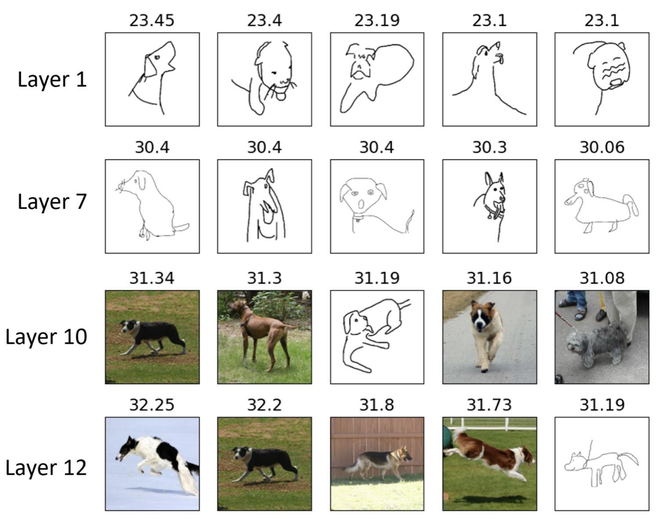
Efficient Transfer Learning driven by Layer-wise Features Aggregation

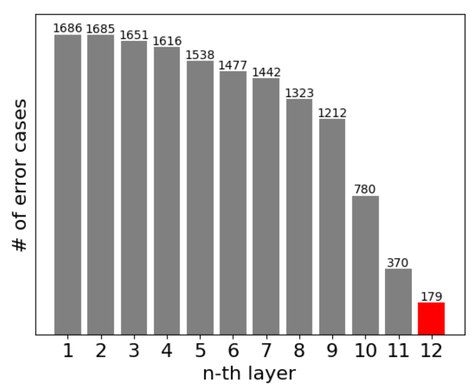
Chanwoo Kim[1] Jeyoon Yeom[2] Joowang Kim[2] Suho Kang[2] Kyungwoo Song[2]

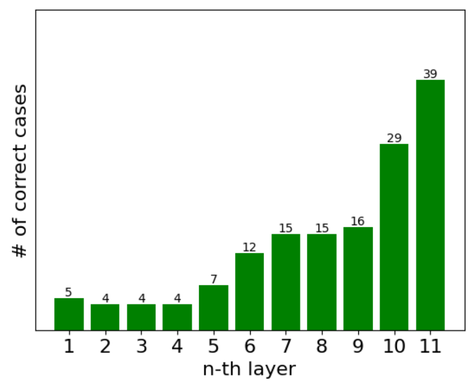
[1] University of Seoul [2] Yonsei University

Paper Code

Motivation: Are Lower-layers Useless?







Top Left: Top-5 most similar images with the text *'a photo of a dog playing on a ground.'*, shown in the n-th layer of the text encoder.
Top Right: Error-cases when using only each layer.
Bottom Right: Correct-case count in other layers for error-cases in the 12th layer.

- The features extracted from each layer differ, not only in image encoders but also in text encoders.
- Lower layers may contain their unique information that is not preserved in the upper layers.

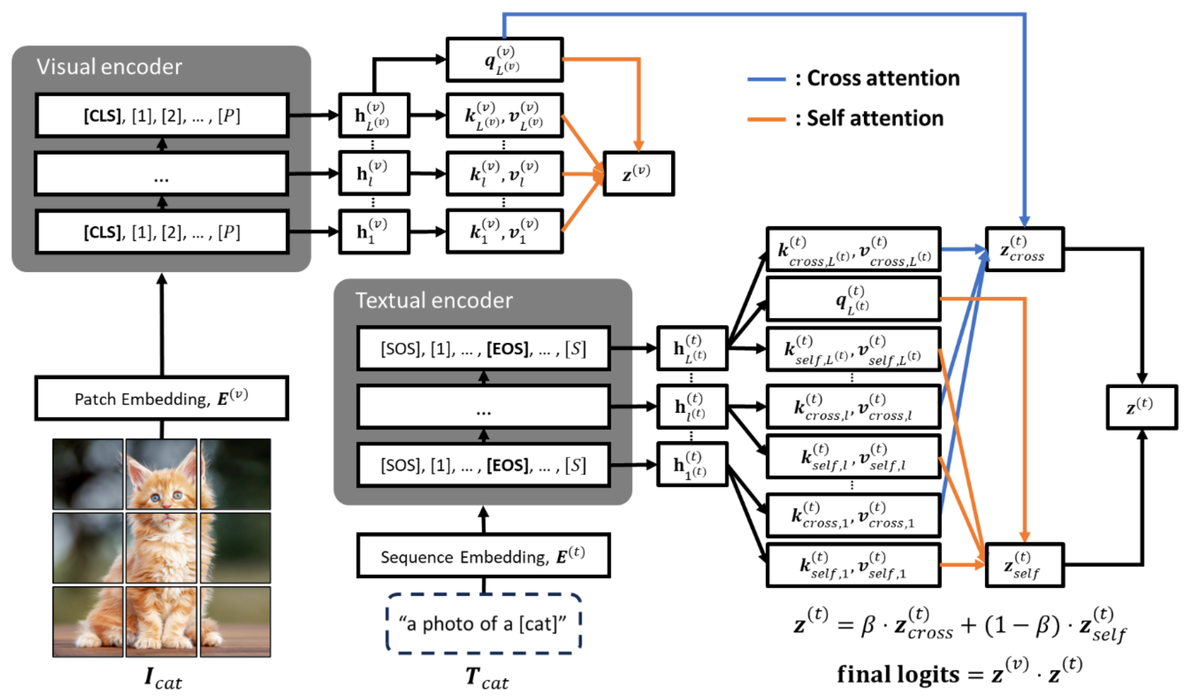
⬇

- By utilizing the lower layer features that have already been extracted, it is possible to enhance the model's generalization performance without incurring additional computational costs.

Overview

- We introduce the **L**ayer-wise **F**eature **A**ggregation (**LFA**) method, a novel approach that significantly mitigates the computational burdens commonly associated with the fine-tuning of pre-trained models.
- LFA **optimizes at the output feature level**, **reducing computational load** while **enhancing performance with richer aggregated features.**

Methodology: Layer-wise Feature Aggregation



- We propose a method that utilizes an attention mechanism to integrate unique features from lower layers with abstracted features from upper layers.
- This simple approach enables the utilization of richer features at a low computational cost.

Theoretical Analysis

Theorem 1. Principle of Rank Diminishing

For a neural network with L layers, we define the function f_l at each layer l as a product of weight matrices: $f_1 = W_1, f_2 = W_2W_1$, and generally, $f_n = W_n \cdots W_1$ for $n = 1, \cdots, L$. Therefore, basic default model $\widetilde{W} = \Pi_{l=1}^{n} W_l$. The rank of the intrinsic dimension monotonically decrease with depth: $rank(AB) \leq rank(A)\ and\ rank(AB) \leq rank(B), rank(\widetilde{W}) = \min(rank(W_1), \cdots, rank(W_n))$

Theorem 2. Rank Preservation of LFA

Define the simple LFA as: LFA(W) = $C_1W_1 + C_2W_2W_1 + \cdots + C_nW_n \cdots W_1 = \sum_{l=1}^{n} C_l(\Pi_{k=1}^{l} W_k)$
Then, the rank of the default model, denoted by $\widetilde{W}$, $rank(\widetilde{W}) = \min(rank(W_1), \cdots, rank(W_n))$ is lower than the rank of LFA model, $\text{rank}(LFA(W)) \geq rank(\widetilde{W})$.

⬇

- Prevent overfitting caused by rank diminishing and enhance domain generalization performance

Result

Few-shot Image Classification

Method	Setting	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	Average
Zero-Shot CLIP Radford et al. [2021]	0-shot	66.73	93.31	89.13	65.64	70.13	85.86	24.72	62.56	44.03	48.44	67.67	65.29
LinearProbing CLIP Radford et al. [2021]	16-shot	72.24	**96.43**	**92.34**	81.46	97.85	**86.22**	41.19	**75.99**	**71.34**	84.10	**85.33**	80.41
CLIP + LFA	16-shot	**72.83**	95.74	91.28	**83.55**	**97.93**	84.84	**45.96**	74.66	70.63	**87.05**	84.22	**80.79**
CoOp Zhou et al. [2022c]	16-shot	71.80	**95.50**	**91.70**	83.10	96.70	**84.20**	43.20	**74.50**	69.60	84.40	82.40	79.74
CoOp + LFA	16-shot	**72.22**	95.38	90.30	**85.00**	**97.40**	83.18	**49.20**	72.96	**71.57**	**88.10**	**83.00**	**80.76**
CLIP-Adapter Gao et al. [2021a]	16-shot	70.20	94.50	**91.80**	69.20	77.90	**86.70**	29.10	70.40	49.20	62.30	74.80	70.55
CLIP-Adapter + LFA	16-shot	**72.87**	**95.42**	91.20	**83.25**	**98.21**	84.91	**46.32**	**74.54**	**71.28**	**86.63**	**83.88**	**80.77**
MaPLe Khattak et al. [2023]	16-shot	70.62	95.46	**93.68**	73.88	93.79	**87.37**	37.56	**74.66**	66.37	87.17	80.31	78.26
MaPLe + LFA	16-shot	**72.66**	**96.19**	91.58	**85.47**	**97.72**	85.05	**49.44**	74.38	**73.94**	**90.65**	**85.49**	**82.05**

Domain Generalization

Model	Accuracy (%) VLCS	PACS	OfficeHome	Terra	DomainNet	Avg.
ViT-B / 16 [11] with pre-trained weights from CLIP [44]						
ZS-CLIP(C) † [Radford et al., 2021]	76.4	95.7	79.9	33.9	57.8	68.7
ZS-CLIP(PC) † [Radford et al., 2021]	82.4	**96.1**	82.3	31.3	57.7	70.0
MIRO † [Cha et al., 2022]	82.2	95.6	82.5	**54.3**	54.0	73.7
ZS-CLIP(PC) + DPL [Zhang et al., 2021]	81.5	95.8	82.6	44.2	**59.3**	72.7
ZS-CLIP(PC) + QLoRA [Dettmers et al., 2024]	82.5	**96.1**	82.5	43.4	59.0	72.7
ZS-CLIP(PC) + LFA	81.0	**96.1**	**82.7**	50.3	59.2	73.9
ZS-CLIP(PC) + QLoRA + LFA	81.9	**96.1**	81.6	52.0	58.7	**74.1**

Learning Efficiency

Model		LinearProbing CLIP	CLIP + DPL	CLIP + QLoRA	CLIP + LFA
Accuracy Average (%)		72.0	72.7	72.7	73.9
Peak Memory (MB) (batch-size 32)	VLCS	1707	12763	8716	1757
	PACS	1707	13077	8717	1771
	OfficeHome	1707	19200	9840	1870
	TerraIncognita	1707	13301	8717	1770
	DomainNet	1745	46074*	21579	3089
Training Time (300 step)	VLCS	11m 40s	11m 46s	2m 4s	11m 32s
	PACS	2m 32s	10m 10s	1m 46s	2m 28s
	OfficeHome	5m 44s	10m 31s	1m 49s	5m 18s
	TerraIncognita	2m 50s	10m 31s	1m 51s	3m 1s
	DomainNet	4m 10s	10m 54s*	3m 47s	4m 7s
Inference Time	VLCS	49s	1m 9s	12s	49s
	PACS	21s	46s	7s	22s
	OfficeHome	1m 5s	1m 14s	16s	54s
	TerraIncognita	1m 55s	2m 9s	25s	2m 3s
	DomainNet	24m 22s	1h 4m 46s*	12m 4s	22m 33s

Qualitative Result

		normalized final logits $= z^{(v)}_{I_{\{image\}}} \cdot z^{(t)}_{T_{\{class\}}}$							predict
		T_{dog}	$T_{elephant}$	$T_{giraffe}$	T_{guitar}	T_{horse}	T_{house}	T_{person}	
Linear Probing CLIP	last layer	0	0.12	0.38	0	0.39	0.03	0.08	horse
CLIP + LFA	last layer	0.02	0.1	0.78	0.08	0.01	0	0	giraffe
	aggregated	0.02	0.78	0.12	0.01	0.05	0.02	0.01	elephant

	Normalized Attention Score (= Activation Probability per Layer)											
layer	1	2	3	4	5	6	7	8	9	10	11	12
$\sigma(e^{(t)}_{T_{elephant}, cross})$	0.05	0.04	0.04	0.04	0.04	0.05	0.06	0.06	0.06	0.06	0.11	0.39
$\sigma(e^{(t)}_{T_{elephant}, self})$	0.05	0.05	0.05	0.05	0.04	0.04	0.06	0.06	0.07	0.08	0.12	0.35
$\sigma(e^{(v)}_{I_{elephant}})$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.03	0.04	0.81

Conclusion

- This work suggests that **combining and utilizing existing lower-layer features**, without the need for extracting new ones, can **enhance model generalization performance at minimal cost**. It has demonstrated improved performance in both **Domain Generalization** and **Few-shot Image Classification**.
- It requires **less memory than QLoRA**, an efficient fine-tuning approach, and **outperforms MIRO**, a model known for its strong performance in domain generalization. This method achieves **better performance at a cost similar to that of a linear probing model**, which trains only the projection weights of the final layer.
- LFA's **compatibility with CLIP-based models** and its **efficiency** highlight its practical value.