



UNIVERSITÀ  
DI TRENTO

# Learning prompts for transfer learning with test-time adaptation

Trends and Applications of Computer Vision

Giovanni Scialla, Mattia Franzin, Adnan Irshad, Hira Afzal

# Table of Contents


- Learning prompt for Visual-language models (CoOp)
- Conditional Prompt Learning for Vision-Language Models (CoCoOp)
- Multi-modal Prompt Learning (MaPLe)
- Prompt Pre-Training with Twenty-Thousand Classes for Open-Vocabulary Visual Recognition (POMP)
- Visual-Language Prompt Tuning with Knowledge-guided Context Optimization (KgCoOp)
- Prompt-aligned Gradient for Prompt Tuning (ProGrad)
- Test-time prompt tuning for zero-shot generalization in vision-language models (TPT)
- Diverse Data Augmentation with Diffusions for Effective Test-time Prompt Tuning (DiffTPT)
- Align Your Prompts Test-Time Prompting with Distribution Alignment for Zero-Shot Generalization (PromptAlign)


# Learning to Prompt for Visual-Language Models [1]

**prompting:** transfer visual concepts learned by visual-language models to any downstream task

**prompt engineering:** tune a prompt by adding some context that is meaningful to a task, it requires prior knowledge about the task and ideally the language model's underlying mechanism.

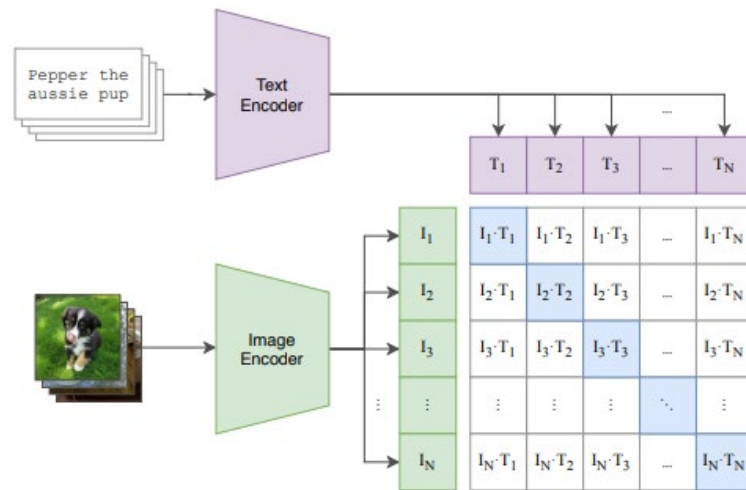
**Context Optimization (CoOp) :** automate prompt engineering by modeling a prompt's context words with learnable vectors

Caltech101	Prompt	Accuracy
	a [CLASS].	82.68
	a photo of [CLASS].	80.81
	a photo of a [CLASS].	86.29
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>91.83</b>

Flowers102	Prompt	Accuracy
	a photo of a [CLASS].	60.86
	a flower photo of a [CLASS].	65.81
	a photo of a [CLASS], a type of flower.	66.14
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	<b>94.51</b>

# Ingredients: Vision-language pre-training model (CLIP) [2]

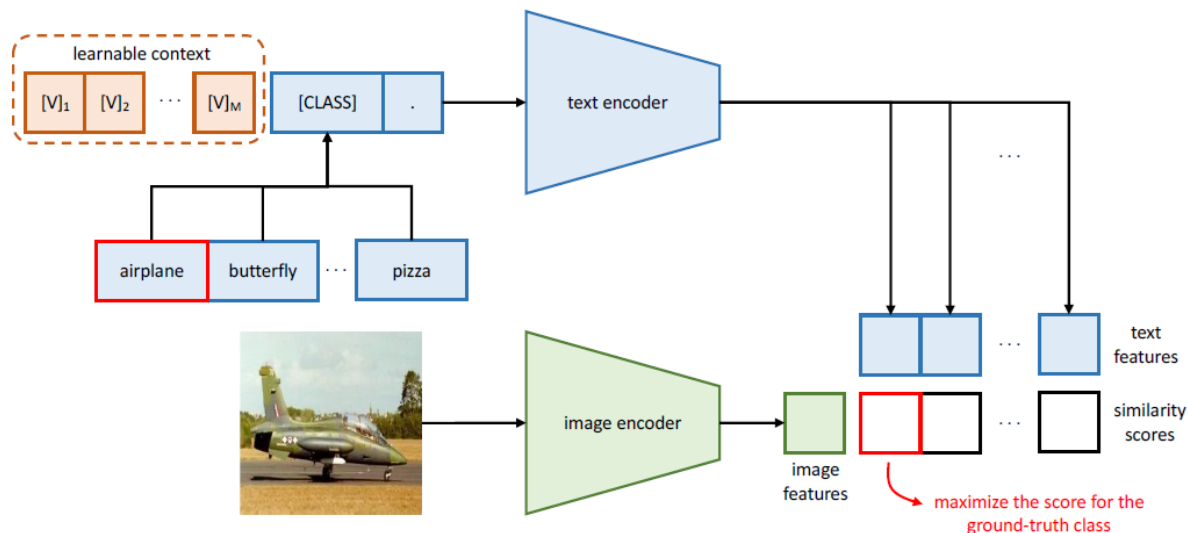
- **image encoder:** map high-dimensional images into a low-dimensional embedding space
- **text encoder:** generate text representations from natural language



**Training:** align the two embedding spaces learned for images and text

- **Contrastive loss:** maximize the cosine similarity for matched pairs while minimizing the cosine similarity for all other unmatched pairs

# Context Optimization (CoOp) - Architecture



prompt tuning by modeling context words with continuous vectors that are learned from data while the vision-language model pre-trained parameters are frozen

# Context Optimization (CoOp) - Method

- **Unified Context** : shares the same context with all classes.  
(**M** = number of context tokens).

$$prompt \rightarrow t = [V]_1[V]_2 \dots [V]_M [CLASS]$$

The class token can be also placed in the middle (increase flexibility)

$$prompt \rightarrow t = [V]_1 \dots [V]_{\frac{M}{2}} [CLASS] [V]_{\frac{M}{2}+1} \dots [V]_M$$

- **Class-Specific Context (csc)**: context vectors are independent to each class

$$[V]_1^i [V]_2^i \dots [V]_M^i \neq [V]_1^j [V]_2^j \dots [V]_M^j \quad \text{for } i \neq j \text{ and } i, j \in \{1, \dots, K\}.$$

# Context Optimization (CoOp) - Training and Experiments

- forward a prompt  $t$  to the text encoder  $g(\cdot)$  to obtain a classification weight vector representing a visual concept, the prediction probability is:

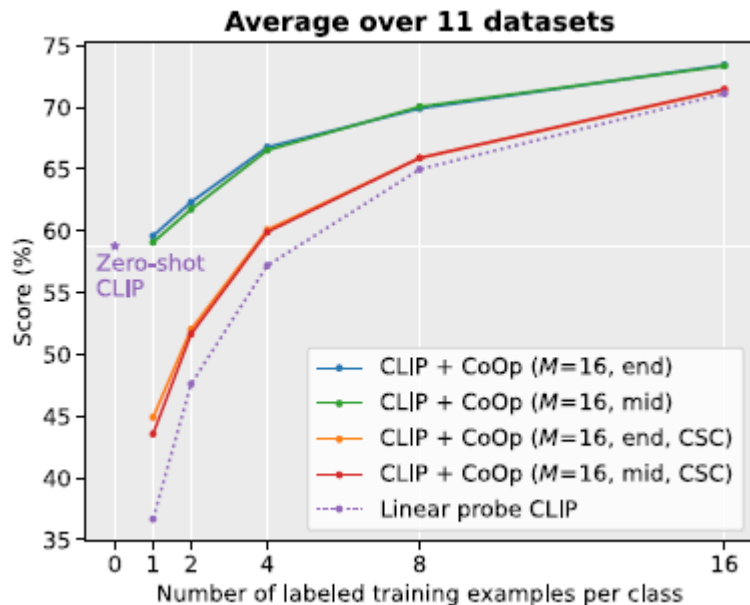
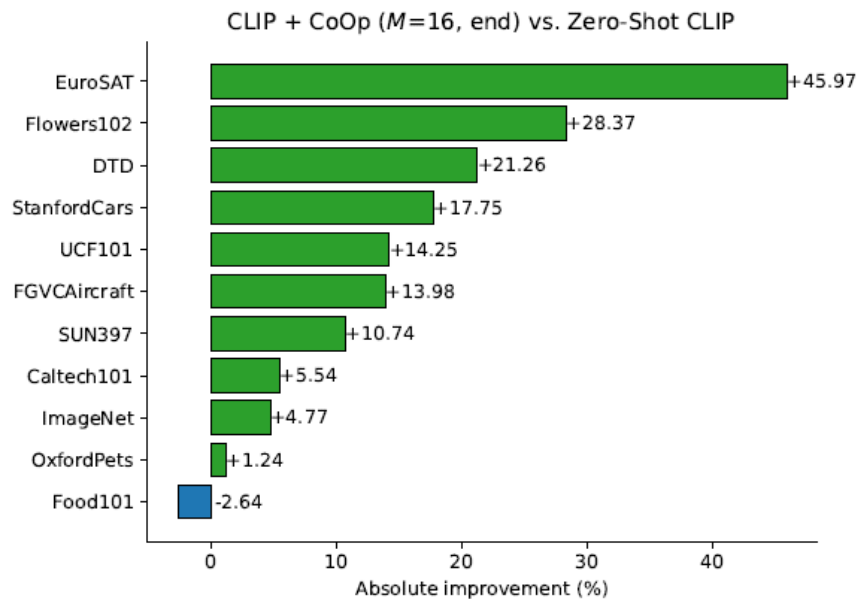
$$p(y = i|x) = \frac{\exp(\cos(g(t_i), f)/\tau)}{\sum_{j=1}^K \exp(\cos(g(t_j), f)/\tau)},$$

- optimize the context by using the knowledge encoded in the parameters.

➡ **full exploration in the word embedding space.**

# Context Optimization (CoOp) - Training and Experiments

Datasets: 11 publicly available image classification datasets used in CLIP





# Conditional Context Optimization (CoCoOp) [3]

**Problem with CoOp:** The context is fixed and optimized only for a specific set of classes (training classes)

**Solution**  **Conditional Prompt Learning**

- Make a prompt conditioned on each instance image rather than fixed once learned (**More generalizable**)
- Conditional prompts are optimized to characterize each instance, rather than to serve only for some specific classes. (**More robust to class shift**)

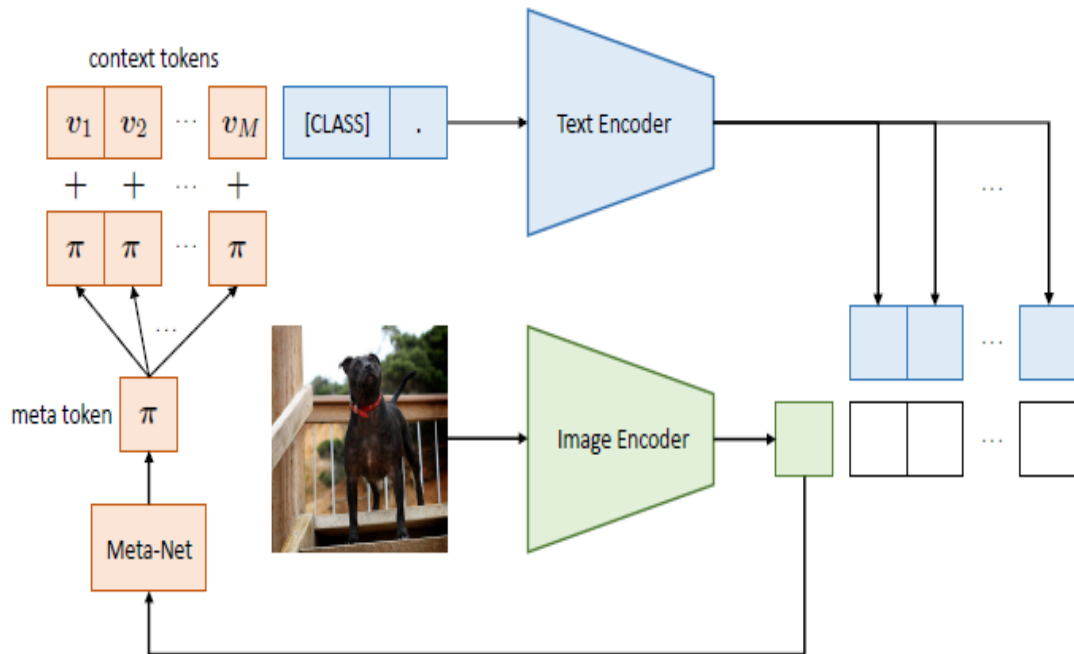
# Conditional Context Optimization (CoCoOp) : Architecture

## Main idea

- Extend CoOp by generate an input-conditional token vector, which is combined with the context vectors

## Learnable components

- Set of context vectors
- Light-weight neural network (Meta-Net) to generate input conditional tokens



# Conditional Context Optimization (CoCoOp) : Method

- Obtain the context token ( $h_\theta(x)$  denote the Meta-Net parametrized by  $\theta$ )

$$v_m(x) = v_m + h_\theta(x) \quad m \in \{1, 2, \dots, M\}$$

- Combine each input conditional token with the context vector

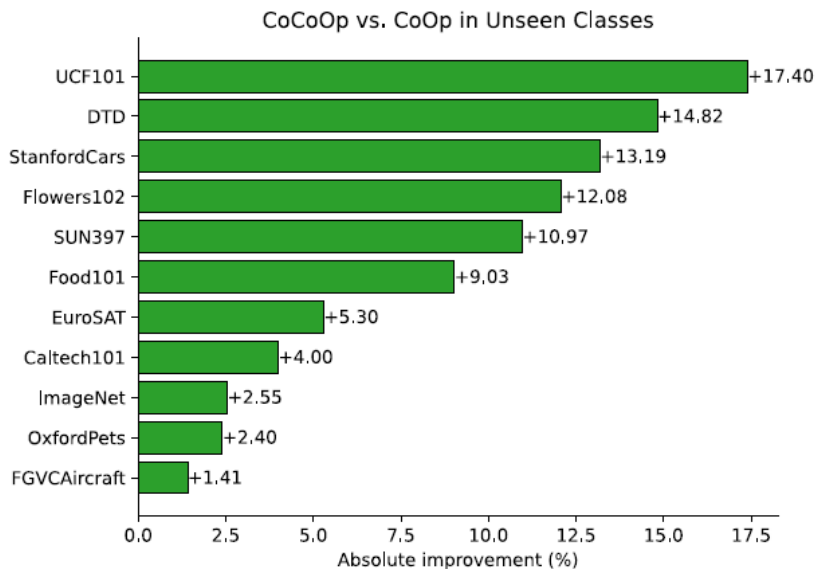
$$t_i(x) = \{v_1(x), v_2(x), \dots, v_M(x), c_i\}$$

- During training, update the context vectors together with the Meta-Net's parameters  $\theta$
- The prediction probability is computed as:

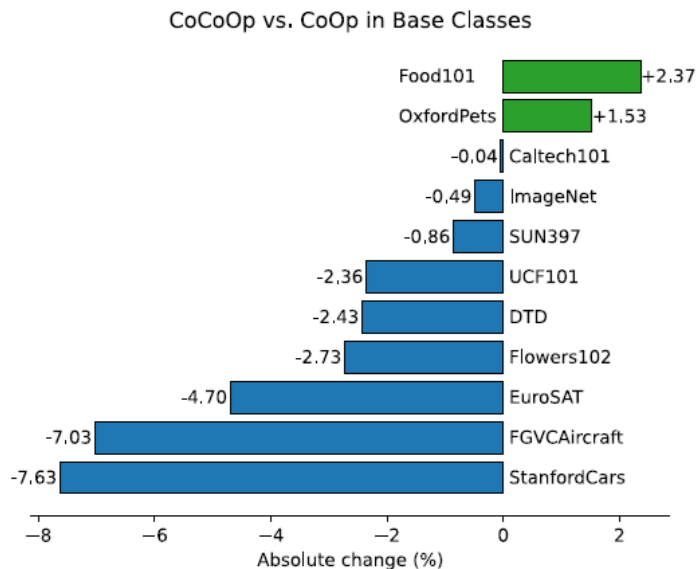
$$p(y|x) = \frac{\exp(\text{sim}(x, g(t_y(x))))/\tau}{\sum_{i=1}^K \exp(\text{sim}(x, g(t_i(x))))/\tau}.$$

# Conditional Context Optimization (CoCoOp) : Results

**Focus:** Comparison of CLIP, CoOp and CoCoOp in the base-to-new generalization setting.



(a)



(b)

# Static prompts (CoOp) vs Dynamic prompts (CoCoOp)

## Pro:

- **CoCoOp Significantly Narrows Generalization Gap:** Dynamic conditional prompts are more generalizable in unseen classes
- **CoCoOp is more compelling than CLIP:** better potential in capturing relevant elements for a recognition task

## Limitations:

- CoCoOp shows performance drops in the base classes
- Significant amount of GPU memory consumptions

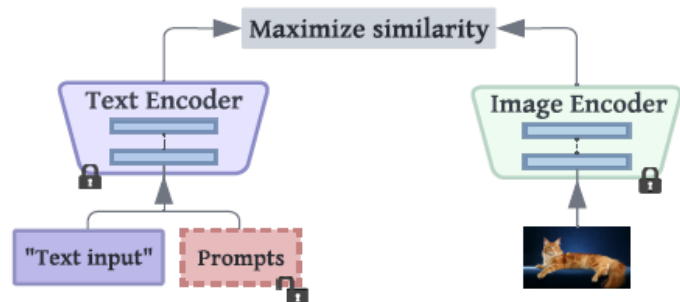
# MaPLe : Multi-Modal Prompt Learning [4]

**Key Idea:** Learn context prompts in both vision and language branches

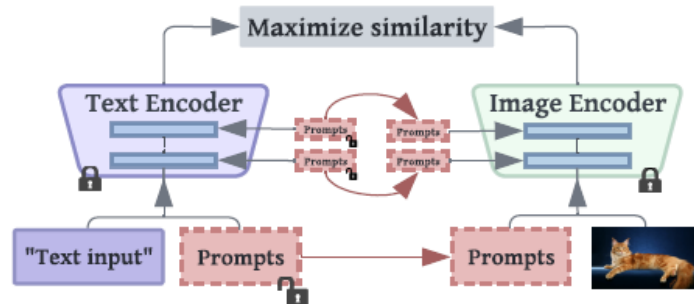
➡ **Align vision-language representations**

**How:**

- *Coupling functions* to link prompts learned in text and image encoders
- *Progressively* learn multi-modal prompts across multiple transformer blocks in both vision and language branches



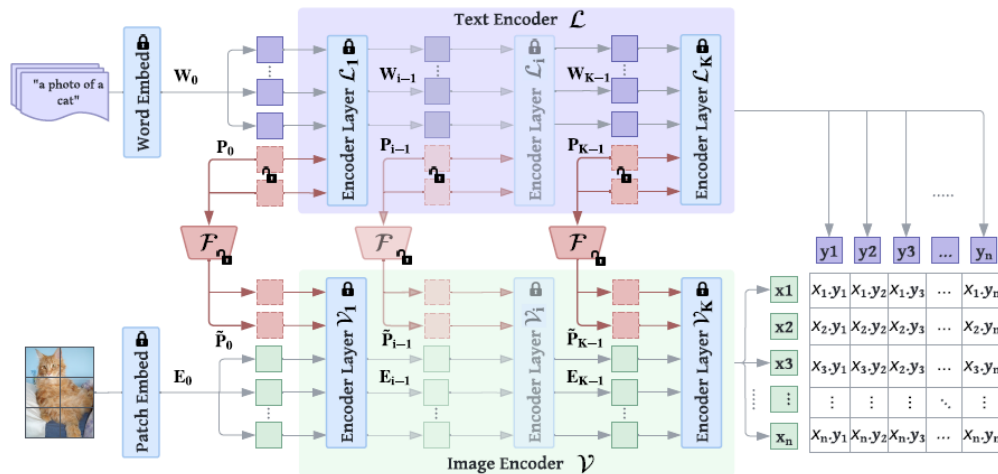
(a) Existing prompt tuning methods (Uni-modal)



(b) Multi-modal Prompt Learning (MaPLe)

# MaPLe : Architecture

- MaPLe tunes both Vision and Language branches
- only the Context prompts are learned



Introduce learnable tokens in the first J layers of both vision and language branches

➡ Utilize the knowledge embedded in CLIP to effectively learn contextual representations

# MaPLe : Method

## Deep Language Prompting

- introduce  $b$  learnable tokens in the language branch of CLIP

$[P^1, P^2, \dots, P^b, W_0]$ , where  $W_0 = [w^1, w^2, \dots, w^N]$

- Introduce new learnable tokens in each transformer block up to a specific depth  $J$
- compute final text representation  $z$

$$[P_j, W_j] = \mathcal{L}_j([P_{j-1}, W_{j-1}]) \quad j = J + 1, \dots, K,$$
$$z = \text{TextProj}(w_K^N).$$

## Deep Vision Prompting

- introduce  $b$  learnable tokens in the vision branch of CLIP
- introduce new learnable tokens in deeper transformer layers of the image encoder up to depth  $J$

$$[c_i, E_i, \_ ] = \mathcal{V}_i([c_{i-1}, E_{i-1}, \tilde{P}_{i-1}]) \quad i = 1, 2, \dots, J,$$

$$[c_j, E_j, \tilde{P}_j] = \mathcal{V}_j([c_{j-1}, E_{j-1}, \tilde{P}_{j-1}]) \quad j = J + 1, \dots, K,$$
$$x = \text{ImageProj}(c_K).$$

## Vision Language Coupling

- MaPLe conditions the vision prompts on language prompts via a V-L coupling function



**Induce mutual synergy**

- Coupling Function: Linear layer which maps  $d_l$  dimensional input to  $d_v$ .

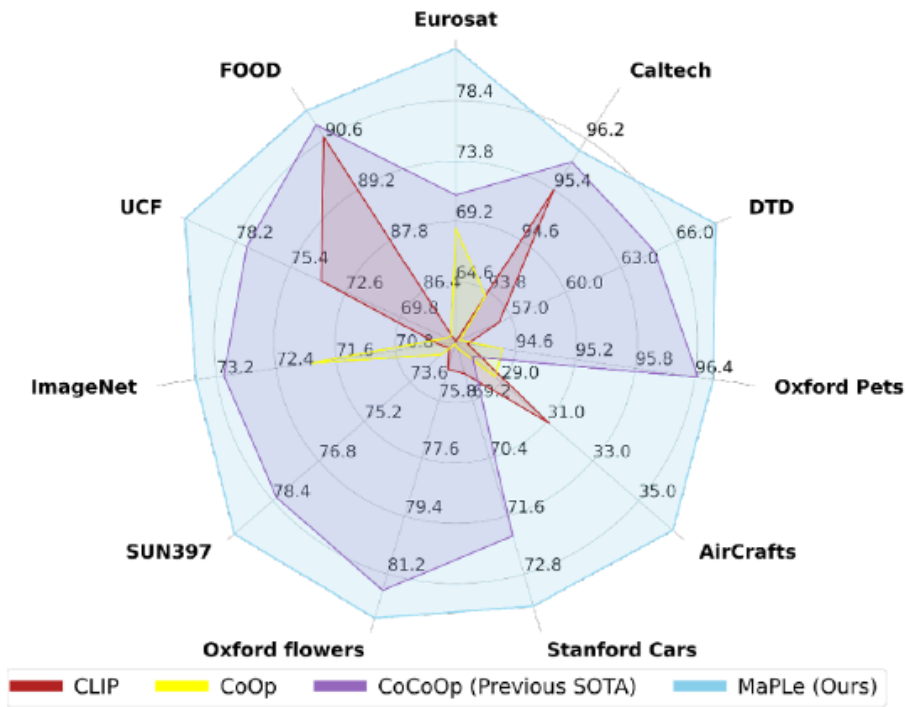


# MaPLE : Results (novel class generalization task)

MaPLE surpasses state-of-the-art methods on 11 diverse image recognition datasets

	Base	Novel	HM
CoOp	<b>82.69</b>	63.22	71.66
Co-CoOp	80.47	71.69	75.83
CoOp <sup>†</sup>	80.85	70.02	75.04
MaPLE	82.28	<b>75.14</b>	<b>78.55</b>

Table 2. Generalization comparison of MaPLE with CoOp<sup>†</sup>.



# POMP : Prompt Pre-Training [...] for [...] Visual Recognition

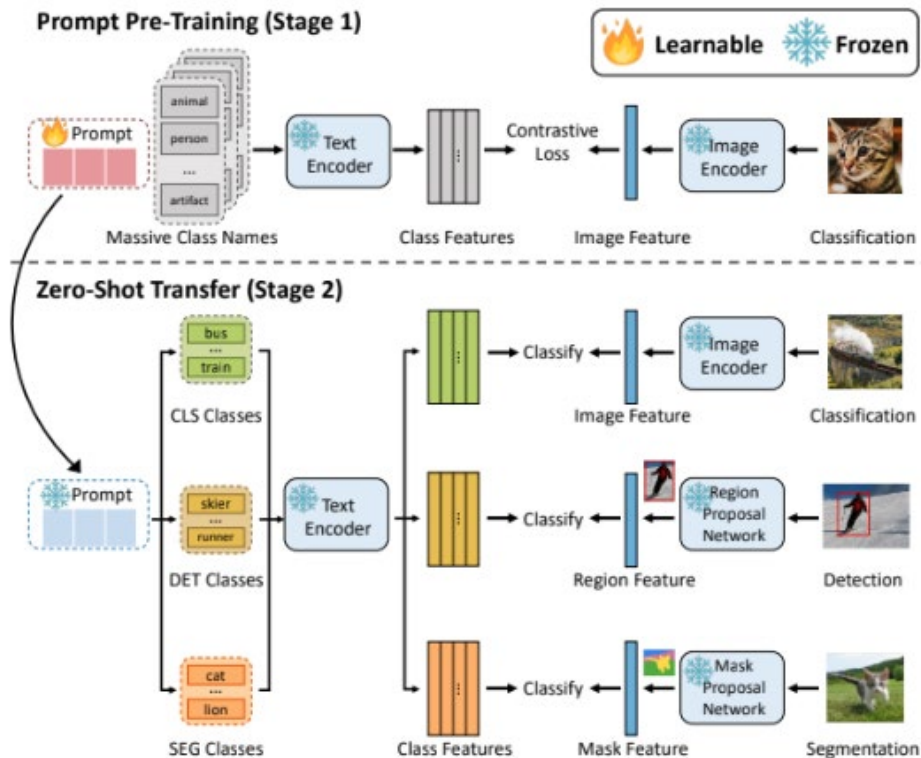
- **CoOp** and **CoCoOp** works good for **few-class** datasets
- When the number of classes raises up, these models require:
  - more **GPU** memory
  - more **training** time
- **POMP** instead of comparing the current class with all the others, samples just  $K$  of them (with an equal probability of  $\frac{1}{N-1}$ ), with  **$K \ll N$**

- For example, with  **$N=1000$** :

Method	Acc. (%)	GPU Mem. (GB)	Training Time (h)
CoOp	71.9	28.2	5.9
CoCoOp	70.1	28.3	27.5
POMP ( $K = 128$ )	71.2	5.3	2.7
POMP ( $K = 256$ )	71.4	8.8	3.3
POMP ( $K = 512$ )	71.6	15.9	4.2

# POMP - Architecture

- Similar setup
- Suitable for downstream task
- Two major components:
  - **local contrast**
  - **local correction**
- **Local contrast** picks up **K** classes
- **Local correction** compensates the smaller gradient
  - caused by less negative classes
  - resulting in a more stringent **decision boundary**



$$m = -\log((K-1)/(N-1)).$$

$$\tilde{P}(y | \mathbf{x}; \Theta) = \frac{\exp(\mathbf{x}^\top \mathbf{w}_y^{(\Theta)} / \tau)}{\exp(\mathbf{x}^\top \mathbf{w}_y^{(\Theta)} / \tau) + \sum_{i \sim \mathcal{N}} \exp(\mathbf{x}^\top \mathbf{w}_i^{(\Theta)} / \tau + m)}.$$

# POMP - Results

Method	ResNet50	ViT-B/32	ViT-B/16
ZeroshotCLIP [42]	17.5	19.8	21.8
Prompt Ensemble [42]	18.8	20.9	23.5
CoOp [65]	16.6	18.1	20.8
MaPLe [29]	-	21.6	24.2
Linear Probing [42]	6.5	18.2	20.9
VPT [12]	-	21.8	24.8
POMP (Ours)	<b>20.2</b>	<b>22.2</b>	<b>25.3</b>

- Pre-training on **Image-21K** dataset when applicable (otherwise Image-1K)
- 16 **shots** per class, 16 **prompt length**, 1K **classes** per training step
- Best choice for downstream task like **semantic segmentation** and **object detection**

	Target (cross-dataset)											Target (cross-domain)				
	<i>Caltech101</i>	<i>OxfordPets</i>	<i>StanfordCars</i>	<i>Flowers102</i>	<i>Food101</i>	<i>Aircraft</i>	<i>SUN397</i>	<i>DTD</i>	<i>EuroSAT</i>	<i>UCF101</i>	<i>Average</i>	<i>ImageNetV2</i>	<i>ImageNet-S</i>	<i>ImageNet-A</i>	<i>ImageNet-R</i>	<i>Average</i>
hard prompt	93.3	88.2	65.6	67.4	85.3	23.7	62.6	44.3	42.0	65.1	63.7	60.9	46.1	47.8	74.0	57.2
CoOp [65]	93.7	89.1	64.5	68.7	85.3	18.5	64.2	41.9	46.4	66.6	63.9	64.2	48.0	49.7	75.2	59.3
CoCoOp [66]	94.4	90.1	65.3	71.9	86.1	22.9	67.4	45.7	45.4	68.2	65.7	64.1	48.8	50.6	76.2	59.9
LASP [5]	94.5	89.4	64.8	70.5	86.3	23.0	67.0	45.5	48.3	68.2	65.8	63.8	49.0	50.7	77.1	60.1
VPT [12]	93.7	<b>90.6</b>	65.0	70.9	86.3	24.9	67.5	46.1	45.9	68.7	66.0	<b>64.2</b>	49.2	51.3	77.0	60.4
MaPLe [29]	93.5	90.5	65.6	72.2	86.2	24.7	67.0	<b>46.5</b>	48.1	<b>68.7</b>	66.3	64.1	49.2	50.9	77.0	60.3
POMP (Ours)	<b>95.0</b>	89.5	<b>66.8</b>	<b>72.4</b>	<b>86.3</b>	<b>25.6</b>	<b>67.7</b>	46.2	<b>52.1</b>	68.5	<b>67.0</b>	63.8	<b>49.8</b>	<b>51.6</b>	<b>77.9</b>	<b>60.8</b>

- Evaluation for **image classification**
- Backbone is **ViT/B-16**
- Higher accuracy indicates **better generalization**

# Generalization problem

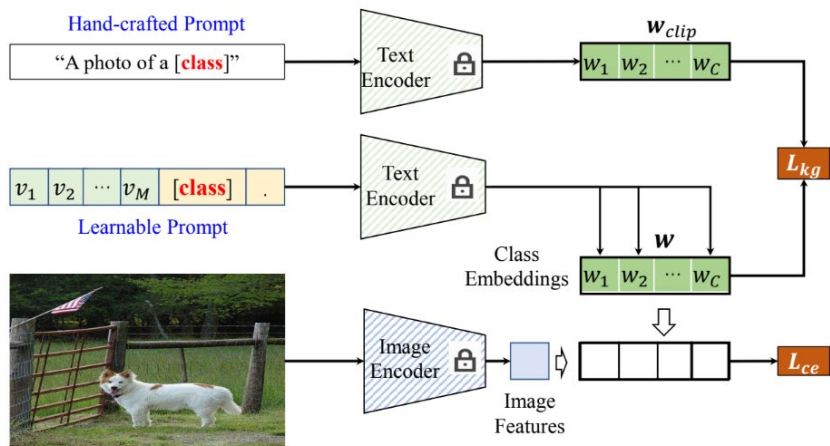
- On **unseen** classes CoOp perform **worse** than CLIP, losing generalization
- CoCoOp attenuates this, but it ends up **slowing** the training by adding the **Meta-Net**
- The prompts becomes little by little specialized for **seen** classes, tending to forget **general knowledge**
- POMP continuously sample **random** classes (usually 1000 per training step) to **improve** generalization

# Solution?

Just keep it simple

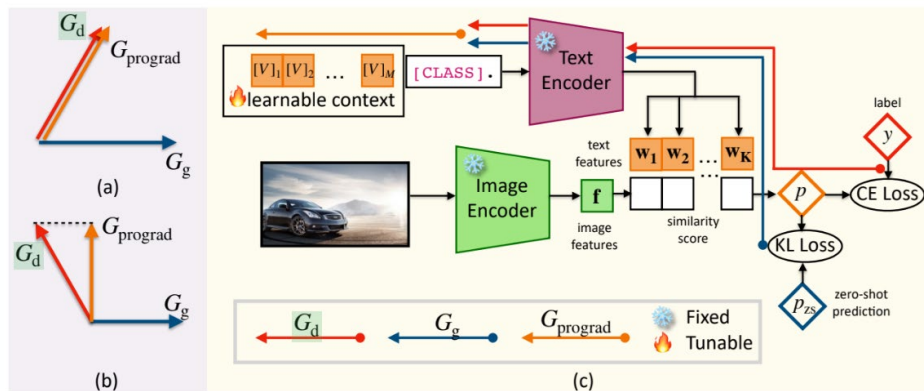
# Just keep it simple - Multiple losses

- Visual-Language Prompt Tuning with Knowledge-guided Context Optimization



- Combine a **learnable prompt** with the **general** one: "a photo of a [class]"

- Prompt-aligned Gradient for Prompt Tuning



- Update the prompt whose **gradient** is aligned (or non-conflicting) to the "**general direction**"

# Visual-Language Prompt Tuning with Knowledge-guided Context Optimization

- The setup is very similar to CoOp
- The **Cross-Entropy** loss is computed as usual, but also another loss is calculated using **textual embeddings**

$$\mathcal{L}_{kg} = \frac{1}{N_c} \sum_{i=1}^{N_c} \|\mathbf{w}_i - \mathbf{w}_i^{clip}\|_2^2,$$

- So now the actual loss is just a **combination** of the two, with  $\lambda$  used for **balancing the effect** of the custom loss

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{kg},$$



# Results - KgCoOp

- Higher performance with **less** training time

Methods	Prompts	Accuracy			Training-time
		Base	New	H	
CLIP	hand-crafted	69.34	74.22	71.70	-
CoOp	textual	<b>82.63</b>	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
<b>KgCoOp</b>	textual	80.73	<b>73.6</b>	<b>77.0</b>	<b>6ms/image</b>

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	<b>73.34</b>	74.85	78.56	72.0	74.9	80.47	71.69	75.83
	ProGrad	79.18	71.14	74.62	<b>80.62</b>	71.02	75.2	<b>82.48</b>	70.75	76.16
	<b>KgCoOp</b>	<b>79.92</b>	73.11	<b>75.90</b>	78.36	<b>73.89</b>	<b>76.06</b>	80.73	<b>73.6</b>	<b>77.0</b>
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	63.64	68.9
	ProGrad	<b>73.88</b>	64.95	69.13	<b>76.25</b>	64.74	70.03	<b>77.98</b>	64.41	69.94
	<b>KgCoOp</b>	72.42	<b>68.00</b>	<b>70.14</b>	74.08	<b>67.86</b>	<b>70.84</b>	75.51	<b>67.53</b>	<b>71.30</b>

# Prompt-aligned Gradient for Prompt Tuning

- Same here, another loss is used, based on **Kullback-Leibler** divergence between CoOp model and Zero-Shot CLIP

$$\mathcal{L}_{\text{kl}}(\mathbf{v}) = - \sum_i p_{\text{zs}}(\mathbf{w}_i | \mathbf{x}) \log \frac{p(\mathbf{t}_i | \mathbf{x})}{p_{\text{zs}}(\mathbf{w}_i | \mathbf{x})}.$$

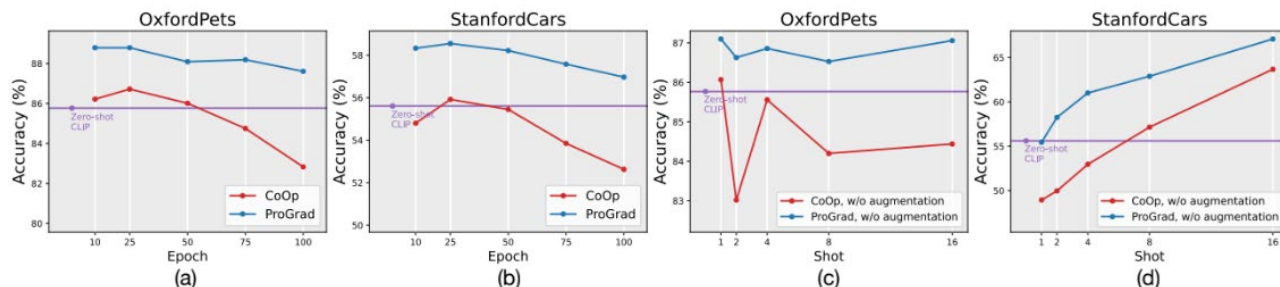
- Then, based on the **angle** formed between the gradients of KL loss and CE loss, the new gradient is set as follows:

$$\mathbf{G}_{\text{prograd}} = \begin{cases} \mathbf{G}_{\text{d}}, & \text{if } \mathbf{G}_{\text{d}} \cdot \mathbf{G}_{\text{g}} \geq 0 \\ \mathbf{G}_{\text{d}} - \lambda \cdot \frac{\mathbf{G}_{\text{d}} \cdot \mathbf{G}_{\text{g}}}{\|\mathbf{G}_{\text{g}}\|^2} \mathbf{G}_{\text{g}}, & \text{otherwise.} \end{cases}$$

- $\lambda=1$  means **orthogonal** projection, while  $\lambda=0$  means CoOp

# Results - ProGrad

- Better **generalization** with new classes, **outperforming** CLIP and CoOp



				Source	Target										
	Base	New	H.												
				ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	Average
CLIP	61.72	65.91	63.64												
CoOp	71.96	61.26	65.58												
CoCoOp	72.23	60.77	65.35												
ProGrad	<b>73.29</b>	<b>65.96</b>	<b>69.06</b>												

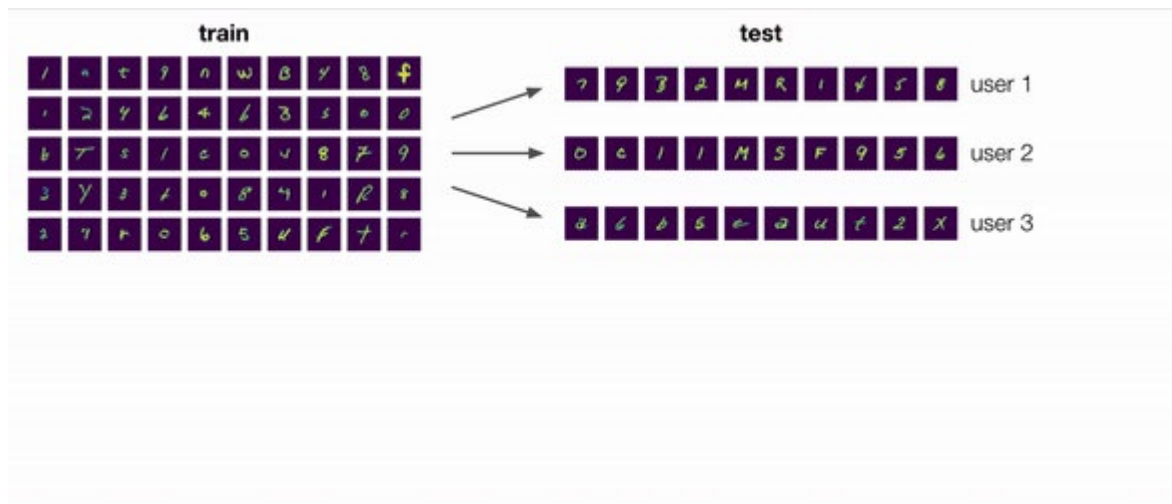
# Notable mentions

- Unified Vision and Language Prompt Learning (2022)
  - start from a unified prompt and split it in a text and visual prompt
- Knowledge-Aware Prompt Tuning for Generalizable Vision-Language Models
  - use Wikipedia as external knowledge base, appending class name to both continuous and discrete prompts

## **Second Part:** Prompt Tuning with Test-Time Adaption

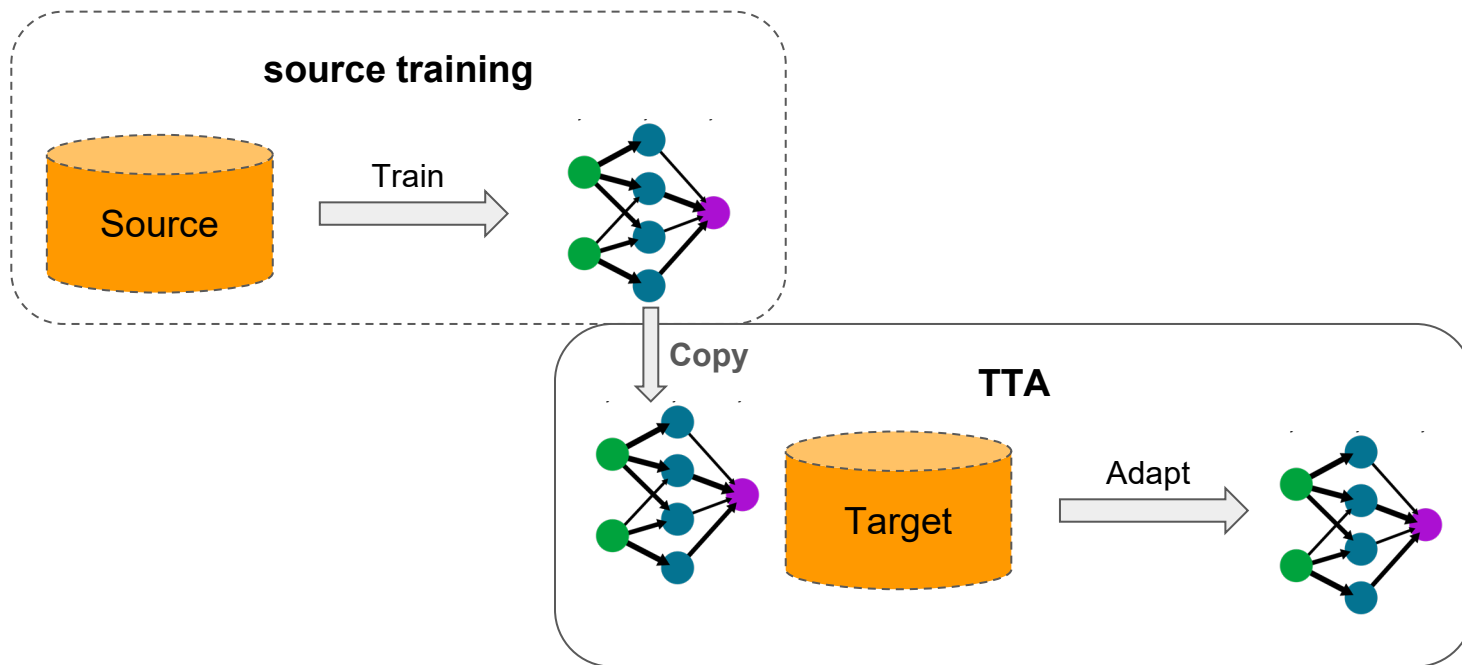
# Why Test-Time Adaption?

- Domain Shifts Happens in Natural Data
  - Testing data varies in many ways (Like style, light condition, sim-to-real etc.)
  - Yet Model remains same
  - Model fail to generalize
- Training data with annotations, which can be expensive and is not available for ZSL.



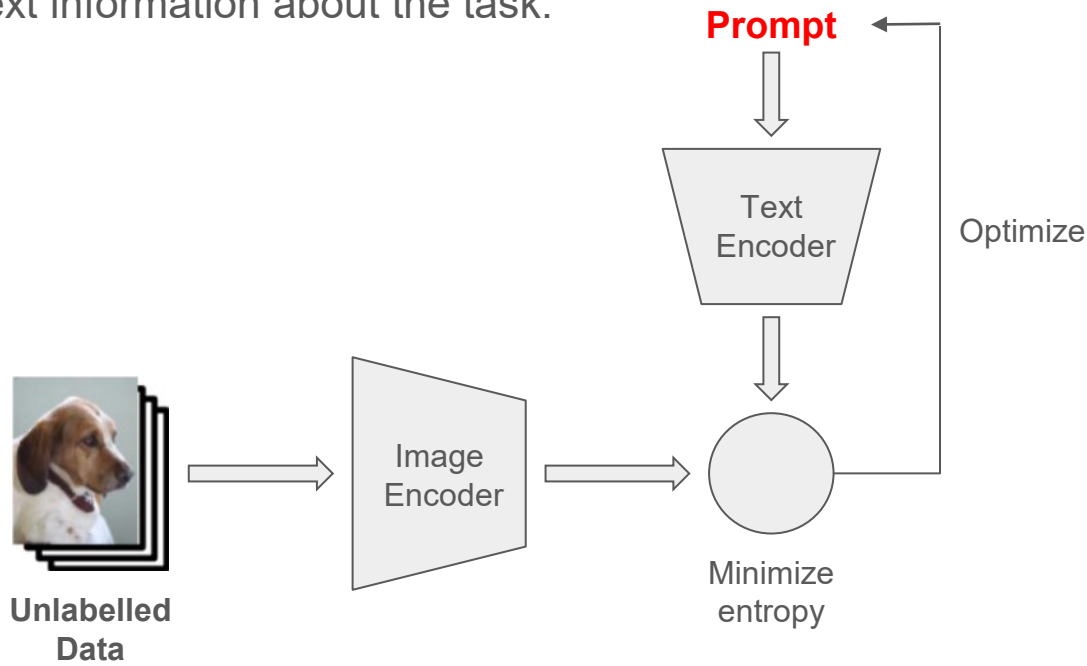
# Test-Time Adaption

- Test-Time Adaptation (TTA) doesn't Access Source Data



# Test-Time Prompting

- Test-time prompting optimizes the prompt (**p**) using target data from the downstream task at test time.
  - Goal is to obtain text inputs  $\{p; Y\} = \{\{p; y_i\} \text{ for } y_i \in Y\}$  that can provide the model with the most helpful context information about the task.
  - To improve model ZSL.





# Test-time prompt tuning for zero-shot generalization in Vision Language Models

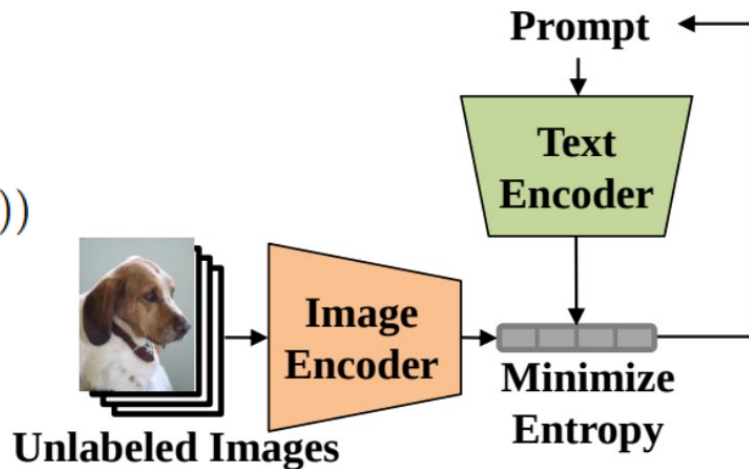
**Test-Time Prompt Tuning (TPT):** A method that can learn adaptive prompts on the fly with a single test sample.

**TPT** optimize the prompt  $\mathbf{p}$  at test time based on the single test sample  $\mathbf{X}_{\text{test}}$ .

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \mathcal{L}(\mathcal{F}, \mathbf{p}, \mathbf{X}_{\text{test}})$$

Where

$$\mathcal{F}_{\mathbf{p}}(X) = \text{sim}(\mathbf{E}_{\text{text}}(\{\mathbf{p}; \mathcal{Y}\}), \mathbf{E}_{\text{visual}}(X))$$



# TPT: Image Classification

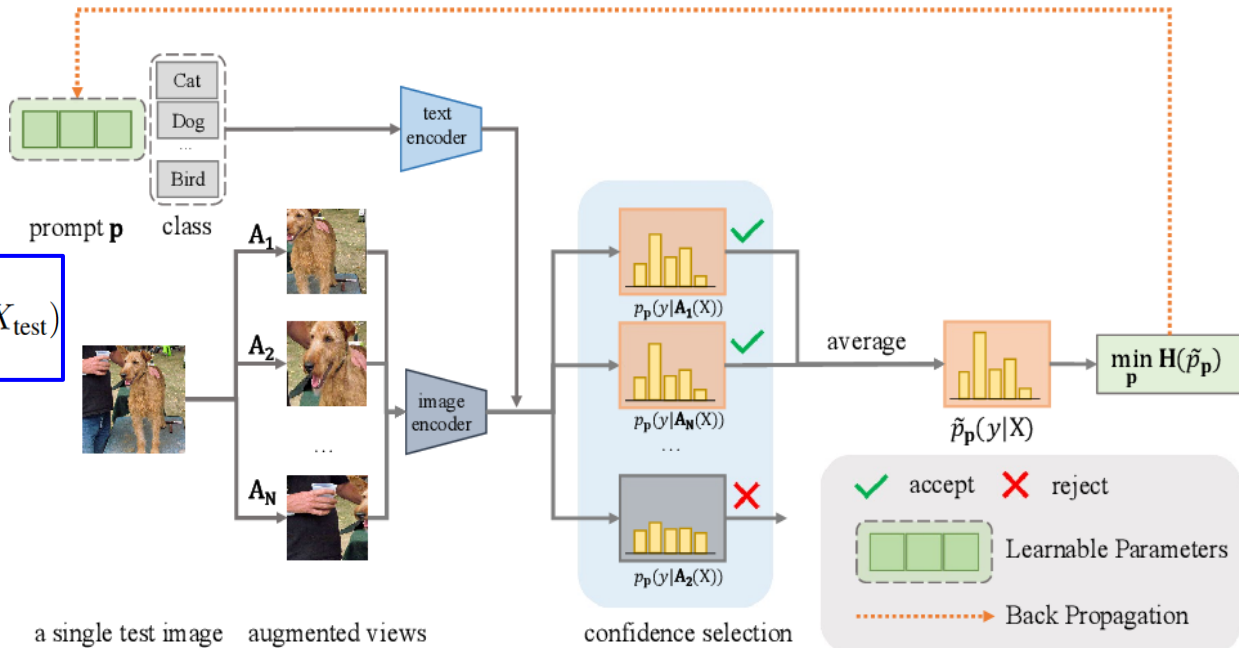
**Objective:** TPT optimizes predictions by adjusting prompts across augmented views of a given test image.

Minimize the entropy of the averaged prediction probability distribution across  $\mathbf{A}$  of  $\mathbf{X}_{\text{test}}$ .

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} - \sum_{i=1}^K \tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}}) \log \tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}})$$

Where

$$\tilde{p}_{\mathbf{p}}(y_i | X_{\text{test}}) = \frac{1}{N} \sum_{i=1}^N p_{\mathbf{p}}(y_i | \mathcal{A}_i(X_{\text{test}}))$$

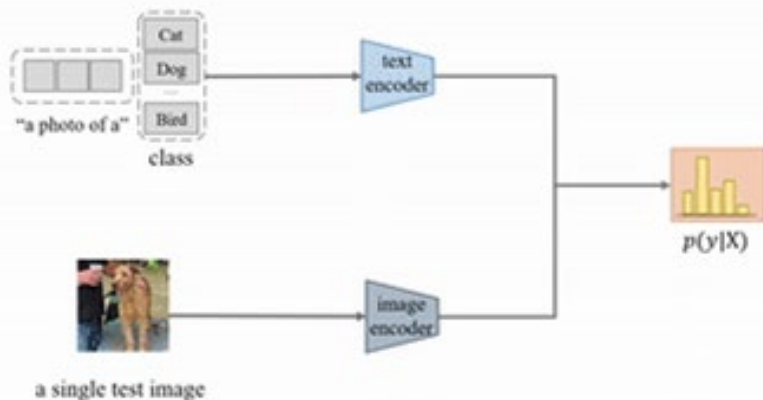


# TPT: Image Classification

**TPT** Propose confidence selection to filter out views that generate high-entropy with a prediction entropy below a threshold  $\tau$

$$\tilde{p}_{\mathbf{p}}(y|X_{\text{test}}) = \frac{1}{\rho N} \sum_{i=1}^N \mathbb{1}[\mathbf{H}(p_i) \leq \tau] p_{\mathbf{p}}(y|\mathcal{A}_i(X_{\text{test}}))$$

- Where  $\tau$  is is adaptive threshold.
- And  $H(p_i)$  is is the self-entropy of the predictions on  $\mathbf{A}_i$ .



# TPT: Context-Dependent Visual Reasoning

- In Bongard-HOI, the task involves context-dependent visual reasoning.
- Test sample comprises two sets of support images and a query image for evaluation.
- The TPT objective for context-dependent reasoning:

$$p^* = \arg \min_{p, cls} \frac{1}{M} \sum_{X \in \{X_P, X_N\}} \mathcal{L}(\mathcal{F}_{c,cls}(X), y)$$

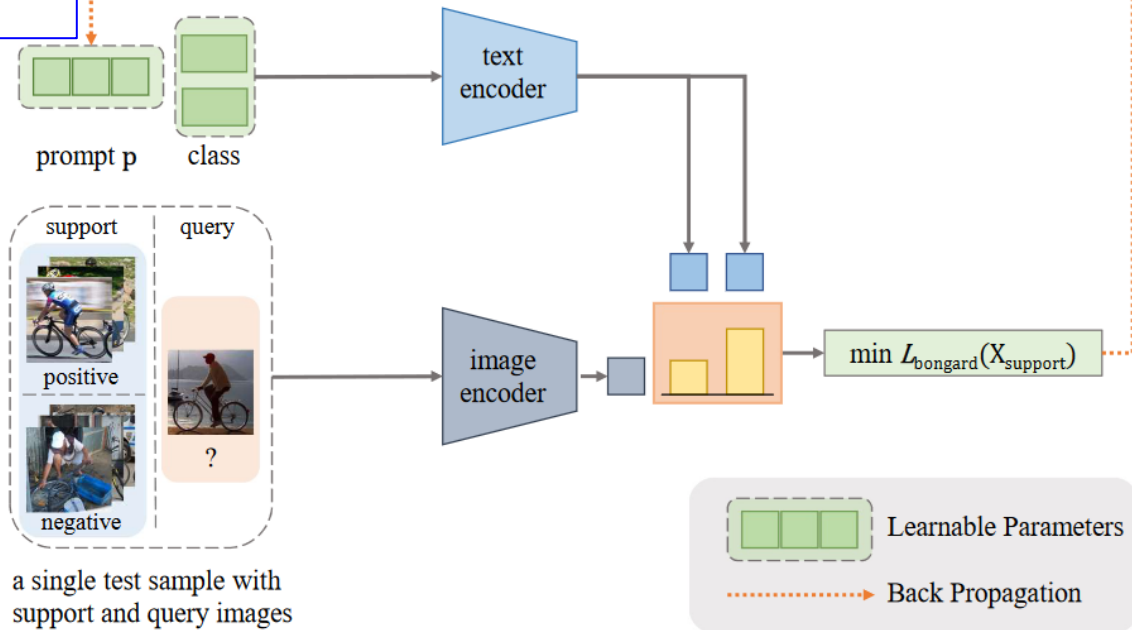
**M** = number of support images

**Binary label tokens:**

$y \in \{0, 1\} \Rightarrow$  binary label tokens **cls**

**cls** = {cls1, cls2}

**Text input to CLIP:**  $T = \{T1, T2 \mid Ti = \{p, cls_i\}\}$ .



a single test sample with support and query images

# TPT Experiments: Robustness to Distribution Shifts

## Datasets:

- Follow the setting in CLIP
- Evaluation robustness on 4 ImageNet Variants:
  - **ImageNet-V2** - independent test set containing natural images. test sets were re-sampled.
  - **ImageNet-A** - test set of natural adversarial examples
  - **ImageNet-R** - collects images of ImageNet categories but with artistic renditions
  - **ImageNet-Sketch** - black and white sketches

## Baselines:

- **CoOp**
- **CoCoOp**
- **CLIP-default-prompt**: “a photo of a”
- **CLIP-prompt-ensemble**: ensemble of 80 handcrafted prompts

# TPT Experiments: Robustness to Distribution Shifts

TPT improves the zero-shot top-1 accuracy of CLIP by 3.6% on average

Method	ImageNet Top1 acc. ↑	ImageNet-A Top1 acc. ↑	ImageNet-V2. Top1 acc. ↑	ImageNet-R. Top1 acc. ↑	ImageNet-Sketch Top1 acc. ↑	Average	OOD Average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
Ensemble	59.81	23.24	52.91	60.72	35.48	46.43	43.09
CoOp	63.33	23.06	55.40	56.60	34.67	46.61	42.43
CoCoOp	62.81	23.32	55.72	57.74	34.48	46.81	42.82
TPT	60.74	26.67	54.70	59.11	35.09	47.26	43.89
TPT + CoOp	64.73	30.32	57.83	58.99	35.86	49.55	45.75
TPT + CoCoOp	62.93	27.40	56.60	59.88	35.43	48.45	44.83
CLIP-ViT-B/16	66.73	47.87	60.86	73.98	46.09	59.11	57.2
Ensemble	68.34	49.89	61.88	77.65	48.24	61.20	59.42
CoOp	71.51	49.71	64.20	75.21	47.99	61.72	59.28
CoCoOp	71.02	50.63	64.07	76.18	48.75	62.13	59.91
TPT	68.98	54.77	63.45	77.06	47.94	62.44	60.81
TPT + CoOp	73.61	57.95	66.83	77.27	49.29	64.99	62.83
TPT + CoCoOp	71.07	58.47	64.85	78.65	48.47	64.30	62.61

# TPT Experiments: Cross-Datasets Generalization

- 10 datasets including plants, animals, scenes, textures etc.
- Two settings:
  - ImageNet as a comprehensive source dataset, fine-tuned datasets for evaluation
  - Fine-tuned datasets are both source and target with no overlaps
- TPT performs on par with the state-of-the-art approaches without training

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Average
CLIP-RN50	61.75	40.37	83.57	55.70	58.84	85.88	73.97	58.80	15.66	23.69	55.82
Ensemble	62.77	40.37	82.97	55.89	59.48	<b>87.26</b>	74.82	60.85	16.11	25.79	56.63
CoOp	61.55	37.29	87.00	55.32	59.05	86.53	75.59	58.15	15.12	26.20	56.18
CoCoOp	<b>65.57</b>	38.53	<b>88.39</b>	56.22	57.10	87.38	<b>76.2</b>	59.61	14.61	<b>28.73</b>	57.23
TPT	62.69	<b>40.84</b>	84.49	<b>58.46</b>	<b>60.82</b>	87.02	74.88	<b>61.46</b>	<b>17.58</b>	28.33	<b>57.66</b>
CLIP-ViT-B/16	67.44	44.27	88.25	65.48	65.13	93.35	83.65	62.59	23.67	42.01	63.58
Ensemble	66.99	45.04	86.92	66.11	65.16	93.55	82.86	65.63	23.22	<b>50.42</b>	64.59
CoOp	68.71	41.92	89.14	64.51	66.55	93.70	<b>85.30</b>	64.15	18.47	<b>46.39</b>	63.88
CoCoOp	<b>70.85</b>	45.45	<b>90.46</b>	64.90	<b>68.44</b>	<b>93.79</b>	83.97	<b>66.89</b>	22.29	39.23	64.63
TPT	68.98	<b>47.75</b>	87.79	<b>66.87</b>	68.04	94.16	84.67	65.5	<b>24.78</b>	42.44	<b>65.10</b>

# TPT Experiments: Visual Reasoning

Baselines:

- CNN Baseline
- The Meta-baseline regards each sample as a few shot task.
- HOITrans: transformer-based approach that achieves state-of-the-art accuracy on HOI

**Results:** TPT outperform the state-of-the-art method by 4.1% Bongard-HOI benchmark.

Method	Test Splits				Average
	seen act., seen obj.,	unseen act., seen obj.,	seen act., unseen obj.,	unseen act., unseen obj.,	
CNN-baseline	50.03	49.89	49.77	50.01	49.92
Meta-baseline*	58.82	58.75	58.56	57.04	58.30
HOITrans	59.50	64.38	63.10	62.87	62.46
<b>TPT (w/ CLIP-RN50)</b>	<b>66.39</b>	<b>68.50</b>	<b>65.98</b>	<b>65.48</b>	<b>66.59</b>



# TPT: Strength & Weaknesses

## Pros:

- Does not requires additional data or supervision
- Even without additional pre-training, the model improves the performance
- One step of optimization can increase the performance

## Limitations

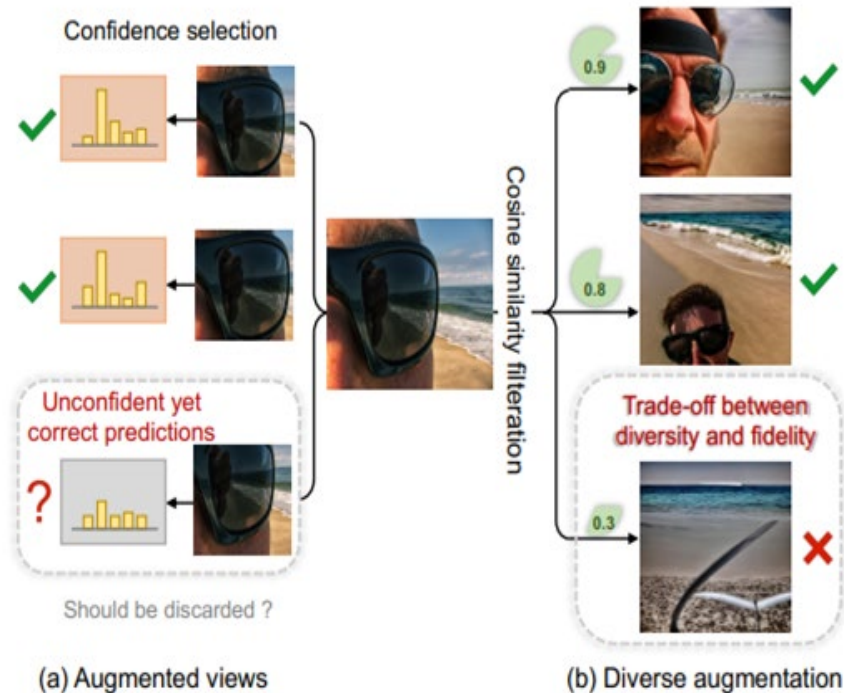
- Existing TPT methods typically rely on data augmentation and confidence selection.
- Conventional data augmentation techniques, such as random resized crops, are noted to suffer from a lack of data diversity.
- Entropy-based confidence selection alone is deemed insufficient for ensuring prediction fidelity.
- Distribution Shift.

**Solutions?**

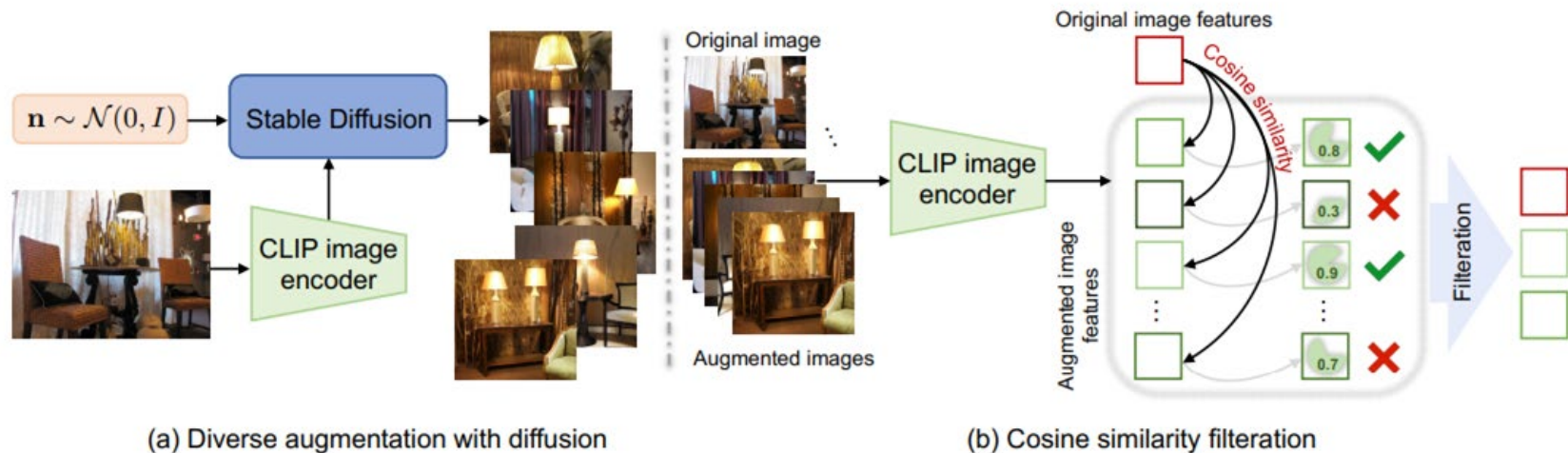
# Diverse Data Augmentation with Diffusions for Effective Test-time Prompt Tuning (DiffTPT)

The key contributions include:

- A balanced trade-off between data diversity and prediction fidelity,
- Diffusion-based augmentation for generating richer visual variations,
- Cosine similarity filtration for removing spurious augmentations, and significant performance gains over existing TPT methods



# Architecture:



**Overview** of our proposed **DiffTPT**. We first **(a)** use the pre-trained stable diffusion to generate data with richer visual appearance variation, then **(b)** uses a cosine similarity based filtration with the single test sample to remove spurious augmentations, making our method a **trade-off between diversity and fidelity**.

# Experiments & Results:

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-Sk.	Average	OOD Avg.
CLIP-RN50	58.10(bs.)	22.81(bs.)	53.00(bs.)	53.90(bs.)	33.50(bs.)	42.26(bs.)	40.80(bs.)
Ensemble	59.90(1.80) ↑	24.12(1.31) ↑	53.50(0.50) ↑	58.00(4.10) ↑	35.20(1.70) ↑	46.14(3.88) ↑	42.70(1.90) ↑
TPT	59.40(1.30) ↑	27.34(4.53) ↑	55.20(2.20) ↑	56.80(2.90) ↑	34.50(1.00) ↑	46.65(4.39) ↑	43.46(2.66) ↑
<b>DiffTPT</b>	<b>60.80(2.70) ↑</b>	<b>31.06(8.25) ↑</b>	<b>55.80(2.80) ↑</b>	<b>58.80(4.90) ↑</b>	<b>37.10(3.60) ↑</b>	<b>48.71(6.45) ↑</b>	<b>45.69(4.89) ↑</b>
CoOp	63.30(bs.)	24.52(bs.)	57.90(bs.)	55.10(bs.)	34.80(bs.)	47.12(bs.)	43.08(bs.)
TPT&CoOp	63.70(0.40) ↑	29.75(5.23) ↑	60.90(3.00) ↑	57.80(2.70) ↑	36.50(1.70) ↑	49.73(2.61) ↑	46.24(3.16) ↑
<b>DiffTPT&amp;CoOp</b>	<b>64.70(1.40) ↑</b>	<b>32.96(8.44) ↑</b>	<b>61.70(3.80) ↑</b>	<b>58.20(3.10) ↑</b>	<b>36.80(2.00) ↑</b>	<b>50.87(3.75) ↑</b>	<b>47.42(4.34) ↑</b>
CoCoOp	61.50(bs.)	25.73(bs.)	54.80(bs.)	56.00(bs.)	35.50(bs.)	46.71(bs.)	43.01(bs.)
TPT&CoCoOp	62.40(0.90) ↑	26.43(0.70) ↑	56.10(1.30) ↑	56.50(0.50) ↑	35.60(0.10) ↑	47.41(0.70) ↑	43.66(0.65) ↑
<b>DiffTPT&amp;CoCoOp</b>	<b>63.50(2.00) ↑</b>	<b>30.45(0.72) ↑</b>	<b>57.70(2.90) ↑</b>	<b>58.50(2.50) ↑</b>	<b>37.90(2.40) ↑</b>	<b>49.61(2.90) ↑</b>	<b>46.14(3.13) ↑</b>
CLIP-ViT-B/16	67.30(bs.)	47.14(bs.)	59.90(bs.)	71.20(bs.)	43.00(bs.)	57.71(bs.)	55.31(bs.)
Ensemble	68.50(1.20) ↑	48.44(1.30) ↑	62.70(2.80) ↑	73.50(2.30) ↑	45.50(2.20) ↑	59.73(2.02) ↑	57.53(2.22) ↑
TPT	69.70(2.40) ↑	53.67(6.53) ↑	64.30(4.40) ↑	73.90(2.70) ↑	46.40(3.40) ↑	61.59(3.88) ↑	59.57(4.26) ↑
<b>DiffTPT</b>	<b>70.30(3.00) ↑</b>	<b>55.68(8.54) ↑</b>	<b>65.10(5.20) ↑</b>	<b>75.00(3.80) ↑</b>	<b>46.80(3.80) ↑</b>	<b>62.28(4.57) ↑</b>	<b>60.52(5.21) ↑</b>
CoOp	72.30(bs.)	49.25(bs.)	65.70(bs.)	71.50(bs.)	47.60(bs.)	61.27(bs.)	58.51(bs.)
TPT&CoOp	73.30(1.00) ↑	56.88(7.63) ↑	66.60(0.90) ↑	73.80(2.30) ↑	49.40(1.80) ↑	64.00(2.73) ↑	61.67(3.16) ↑
<b>DiffTPT&amp;CoOp</b>	<b>75.00(2.70) ↑</b>	<b>58.09(8.84) ↑</b>	<b>66.80(1.10) ↑</b>	<b>73.90(2.40) ↑</b>	<b>49.50(1.90) ↑</b>	<b>64.12(2.85) ↑</b>	<b>61.97(3.46) ↑</b>
CoCoOp	71.40(bs.)	50.05(bs.)	63.80(bs.)	73.10(bs.)	46.70(bs.)	61.01(bs.)	58.41(bs.)
TPT&CoCoOp	67.30(4.10) ↓	50.25(0.20) ↑	62.30(1.50) ↓	73.90(0.80) ↑	47.10(0.40) ↑	60.17(0.84) ↓	58.39(0.02) ↓
<b>DiffTPT&amp;CoCoOp</b>	<b>69.30(2.10) ↓</b>	<b>52.56(2.51) ↑</b>	<b>63.20(0.60) ↓</b>	<b>75.30(2.20) ↑</b>	<b>47.50(0.80) ↑</b>	<b>61.57(0.56) ↑</b>	<b>59.64(1.23) ↑</b>

**Top 1 accuracy** % of state-of-the-art baselines under S1, where **ImageNet-Sk.** indicates the ImageNet-Sketch dataset, **OOD Avg.** indicates the OOD average results. bs. indicates the baseline of each group, i.e., CLIP-RN50 / CLIPViT-B-16, CoOp, and CoCoOp. The arrow ↑ and ↓ indicate improvements and decrements compared with bs.

Method	Flower [34]	DTD [8]	Pets [35]	Cars [26]	UCF101 [49]
CLIP-RN50	62.45( <i>bs.</i> )	39.65( <i>bs.</i> )	80.50( <i>bs.</i> )	57.48( <i>bs.</i> )	56.73( <i>bs.</i> )
Ensemble	63.14	41.68	80.79	58.33	55.74
CoOp <sub>2022</sub> [62]	62.25	37.33	86.00	56.29	59.01
CoCoOp <sub>2022</sub> [61]	63.53	38.49	86.29	55.70	60.40
TPT <sub>2022</sub> [46]	62.25(0.20) ↓	40.04(0.39) ↑	82.82(2.32) ↑	60.54(3.06) ↑	60.79(4.06) ↑
<b>DiffTPT</b>	<b>63.53(1.08) ↑</b>	<b>40.72(1.07) ↑</b>	<b>83.40(3.35) ↑</b>	<b>60.71(3.23) ↑</b>	<b>62.67(5.94) ↑</b>

Method	Caltech11 [11]	Food101 [4]	SUN397 [53]	Aircraft [29]	EuroSAT [16]	Avg.
CLIP-RN50	81.58( <i>bs.</i> )	74.85( <i>bs.</i> )	57.43( <i>bs.</i> )	16.20( <i>bs.</i> )	24.30( <i>bs.</i> )	55.12( <i>bs.</i> )
Ensemble	83.68	74.95	59.53	17.40	27.69	56.29
CoOp <sub>2022</sub> [62]	82.38	78.81	57.18	15.40	26.99	56.16
CoCoOp <sub>2022</sub> [61]	83.38	77.43	59.28	15.70	27.39	56.76
TPT <sub>2022</sub> [46]	84.58(3.00) ↑	77.23(2.38) ↑	61.80(4.37) ↑	17.50(1.30) ↑	22.21(2.09) ↓	56.98(1.86) ↑
<b>DiffTPT</b>	<b>86.89(5.31) ↑</b>	<b>79.21(4.36) ↑</b>	<b>62.72(5.29) ↑</b>	<b>17.60(1.40) ↑</b>	<b>41.04(16.74) ↑</b>	<b>59.85(4.68) ↑</b>

Method	Flower [34]	DTD [8]	Pets [35]	Cars [26]	UCF101 [49]
CLIP-ViT-B/16	67.94( <i>bs.</i> )	44.10( <i>bs.</i> )	85.71( <i>bs.</i> )	66.58( <i>bs.</i> )	63.37( <i>bs.</i> )
Ensemble	67.65	44.87	86.20	67.60	64.36
CoOp <sub>2022</sub> [62]	66.08	42.17	89.00	63.44	66.04
CoCoOp <sub>2022</sub> [61]	70.88	44.78	88.71	65.22	68.42
TPT <sub>2022</sub> [46]	69.31(1.37) ↑	46.23(2.13) ↑	86.49(0.78) ↑	66.50(0.08) ↓	66.44(3.07) ↑
<b>DiffTPT</b>	<b>70.10(2.16) ↑</b>	<b>47.00(2.90) ↑</b>	<b>88.22(2.51) ↑</b>	<b>67.01(0.43) ↑</b>	<b>68.22(4.85) ↑</b>

Method	Caltech11 [11]	Food101 [4]	SUN397 [53]	Aircraft [29]	EuroSAT [16]	Avg.
CLIP-ViT-B/16	90.29( <i>bs.</i> )	85.05( <i>bs.</i> )	61.88( <i>bs.</i> )	24.70( <i>bs.</i> )	40.64( <i>bs.</i> )	63.03( <i>bs.</i> )
Ensemble	90.89	85.35	64.65	24.40	47.01	64.30
CoOp <sub>2022</sub> [62]	91.69	85.15	61.54	18.00	35.36	61.85
CoCoOp <sub>2022</sub> [61]	92.49	86.53	64.65	24.20	46.22	65.21
TPT <sub>2022</sub> [46]	92.49(2.20) ↑	86.93(1.88) ↑	63.48(1.60) ↑	24.90(0.20) ↑	37.15(3.49) ↓	63.99(0.96) ↑
<b>DiffTPT</b>	<b>92.49(2.20) ↑</b>	<b>87.23(2.18) ↑</b>	<b>65.74(3.86) ↑</b>	<b>25.60(0.90) ↑</b>	<b>43.13(3.49) ↑</b>	<b>65.47(2.44) ↑</b>

**Top 1 accuracy** % of state-of-the-art baselines under S2, where Avg. indicates average accuracies of the Cross-Datasets Generalization. The arrow ↑ and ↓ indicate improvements and decrements of our method against the CLIP method, i.e., CLIP-RN50 and CLIP-ViT-B/16.

# Test-Time Prompting with Distribution Alignment for Zero-Shot Generalization

## Key Points:

PromptAlign utilizes multi-modal prompt learning to handle distribution shift in each test sample.

For each test sample, multiple augmented views are passed through the visual encoder with learnable prompts.

Token alignment loss is computed between test sample and source dataset statistics (mimicking CLIP's pre-training data).

The final objective combines entropy and alignment losses to update prompts for a given test sample.

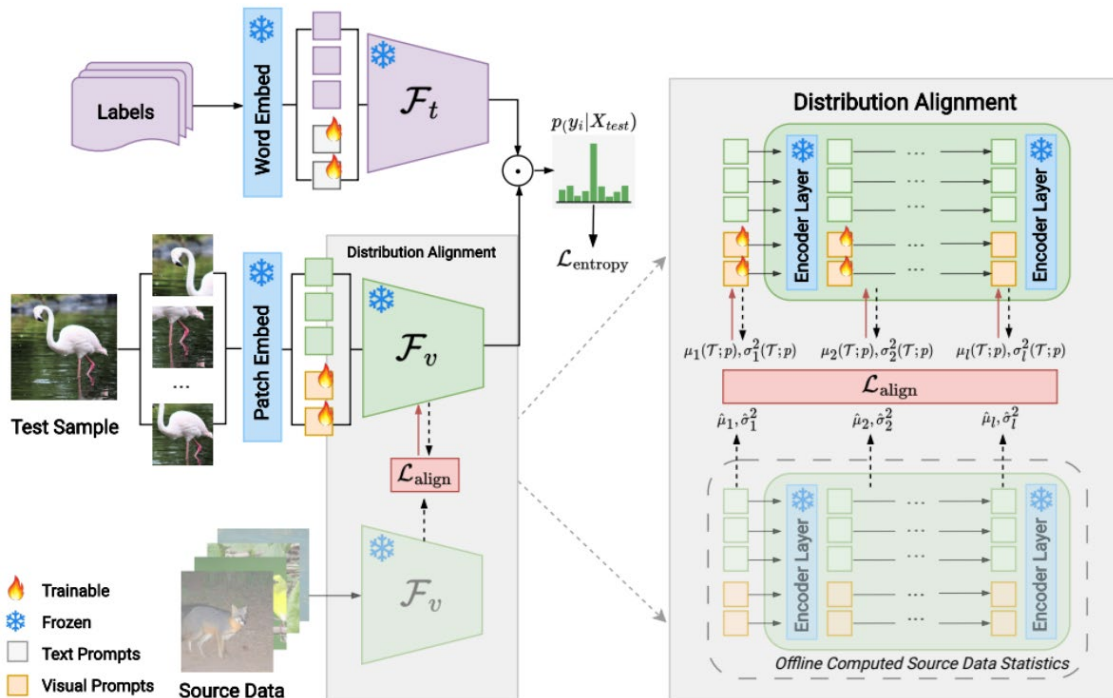
# PromptAlign: Architecture

At test time, a single test sample along with its augmented views is passed through the CLIP image encoder, and the text labels are passed to the CLIP text encoder.

The token distribution statistics – mean and variance – of the test sample are aligned with the offline computed source data statistics using a distribution alignment loss.

The resulting alignment loss from the distribution shift is combined with the entropy loss to update the multi-modal prompts.

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{entropy}} + \beta \mathcal{L}_{\text{align}}$$





# PromptAlign: Experiments & Results

## Datasets:

- For the domain generalization setting, evaluate the four OOD datasets:
  - ImageNetV2, ImageNet-Sketch, ImageNet-A, and ImageNet-R.
- Cross dataset OOD evaluation, PromptAlign follows TPT and evaluate on 10 datasets (food, flowers, cars, pets etc.)

## Baselines:

- CLIP
- CoOp
- CoCoOp
- TPT
- MaPLe

Table 2: **Comparison of PromptAlign in domain generalization setting.** Prompt learning methods are trained on ImageNet and evaluated on datasets with domain shifts.

	Imagenet V2	Imagenet Sketch	Imagenet A	Imagenet R	OOD Avg.
CLIP [28]	60.86	46.09	47.87	73.98	57.20
CLIP+TPT [32]	64.35	47.94	54.77	77.06	60.81
CoOp [46]	64.20	47.99	49.71	75.21	59.28
CoOp+TPT [32]	<b>66.83</b>	49.29	57.95	77.27	62.84
Co-CoOp [45]	64.07	48.75	50.63	76.18	59.91
Co-CoOp+TPT [32]	64.85	48.27	58.47	78.65	62.61
<b>PromptAlign</b>	65.29	<b>50.23</b>	<b>59.37</b>	<b>79.33</b>	<b>63.55</b>

# PromptAlign: In Cross-Dataset Evaluation

PromptAlign consistently improves upon TPT here as well. Overall it performs well except Caltech, DTD, SUN397 and EuroSAT. Overall average it performs bit well than other methods.

	Caltech	Pets	Cars	Flowers	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CLIP [28]	93.35	88.25	65.48	67.44	83.65	23.67	62.59	44.27	42.01	65.13	63.58
CLIP+TPT [32]	<b>94.16</b>	87.79	66.87	68.98	84.67	24.78	65.50	<b>47.75</b>	42.44	68.04	65.10
CoOp [46]	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp [45]	93.79	90.46	64.90	70.85	83.97	22.29	66.89	45.45	39.23	68.44	64.63
ProDA [44]	86.70	88.20	60.10	77.50	80.80	22.20	-	50.90	58.50	-	65.62
MaPLe	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	<b>48.06</b>	68.69	66.30
MaPLe+TPT	93.59	90.72	66.50	72.37	86.64	24.70	<b>67.54</b>	45.87	47.80	69.19	66.50
PromptAlign	94.01	<b>90.76</b>	<b>68.50</b>	<b>72.39</b>	<b>86.65</b>	<b>24.80</b>	<b>67.54</b>	47.24	47.86	<b>69.47</b>	<b>66.92</b>

# Future Works

- Compare performance against Zero-shot Clip
- Compare performance within each others.
- Blend together prompt learning methods with test-time adaptation techniques
- Compare the performance of prompt learning methods with and without test-time adaptation
- Experiment with various hyperparameters

Thank you for your attention!

# Bibliography

- Learning to Prompt for Vision-Language Models [Kaiyang Zhou et al, 2021]
- Learning transferable visual models from natural language supervision. [Radford A. et al, 2021]
- Conditional Prompt Learning for Vision-Language Models [Kaiyang Zhou et al, 2022]
- MaPLe: Multi-modal Prompt Learning [Muhammad Uzair Khattak et al, 2023]
- Prompt Pre-Training with Twenty-Thousand Classes for Open-Vocabulary Visual Recognition [Ren Shuhuai et al, 2023]
- Visual-Language Prompt Tuning with Knowledge-guided Context Optimization [Yao Hantao et al, 2023]
- Prompt-aligned Gradient for Prompt Tuning [Zhu Beier et al, 2022]

# Bibliography

- Lester, Brian, et al. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691, arXiv, 2 Sept. 2021
- Jia, Menglin, et al. Visual Prompt Tuning. arXiv:2203.12119, arXiv, 20 July 2022
- Shu, Manli, et al. Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models. arXiv:2209.07511, arXiv, 15 Sept. 2022
- Gao, Yunhe, et al. Visual Prompt Tuning for Test-Time Domain Adaptation. arXiv:2210.04831, arXiv, 30 Nov. 2022
- Feng, Chun-Mei, et al. Diverse Data Augmentation with Diffusions for Effective Test-Time Prompt Tuning. arXiv:2308.06038, arXiv, 17 Aug. 2023
- Hassan, Jameel, et al. “Align Your Prompts: Test-Time Prompting with Distribution Alignment for Zero-Shot Generalization.” ArXiv.org, 2 Nov. 2023