_j, y]).$$

Calculating the similarity between the feature extracted by the fine-tuned model and the spurious prototypes and produce the distribution over spurious prototypes as follows,

$$\mathcal{P}_{\text{spurious}}(x; f_t) = \text{SoftMax}\left[s\left(f_t(x), \mu_{P_1}^{\text{spurious}}\right), \ldots, s\left(f_t(x), \mu_{P_M}^{\text{spurious}}\right)\right].$$

Similarity, producing the distribution of the feature extracted by CLIP over spurious prototypes as follows,

$$\mathcal{P}_{\text{spurious}}(x; f_0) = \text{SoftMax}\left[s\left(f_0(x), \mu_{P_1}^{\text{spurious}}\right), \ldots, s\left(f_0(x), \mu_{P_M}^{\text{spurious}}\right)\right].$$



The class names and prompts are combined and inputted into the text encoder to obtain text embeddings. We calculate the mean separately in the prompt and class dimensions to derive the class prototype and prompt embedding. On the one hand, the image features are extracted using the fine-tuned visual encoder, and class distribution are calculated based on the class prototype to calculate the class loss. On the other hand, we use spurious prototype correction (SPC) module to correct the prompt embedding. By calculating the cosine similarity between the image features and the spurious prototype, we obtain the distribution over spurious features and calculate the spurious loss.

Keeping the distribution of the models over spurious features consistent before and after fine-tuning,

$$\mathcal{L}_{\text{spurious}} = \frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \text{KL}\left(\mathcal{P}_{\text{spurious}}(x_i; f_t) \,||\, \mathcal{P}_{\text{spurious}}(x_i; f_0)\right).$$

Optimizing the class loss and spurious loss during fine-tuning to ensure the classification ability and OOD robustness.

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{class}} + \beta \cdot \mathcal{L}_{\text{spurious}}.$$

Spurious Prototype Correlation

Removing unsuitable spurious features,

$$\mu^{spurious} := ISOLATIONFOREST(\mu^{spurious}, n).$$

Merging duplicate spurious features that arise from similar prompts,

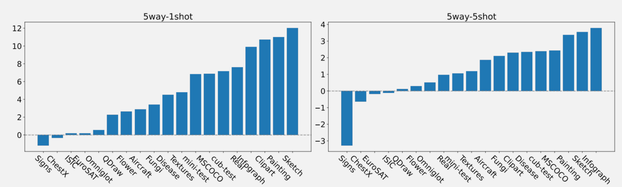
$$\mu^{spurious} := k - Means(\mu^{spurious}, k).$$

Experiments

OOD Results

Method	CLIP	Baselines					Baselines + FD-Align				
		FT	Tip	Tip-F	APE	APE-T	FT	Tip	Tip-F	APE	APE-T
ImageNet	63.34	64.91	65.49	68.43	66.55	68.74	**66.39**	**65.49**	**68.70**	**67.59**	**69.15**
ImageNetS	42.31	42.24	42.48	42.54	43.28	43.23	**43.50**	**43.84**	**43.67**	**44.23**	**44.04**
ImageNetV2	55,92	57.63	57.58	59.58	58.31	59.58	**57.73**	**59.10**	**60.17**	**59.36**	**60.83**

Fine-tuning the model on 16-shot ImageNet and evaluate it on the variants of ImageNet. FT means fully fine-tuning.

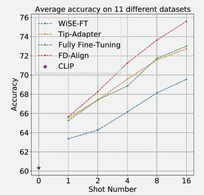


Performance improvement/decline of our method on different OOD datasets.

Datasets	5way-1shot			5way-5shot		
	CLIP	WiSE-FT	FD-Align	CLIP	WiSE-FT	FD-Align
Mini-test [43]	88.21±0.33	93.55±0.17	**95.04±0.18**	97.46±0.07	98.44±0.06	**98.52±0.07**
CUB-test [44]	75.21±0.78	81.16±0.71	**82.38±0.69**	91.48±0.34	93.41±0.32	**93.87±0.24**
Textures [38]	61.26±0.24	63.55±0.19	**66.05±0.12**	82.40±0.40	83.31±0.31	**83.60±0.34**
Traffic Signs [45]	58.51±0.11	**60.84±0.29**	57.32±0.26	76.67±0.18	**78.11±0.24**	73.39±0.29
Aircraft [40]	60.57±0.54	62.64±0.62	**63.45±0.65**	76.35±0.59	77.66±0.59	**78.21±0.58**
Omniglot [46]	83.27±0.37	83.56±0.28	**83.81±0.25**	94.29±0.13	**95.26±0.09**	94.81±0.19
VGG Flower [36]	90.88±0.31	**94.16±0.23**	93.50±0.24	98.65±0.10	**99.06±0.09**	98.95±0.09
MSCOCO [47]	62.30±0.38	67.28±0.32	**69.16±0.28**	78.93±0.38	81.08±0.35	**81.37±0.24**
Quick Draw [48]	62.22±0.61	62.54±0.59	**64.49±0.58**	82.65±0.31	82.78±0.37	**82.78±0.28**
Fungi [49]	50.42±0.32	53.10±0.27	**53.83±0.30**	71.59±0.18	73.28±0.10	**73.69±0.14**
Plant Disease [50]	70.64±0.28	**75.66±0.33**	75.13±0.33	89.50±0.24	91.78±0.31	**91.84±0.19**
ISIC [31, 51]	28.66±0.35	**29.40±0.34**	28.84±0.44	39.02±0.24	**39.54±0.40**	38.91±0.44
EuroSAT [39]	60.20±0.48	**63.99±0.39**	60.39±0.43	77.43±0.21	**80.96±0.19**	77.25±0.16
ChestX [52]	**22.65±0.27**	22.27±0.28	22.31±0.17	**25.58±0.08**	25.08±0.14	24.95±0.15
Real [32]	84.84±0.32	89.96±0.26	**92.45±0.28**	96.39±0.11	97.16±0.02	**97.36±0.04**
Sketch [32]	67.24±0.60	73.84±0.56	**79.27±0.38**	87.66±0.27	89.87±0.16	**91.20±0.19**
Infograph [32]	55.72±0.21	61.93±0.47	**65.61±0.17**	78.23±0.36	80.87±0.30	**82.02±0.37**
Painting [32]	68.05±0.18	74.92±0.33	**79.06±0.32**	87.99±0.21	90.26±0.22	**91.37±0.21**
Clipart [32]	75.14±0.12	81.55±0.26	**85.86±0.21**	92.52±0.09	94.06±0.13	**94.83±0.11**

The performance of model training on miniImageNet and evaluating on OOD datasets.

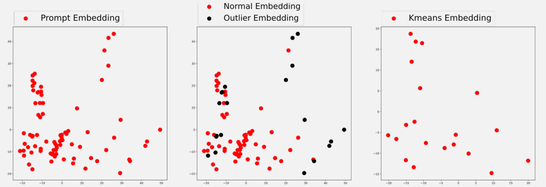
ID Results



Methods	1shot	2shot	4shot	8shot	16shot
Tip	64.11	64.36	64.63	65.17	65.49
Tip + **FD-Align**	**64.51**	**65.33**	**65.76**	**66.79**	**67.28**
Tip-F	64.64	65.18	65.78	67.21	68.43
Tip-F + **FD-Align**	**64.86**	**65.61**	**66.11**	**67.58**	**68.70**
APE	65.36	65.69	66.00	66.55	66.55
APE + **FD-Align**	**66.71**	**67.29**	**67.40**	**67.76**	**67.69**
APE-T	65.89	66.18	66.82	67.99	68.74
APE-T + **FD-Align**	**66.84**	**67.37**	**67.81**	**68.73**	**69.15**

Performance of different methods on 11 datasets. Results of using our backbone on different methods.

Visualization

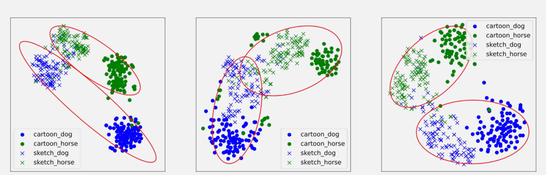


(a) All prompt templates (b) Isolation Forest (c) k-means

Visualization of spurious prototypes.

(a) CLIP (b) Fine-tuned CLIP (c) FD-Align CLIP

Feature visualization for the same class under different domains.



(a) CLIP (b) Fine-tuned CLIP (c) FD-Align CLIP

Feature visualization for different class under different domains.