TUE-PM-251

Visual-Language Prompt Tuning with Knowledge-guided Context Optimization

Hantao Yao[1], Rui Zhang[2], Changsheng Xu [1,3]

1State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, CAS

2State Key Lab of Processors, Institute of Computing Technology, CAS;

3 University of Chinese Academy of Sciences(CAS),

{hantao.yao,csxu}@nlpr.ia.ac.cn;zhangrui@ict.ac.cn

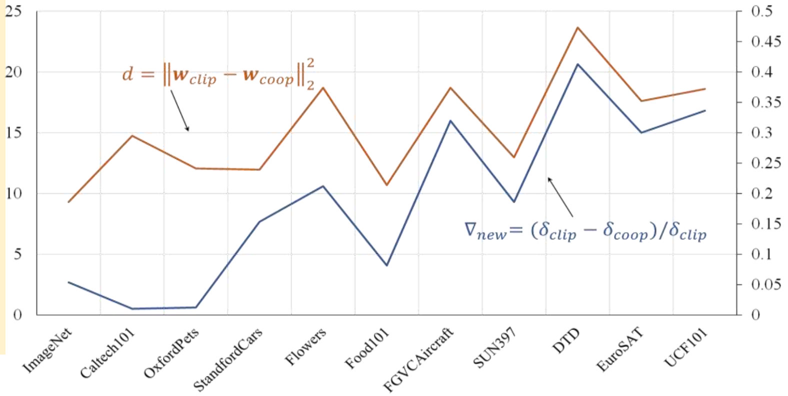
Summary

- **Prompt Tuning** has been proposed to adapt the pretrained VLM to downstream tasks, achieving a fantastic performance on various few-shot or zero-shot visual recognization task.

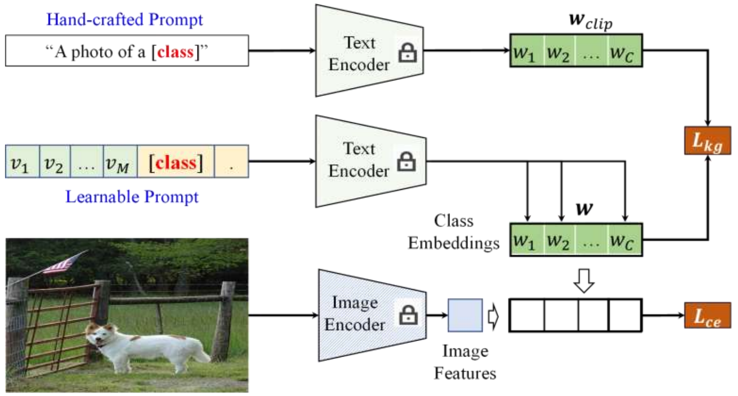
- **Motivation**: Existing Context Optimization (CoOp) prompt tuning methods **have a worse generalization to the unseen classes.**

Methods	Prompts	Accuracy Base	Accuracy New	Accuracy H	Training-time
CLIP	hand-crafted	69.34	74.22	71.70	-
CoOp	textual	**82.63**	67.99	74.60	6ms/image
ProGrad	textual	82.48	70.75	76.16	22ms/image
CoCoOp	textual+visual	80.47	71.69	75.83	160ms/image
KgCoOp	textual	80.73	**73.6**	**77.0**	**6ms**/image

- **Main insight**: The degree of performance degradation on the New class is consist with the distance between the learnable textual embedding and the hand-crafted textual embedding.



- **Method**: an regularizer L_{kg} is proposed to minimize the discrepancy between the hand-craft textual embedding $\mathbf{w}_{clip}$ and the learnable textual embeddings $\mathbf{w}$.



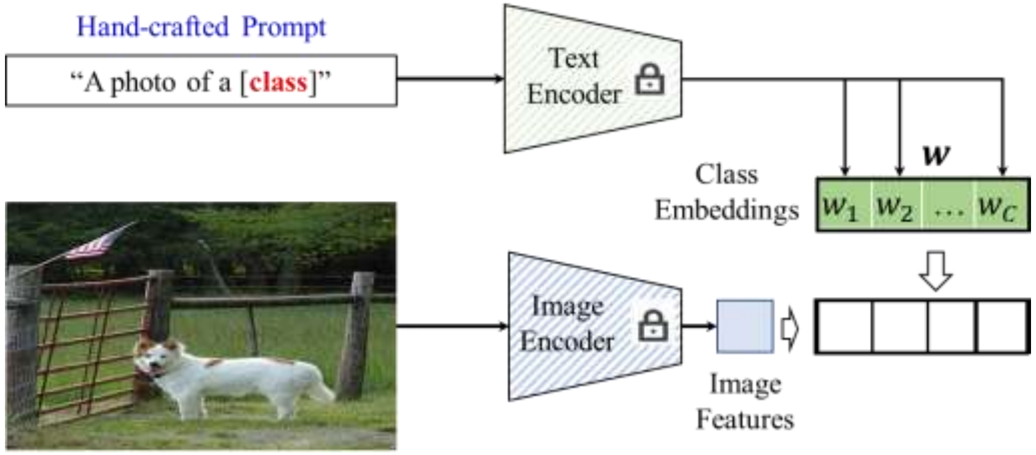
- **Reasonable of minimizing L_{kg}:** lower distance, higher performance.

λ	0.0	1.0	2.0	4.0	6.0	8.0	10.0
L_{kg}	0.18	0.038	0.024	0.015	0.010	0.006	0.005
H	75.38	76.18	76.31	76.86	76.82	77	76.79

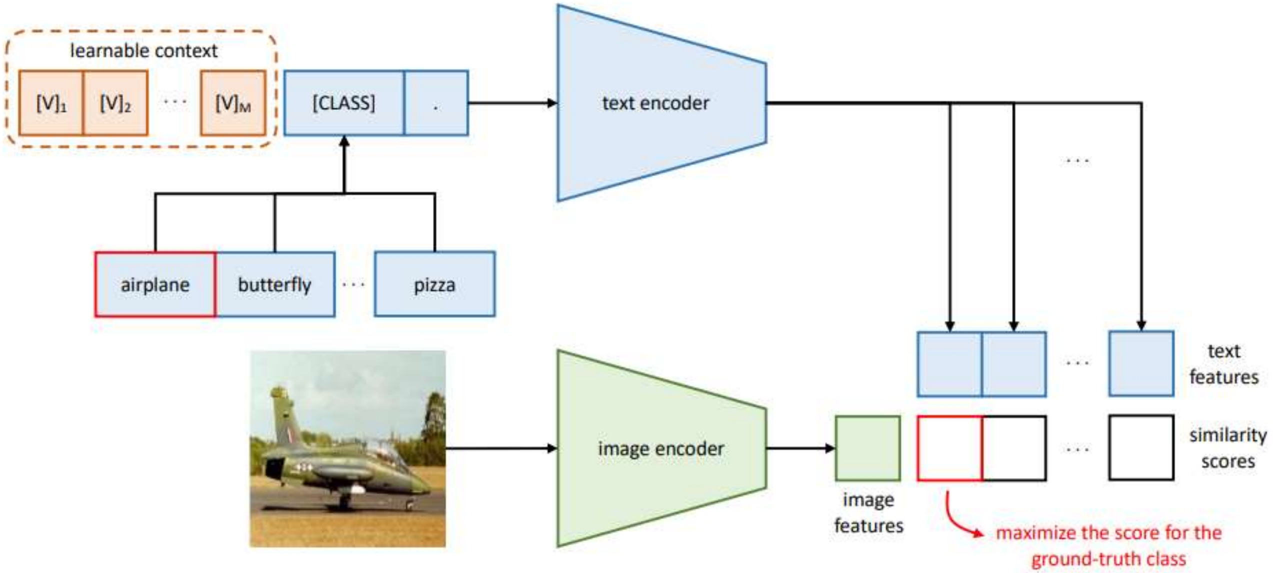
Prompt Tuning

- **Prompt Tuning** has been proposed to adapt the pretrained VLM to downstream tasks, achieving a fantastic performance on various few-shot or zero-shot visual recognization task.

- CLIP uses a **hand-crafted prompts** to model the textual-based class embedding for zero-shot prediction.



- Context Optimization(CoOp) aims to model a prompt's context using **a set of learnable vectors**.

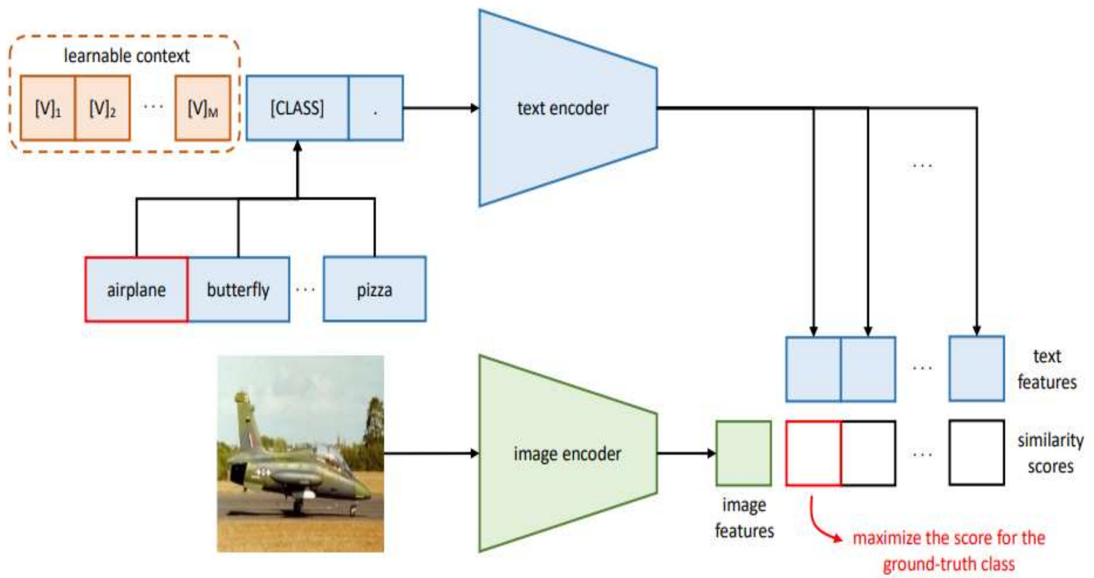


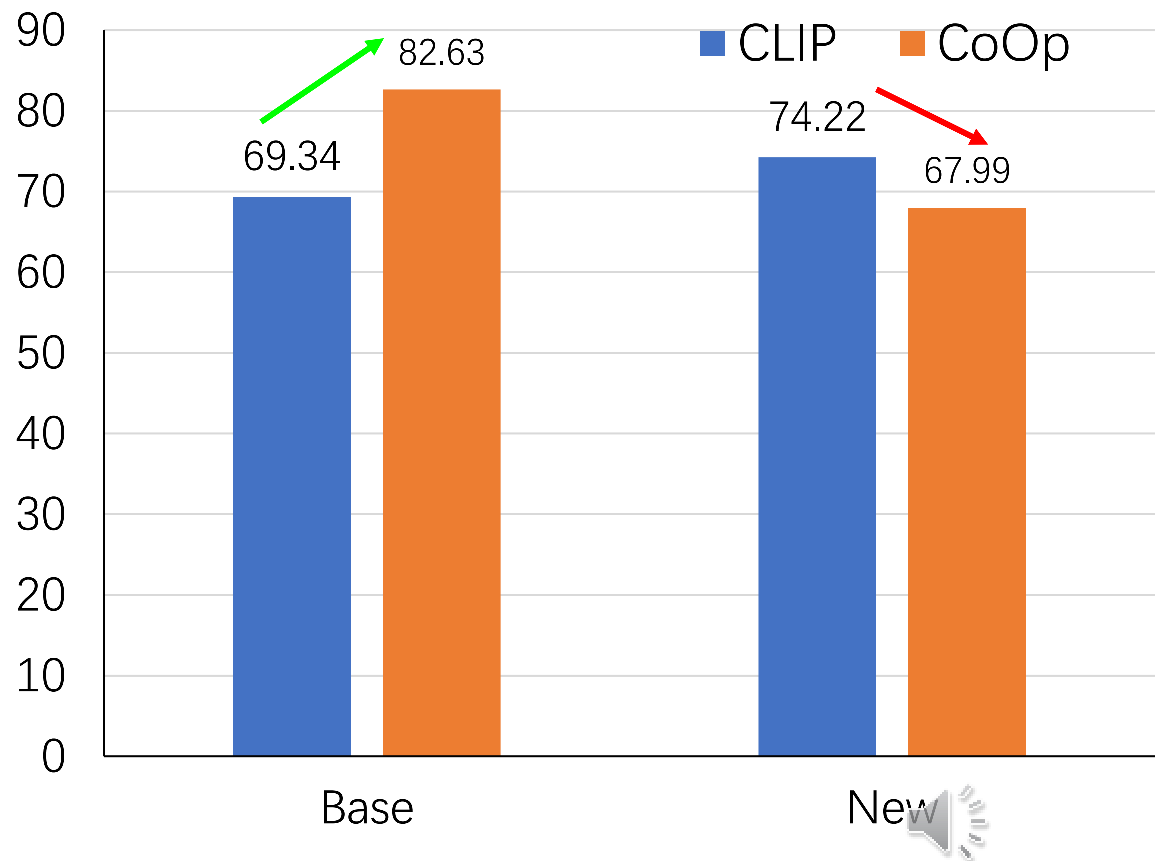
Overview of Context Optimization(CoOp)[1]

1 Image comes from "Learning to Prompt for Vision-Language Models"

Context Optimization(CoOp)

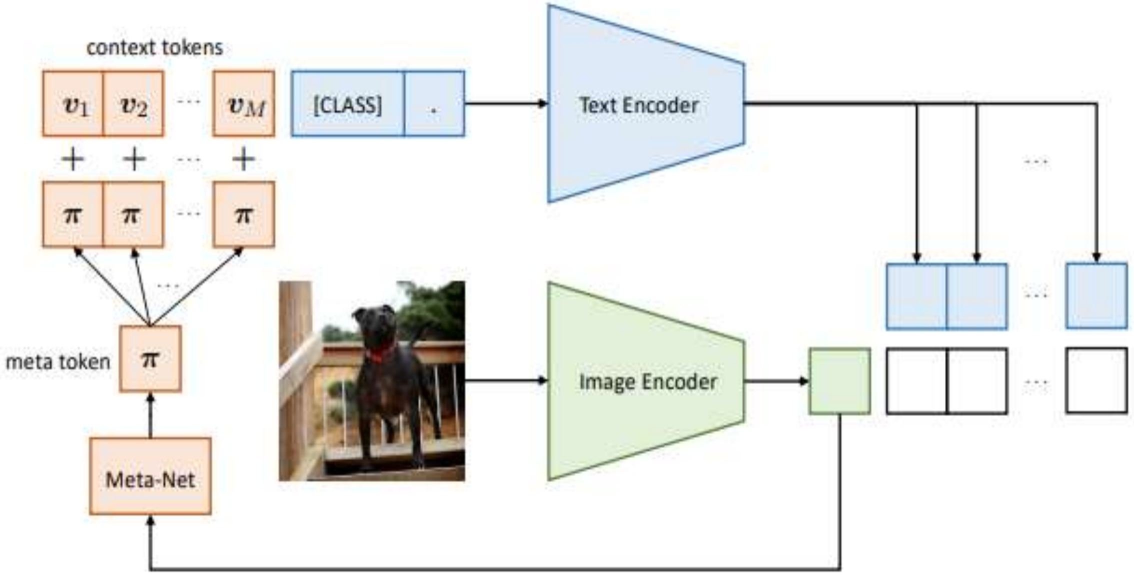
- Context Optimization(CoOp) aims to model a prompt's context using a set of learnable vectors.
- CoOp is overfitted on the trained seen domain(Base), **leading a worse generalization on the unseen domain(New)**.



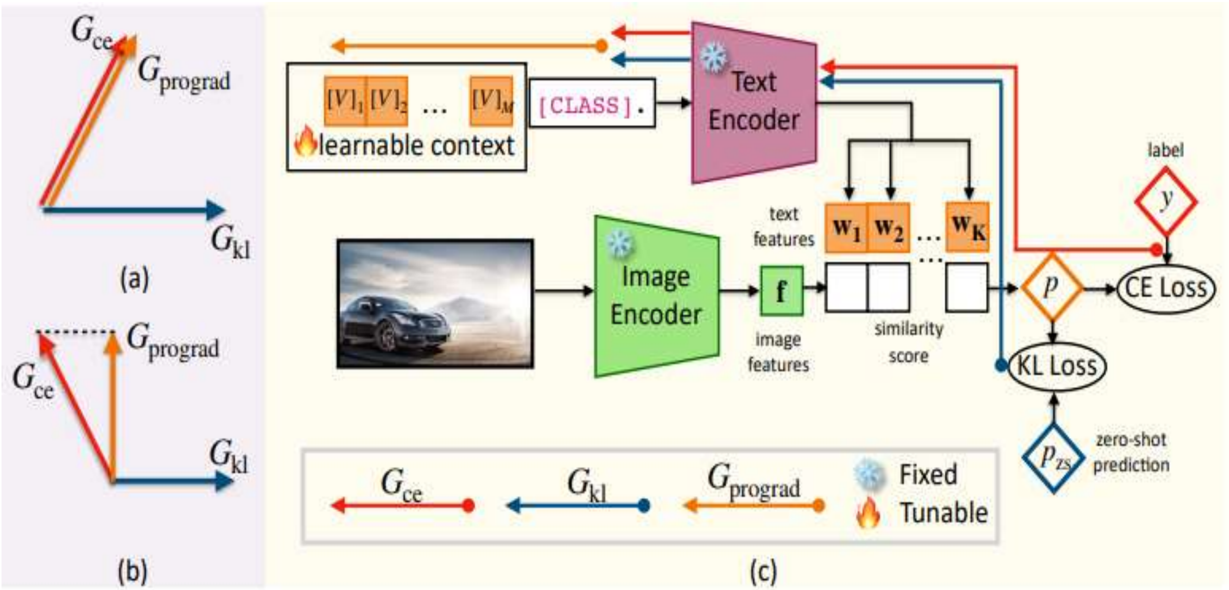


CoOp-based Methods

- CoCoOp and ProGrad are proposed to boost the generalization on the unseen domain.
- CoCoOp combines a set of context vectors and **the generated image-conditional token**
- ProGrad aims to regularize each tuning step **not to conflict with the general knowledge already offered by the original prompt.**



Conditional Context Optimization(CoCoOp)



Prompt-aligned Gradient(ProGrad)

CoOp-based Methods

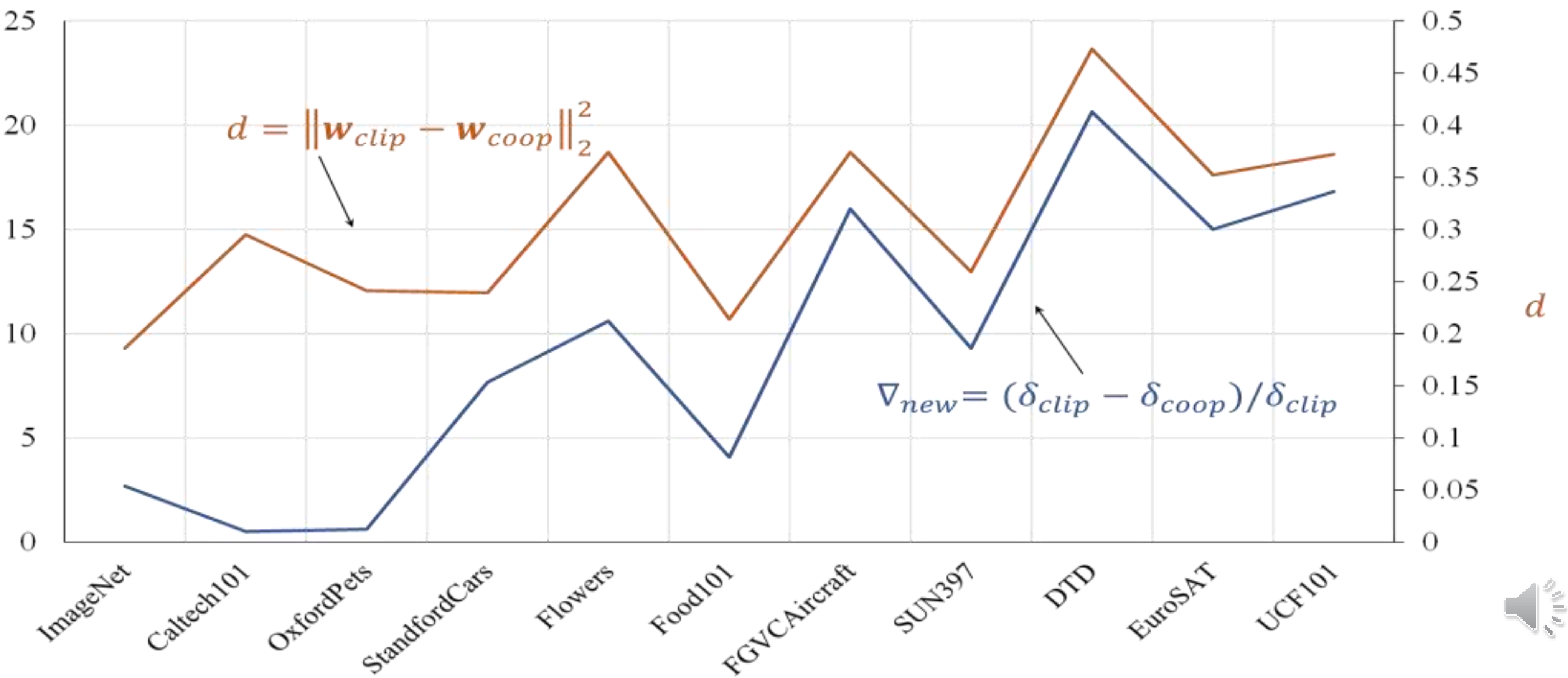
- CoOp, CoCoOp and ProGrad still have the poor the generalization on the unseen domain.
  - The New performance has an obvious gap with the 74.22% obtained by CLIP.

Methods	Prompts	Accuracy			Training-time
		Base	New	H	
CLIP	Hand-crafted	69.34	**74.22**	71.70	-
CoOp	Textual	82.63	**67.99**	74.60	6ms/image
ProGrad	Textual	82.48	**70.75**	76.16	22ms/image
CoCoOp	Textual+visual	80.47	**71.69**	75.83	160ms/image

CoOp-based methods focus on inferring the discriminative learnable prompt on the seen domain, **while ignoring the high generalization knowledge contained in the pretrained CLIP model(Catastrophic Knowledge Forgetting).**

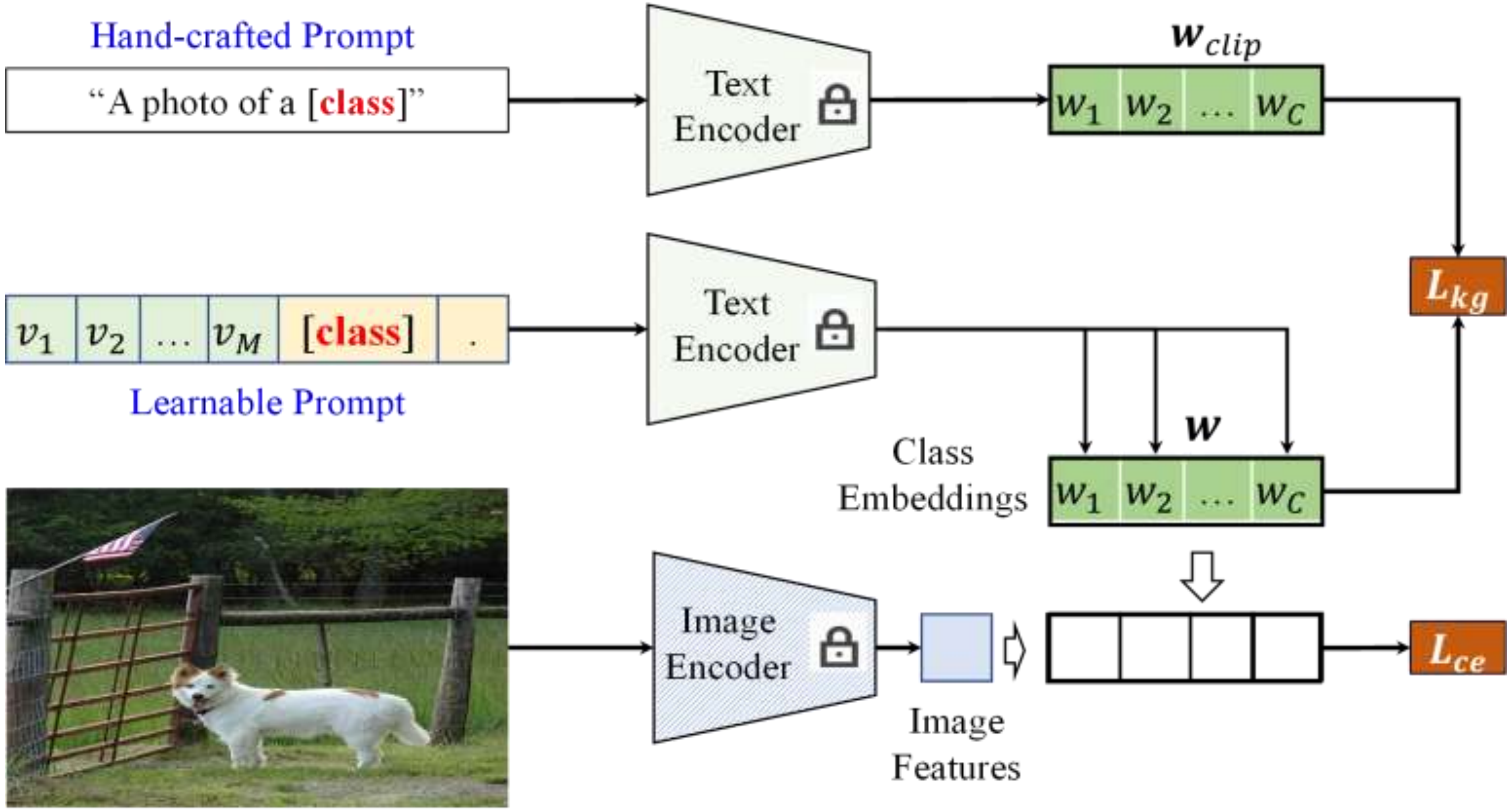
Main Insight

- The degree of performance degradation on the New class is consist with the distance between the learnable textual embedding and the hand-crafted textual embedding.



Knowledge-guided Context Optimization

- Based on the standard CoOp method, an additional regularizer L_{kg} is proposed to minimize the discrepancy between the hand-craft textual embedding $\mathbf{w}_{clip}$ and the learnable textual embeddings $\mathbf{w}$.



$$L_{kg} = \frac{1}{N_c} \sum_{i=1}^{N_c} \left\| \mathbf{w}_i - \mathbf{w}_i^{clip} \right\|_2^2$$

$$L = L_{ce} + \lambda L_{kg}$$

$$\mathcal{L}_{ce} = -\sum_{\mathbf{x} \in \mathbf{X}} \log \frac{\exp(d(\mathbf{x}, \mathbf{w}_y)/\tau)}{\sum_{i=1}^{N_c} \exp(d(\mathbf{x}, \mathbf{w}_i)/\tau)},$$

Experiment

Reasonable of minimizing L_{kg}:

- **lower distance, higher performance**.

λ	0.0	1.0	2.0	4.0	6.0	8.0	10.0
L_{kg}	0.18	0.038	0.024	0.015	0.010	0.006	0.005
H	75.38	76.18	76.31	76.86	76.82	77	76.79

Generalization of L_{kg}:

- **Adding L_{kg} on three type of existing methods boost their performance**.

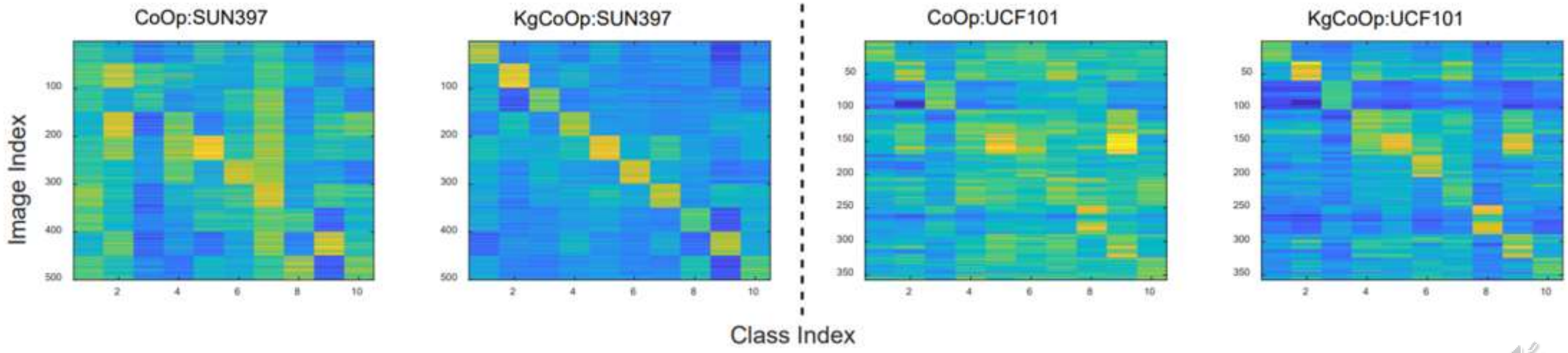
Methods	*Base*	*New*	*H*
CoOP	82.63	67.99	74.6
CoOp+L_{kg}	80.73(↓-1.9)	73.6(↑ 5.61)	77(↑2.4)
CoCoOp	80.43	71.69	75.83
CoCoOp+L_{kg}	77.96(↓-2.50)	74.75(↑3.06)	76.32(↑0.49)
ProGrad	82.48	70.75	71.16
ProGrad+L_{kg}	78.64(↓-3.84)	74.72(↑3.97)	76.63(↑0.47)

Experiment

■ Effectiveness of templates:

Templates	"{}"	"a photo of {}"	"itap of a {}"	"a photo of the large {}"	"a {} in a video game"	"a photo of a {}, a type of {}"
H	76.02	76.85	76.23	76.71	76.12	77.0

■ Visualization:



Experiment

- **Effectiveness of KgCoOp**: *Base-to-new setting*
  - Two Backbones: *ViT-B/16 and ResNet50*
  - Three K-shots: *4/8/16*

Backbones	Methods	K=4			K=8			K=16		
		Base	New	H	Base	New	H	Base	New	H
ViT-B/16	CoOp	78.43	68.03	72.44	80.73	68.39	73.5	82.63	67.99	74.60
	CoCoOp	76.72	**73.34**	74.85	78.56	72.0	74.9	80.47	71.69	75.83
	ProGrad	79.18	71.14	74.62	**80.62**	71.02	75.2	**82.48**	70.75	76.16
	KgCoOp	**79.92**	73.11	**75.90**	78.36	**73.89**	**76.06**	80.73	**73.6**	**77.0**
ResNet-50	CoOp	72.06	59.69	65.29	74.72	58.05	65.34	77.24	57.4	65.86
	CoCoOp	71.39	65.74	68.45	73.4	66.42	69.29	75.2	64.64	68.9
	ProGrad	**73.88**	64.95	69.13	**76.25**	64.74	70.03	**77.98**	64.41	69.94
	KgCoOp	72.42	**68.00**	**70.14**	74.08	**67.86**	**70.84**	75.51	**67.53**	**71.30**

Experiment

- **Effectiveness of KgCoOp:** *Domain generalization with 16-shot*

	Prompts	Source	Target				
		ImageNet	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R	Avg.
CLIP	Hand-crafted	66.73	60.83	46.15	47.77	73.96	57.17
UPT	vp+tp	72.63	64.35	48.66	50.66	76.24	59.98
CoCoOp	vp+tp	71.02	64.07	48.75	50.63	76.18	59.90
CoOp	tp	71.51	64.2	47.99	49.71	75.21	59.28
ProGrad	tp	72.24	64.73	47.61	49.39	74.58	59.07
KgCoOp	tp	71.2	64.1	48.97	50.69	76.7	60.11

Experiment

- **Effectiveness of KgCoOp**: *Few-shot Learning with 4-shots*

Datasets	CoOp	CoCoOp	ProGrad	KgCoOp
ImageNet	69.38	**70.55**	70.21	70.19
Caltech101	94.44	**94.98**	94.93	94.65
OxfordPets	91.3	**93.01**	93.21	93.2
StanfordCars	**72.73**	69.1	71.75	71.98
Flowers102	**91.14**	82.56	89.98	90.69
Food101	82.58	**86.64**	85.77	86.59
FGVCAircraft	33.18	30.87	**32.93**	32.47
SUN397	70.13	70.5	71.17	**71.79**
DTD	**58.57**	54.79	57.72	58.31
EuroSAT	68.62	63.83	70.84	**71.06**
UCF101	77.41	74.99	77.82	**78.40**
Avg.	73.59	71.98	74.21	**74.48**

Conclusion

- We first give a discussion and analysis about the performance's degradation on unseen domains for CoOp-based prompt tuning.
- We demonstrate that minimizing the distance between the learnable textual embedding and general textual embedding can boost the generability on unseen classes.
- A simple and efficient KgCoOp is proposed for visual-language prompt tuning, e.g., achieves better performance with less training time.
- Code: https://github.com/htyao89/KgCoOp

Methods	Prompts	Accuracy			Training-time
		Base	New	H	
CLIP	hand-crafted	69.34	74.22	71.70	-
CoOp	textual	**82.63**	67.99	74.60	6ms/image
ProGrad	textual	82.48	70.75	76.16	22ms/image
CoCoOp	textual+visual	80.47	71.69	75.83	160ms/image
KgCoOp	textual	80.73	**73.6**	**77.0**	**6ms**/image

Visual-Language Prompt Tuning with Knowledge-guided Context Optimization

Hantao Yao[1], Rui Zhang[2], Changsheng Xu [1,3]

1State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, CAS

2State Key Lab of Processors, Institute of Computing Technology, CAS;

3 University of Chinese Academy of Sciences(CAS),

{hantao.yao,csxu}@nlpr.ia.ac.cn;zhangrui@ict.ac.cn