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

Efficient Test-Time Adaptation of Vision-Language Models

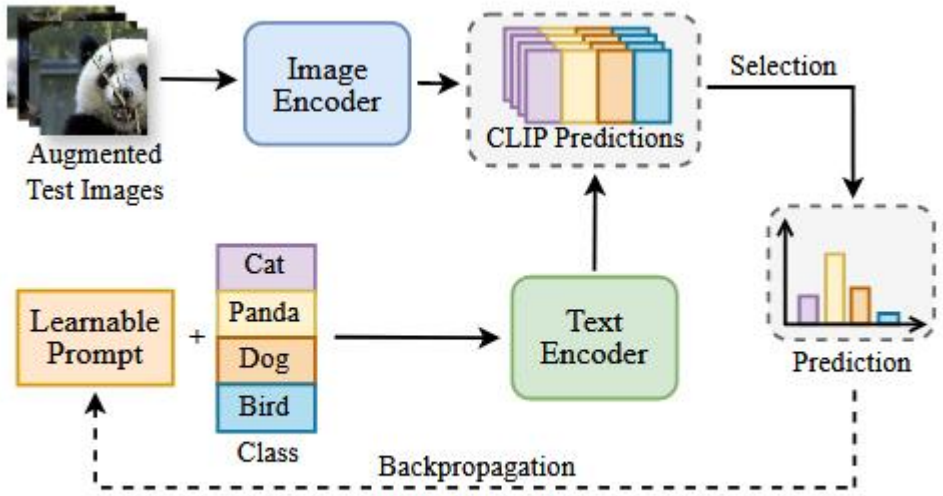
Adilbek Karmanov[1*] Dayan Guan[2*] Shijian Lu[1,2†] Abdulmotaleb El Saddik[1,3] Eric Xing[1,4]
[1]Mohamed bin Zayed University of Artificial Intelligence [2]Nanyang Technological University
[3]University of Ottawa [4]Carnegie Mellon University

CVPR 2024

Preliminaries

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

TPT and its enhancement DiffTPT



(a) Test-time Prompt Tuning [9, 35]

TPT [35] focuses on test-time adaptation of CLIP. In TPT, a prompt tuning method is proposed to learn an adaptive prompt $\mathbf{p}_c$ using individual test samples. A set of augmentation functions $\mathcal{A}$ is used to generate n randomly augmented views $\tilde{x}_{\text{test}} = \mathcal{A}_n(x_{\text{test}})$ of a test sample x_{test}. The objective of TPT is to reduce variation in the model's predictions across different augmentations $\tilde{f}_{\text{test}} = E_v(\tilde{x}_{\text{test}})$ by minimizing the marginal entropy among the outputs of the augmented views. Furthermore, TPT also includes confidence selection to discard noisy augmentations that could result in ambiguous model predictions, as shown in Figure 1a. This is achieved by filtering out augmented views with high-entropy predictions as:

$$P_{\text{TPT}}(\tilde{f}_{\text{test}}) = \frac{1}{\rho n} \sum_{i=1}^{n} \mathbb{1}[\mathrm{H}(\tilde{f}_i \mathbf{p}_c^T) \leq \tau] \tilde{f}_i \mathbf{p}_c^T, \quad (1)$$

where H is the self-entropy function of the softmax logits predictions, $\tilde{f}_i \mathbf{p}_c^T$ is the class probabilities vector of size N generated from the given i-th augmented view of the test image, and the parameter τ determines that only ρ-percentile of confident samples with entropy values below this threshold can be selected out of n augmented views.

Preliminaries

Tip-Adapter

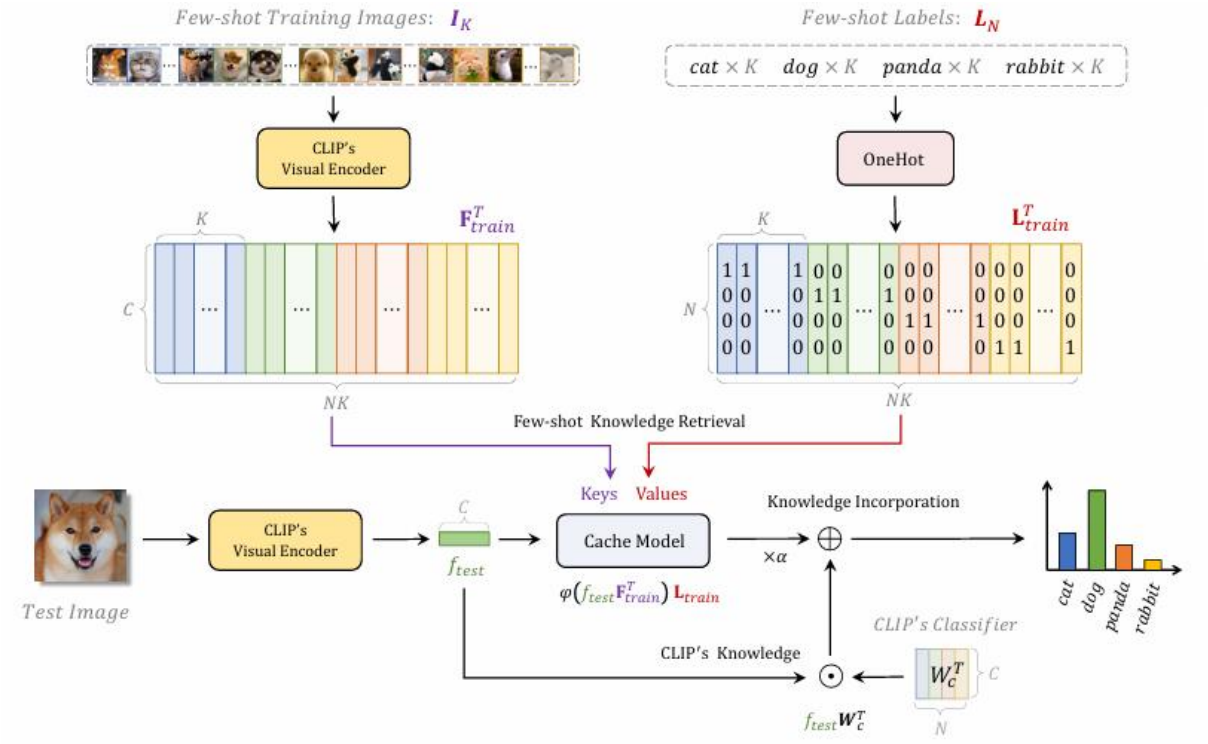


Fig. 1. The Pipeline of Tip-Adapter. Given a K-shot N-class training set, we construct a cache model to adapt CLIP on downstream tasks. It contains few-shot visual features $\mathbf{F}^T_{\text{train}}$ encoded by CLIP and their ground-truth labels $\mathbf{L}^T_{\text{train}}$ under one-hot encodings. After retrieval from the cache model, the few-shot knowledge is incorporated with CLIP's pre-trained knowledge, achieving the training-free adaption.

1-$f_{test}F^T_{train}$ ：计算测试特征f_{test} 与所有少镜头训练图像特征F_{train} 之间的欧氏距离

W***c***：文本分类器的权重，
将每个类别名称放入预定义好的prompt模板中，并通过clip预训练的文本编码器进行编码

$$P_{\text{cache}}(f_{\text{test}}) = A(f_{\text{test}}\mathbf{F}^T_{\text{train}})\mathbf{L}_{\text{train}}, \qquad (2)$$

where $A(z) = \alpha \exp(-\beta(1-z))$ is an adaptation function within a weighting factor α and a sharpness ratio β. During inference, the prediction of Tip-Adapter is computed by combining the pre-trained CLIP model and the cache model as: $P_{\text{TA}}(f_{\text{test}}) = P_{\text{cache}}(f_{\text{test}}) + f_{\text{test}}\mathbf{W}^T_c$.

Methods

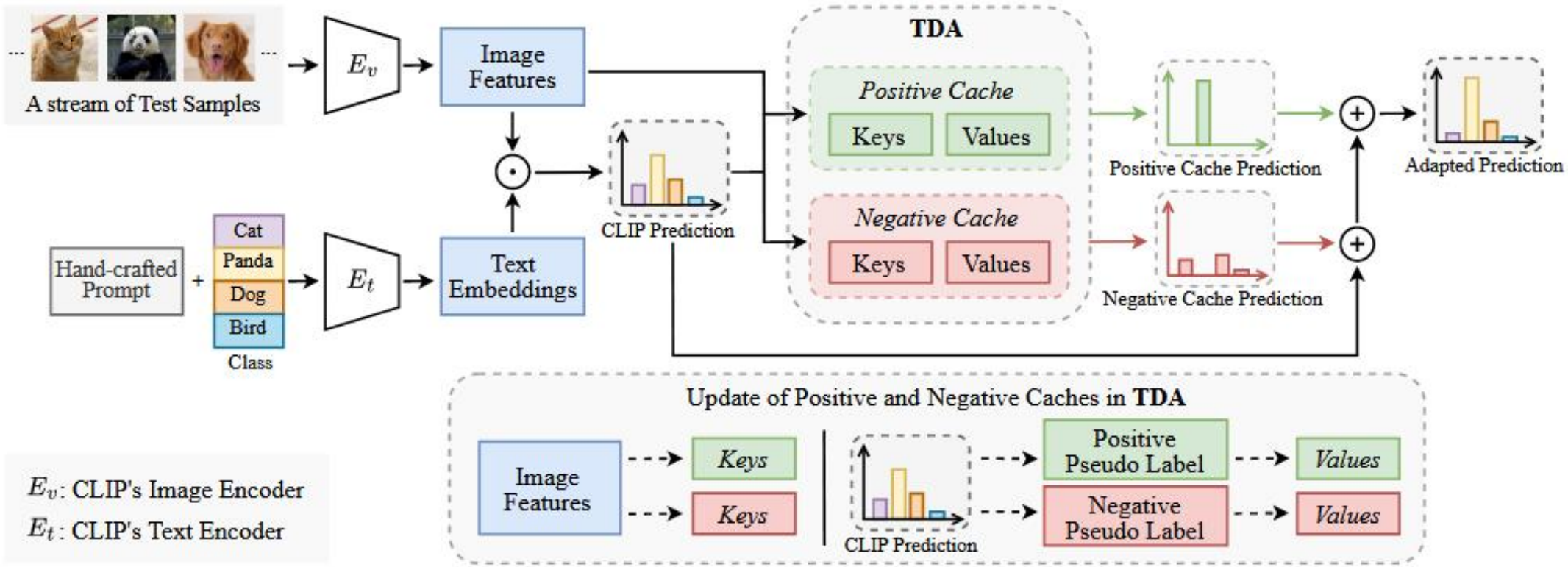


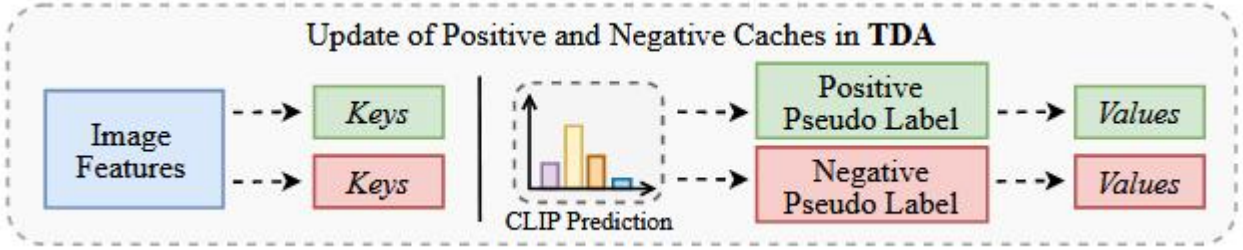
Figure 2. Overview of the proposed Training-free Dynamic Adapter (TDA). TDA constructs and updates two key-value caches to store the knowledge of a stream of test samples, and uses the two caches to generate positive and negative predictions which are combined with CLIP predictions to produce the final prediction. Specifically, the CLIP predictions are generated by performing the dot product between the image features generated by CLIP's image encoder E_v and the text embeddings generated by CLIP's text encoder E_t, using the hand-crafted prompt and class names. The two key-value caches are updated by gradually incorporating the test features and their corresponding pseudo labels calculated from CLIP's predictions, based on prediction entropy and cache capacity.

Methods

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

Positive Cache

It aims to collect high-quality few-shot pseudo labels ˆLp as positive values and the corresponding features Qp as keys.



作用：以高置信度预测更新KV键值对，生成高质量的伪标签。

生成伪标签：通过预测$\tilde{f}_i W_c^T$应用softmax函数，为每个测试样本x_{test}定义伪标签$\hat{l}$，这是一个分类分布的单热编码向量。

进入positive队列的标准：

①定义了关于缓存数量的规则。如果$\hat{L_P}$（可能是某类缓存中的伪标签集合）的数量（每个类别的收集对数）小于最大缓存数量k（每个类别的最大容量对数），那么 TDA 将把$\hat{l}$和f_{test}分别作为新值$\hat{L}_p$和新键添加到和Q_p（可能是分别存储值和键的结构）中。

②当数量达到最大的缓存数量k时，进行比较,将熵值高的k-v键值对移出去，保留下来kv键值对更小的。

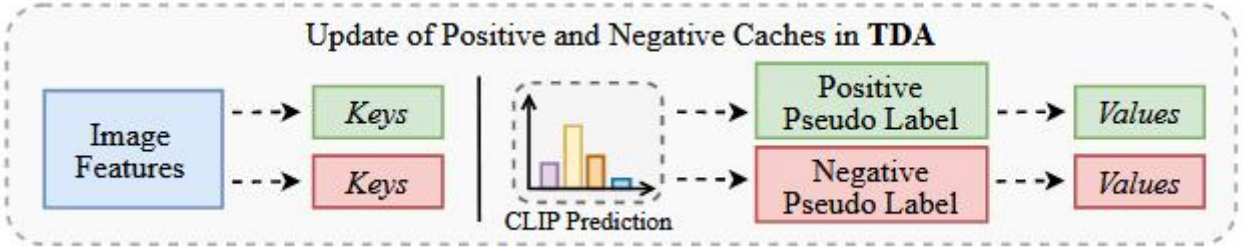
TDA可以在控制镜头容量的同时逐步整合具有较低熵的测试预测，这有助于确保在正缓存中收集高质量的伪标签。

$$P_{\text{pos}}(f_{\text{test}}) = A(f_{\text{test}}\mathbf{Q}_{\mathbf{p}}^{\mathbf{T}})\hat{\mathbf{L}}_{\mathbf{p}}, \qquad (3)$$

where A is the adaptation function defined in Tip-Adapter.

Methods

Negative Cache



作用：选择不确定性的阈值高的样本作为负伪标签，以避免对具有确定预测的数据产生偏差。

筛选负伪标签：

$$\hat{\mathbf{L}}_{\mathbf{n}} = -\mathbb{1}[p_l < P(\mathbf{Q_n})], \tag{4}$$

where higher probabilities than p_l are selected as negative pseudo labels from uncertain predictions and the uncertainty is measured by the entropy of predictions. Here, p_l

该条件旨在通过纳入表现出适度预测不确定性的测试样本，降低由于高熵或偏向某些预测（以非常低的熵为特征）而导致的预测错误的风险。

When constructing the negative cache, the testing feature f_{test} will be included in negative cache if it satisfies the condition $\gamma(f_{\text{test}})$: the entropy of the prediction is in the specified interval between τ_l and τ_h:

$$\gamma(f_{\text{test}}) : \tau_l < \mathrm{H}(f_{\text{test}}\mathbf{W}_c^T) < \tau_h. \tag{5}$$

进入negative Cache的条件和positive Cache条件一样

$$P_{\text{neg}}(f_{\text{test}}) = -A(f_{\text{test}}\mathbf{Q}_{\mathbf{n}}^{\mathbf{T}})\hat{\mathbf{L}}_{\mathbf{n}}, \tag{6}$$

where A is the adaptation function defined in Tip-Adapter. The predictions of TDA can be formulated by combining the negative cache, the positive cache and the pre-trained CLIP model together as follows:

$$P_{\text{TDA}}(f_{\text{test}}) = f_{\text{test}}\mathbf{W}_c^T + P_{\text{pos}}(f_{\text{test}}) + P_{\text{neg}}(f_{\text{test}}). \tag{7}$$

Experiments

The OOD benchmark serves as a measure of the robustness of our approach by involving assessment on 4 out-of-distribution datasets

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-S	Average	OOD Average
CLIP-ResNet-50	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
Tip-Adapter	62.03	23.13	53.97	60.35	35.74	47.04	43.30
TPT	60.74	26.67	54.70	59.11	35.09	47.26	43.89
DiffTPT	60.80	**31.06**	**55.80**	58.80	37.10	48.71	45.69
TDA (Ours)	61.35	30.29	55.54	**62.58**	**38.12**	**49.58**	**46.63**
CLIP-ViT-B/16	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
Tip-Adapter	70.75	51.04	63.41	77.76	48.88	62.37	60.27
TPT	68.98	54.77	63.45	77.06	47.94	62.44	60.81
DiffTPT	70.30	55.68	**65.10**	75.00	46.80	62.28	60.52
TDA (Ours)	69.51	**60.11**	64.67	**80.24**	**50.54**	**65.01**	**63.89**

Table 1. **Results on the OOD Benchmark**. Our TDA is compared with several state-of-the-art methods designed for vision-language models: the baseline method CLIP, three train-time adaptation methods (*i.e.*, CoOp, CoCoOp, and Tip-Adapter), and two test-time adaptation methods (*i.e.*, TPT and DiffTPT). All the compared methods are built upon CLIP-ResNet-50 or CLIP-ViT-B/16 baselines. The two evaluation metrics *Average* and *OOD Average* are calculated by taking the mean accuracy across all five datasets and four OOD datasets excluding ImageNet. The results of CLIP, CoOp, CoCoOp, and TPT are obtained from the TPT paper, the results of DiffTPT are obtained from the DiffTPT paper, while the results of Tip-Adapter are reproduced using the official codes.

Experiments

Cross-Domain benchmark provides a comprehensive evaluation of the model's adaptability during test time across various class spaces.

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	Average
CLIP-ResNet-50	16.11	87.26	55.89	40.37	25.79	62.77	74.82	82.97	60.85	59.48	56.63
CoOp	15.12	86.53	55.32	37.29	26.20	61.55	75.59	87.00	58.15	59.05	56.18
CoCoOp	14.61	87.38	56.22	38.53	28.73	65.57	76.20	**88.39**	59.61	57.10	57.23
TPT	17.58	87.02	58.46	40.84	28.33	62.69	74.88	84.49	61.46	60.82	57.66
DiffTPT	17.60	86.89	**60.71**	40.72	41.04	63.53	**79.21**	83.40	**62.72**	62.67	59.85
TDA (Ours)	**17.61**	**89.70**	57.78	**43.74**	**42.11**	**68.74**	77.75	86.18	62.53	**64.18**	**61.03**
CLIP-ViT-B/16	23.22	93.55	66.11	45.04	50.42	66.99	82.86	86.92	65.63	65.16	64.59
CoOp	18.47	93.70	64.51	41.92	46.39	68.71	85.30	89.14	64.15	66.55	63.88
CoCoOp	22.29	93.79	64.90	45.45	39.23	70.85	83.97	**90.46**	66.89	68.44	64.63
TPT	24.78	94.16	66.87	**47.75**	42.44	68.98	84.67	87.79	65.50	68.04	65.10
DiffTPT	**25.60**	92.49	67.01	47.00	43.13	70.10	**87.23**	88.22	65.74	62.67	65.47
TDA (Ours)	23.91	**94.24**	**67.28**	47.40	**58.00**	**71.42**	86.14	88.63	**67.62**	**70.66**	**67.53**

Table 3. **Results on the Cross-Domain Benchmark.** Our TDA is compared with several state-of-the-art methods designed for vision-language models: the baseline method CLIP, two train-time adaptation methods (*i.e.*, CoOp and CoCoOp), and two test-time adaptation methods (*i.e.*, TPT and DiffTPT). Note that Tip-Adapter is unable to be evaluated on the Cross-Domain Benchmark as it cannot handle new classes during testing. The evaluation metric *Average* is calculated by taking the mean accuracy across all ten datasets. The results of CLIP, CoOp, CoCoOp, and TPT are obtained from the TPT paper, while the results of DiffTPT are obtained from the DiffTPT paper.

Experiments

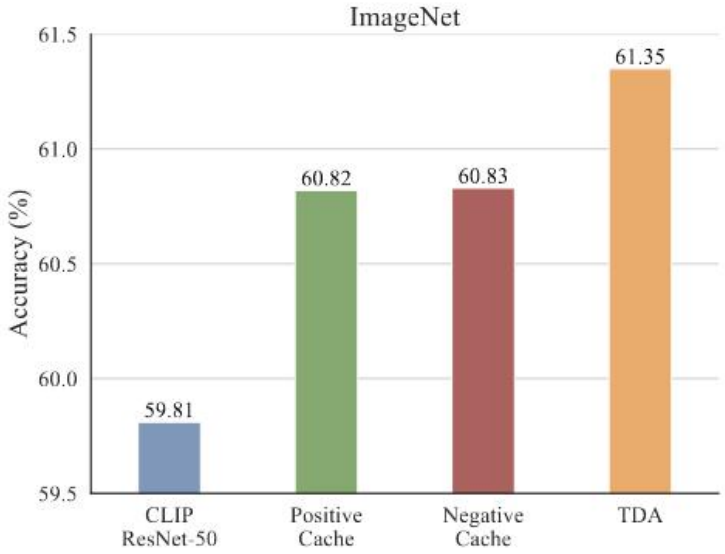


Figure 3. Ablation studies on two cache designs in TDA: *Positive Cache* and *Negative Cache*. All the models are built upon the baseline model CLIP-ResNet-50.

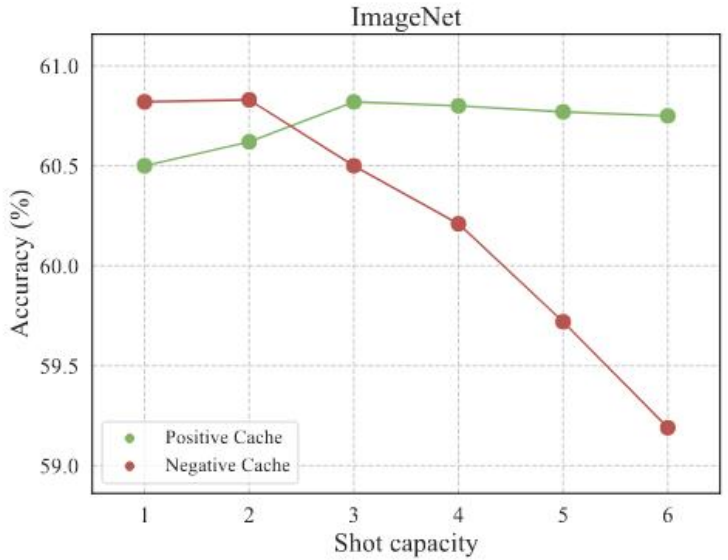


Figure 4. Parameter studies on the *Shot Capacity* in *Positive Cache* and *Negative Cache*.

Method	Testing Time	Accuracy	Gain
CLIP-ResNet-50	**12min**	59.81	0
TPT	12h 50min	60.74	+0.93
DiffTPT	34h 45min	60.80	+0.99
TDA (Ours)	**16min**	**61.35**	**+1.54**

Table 2. Comparisons of our TDA with CLIP-ResNet-50, TPT, and DiffTPT in terms of efficiency (*Testing Time*) and effectiveness (*Accuracy*). The final column shows the accuracy gain relative to the baseline CLIP. Note that the testing time of DiffTPT does not include the duration required for the image generation process with pre-trained diffusion models, which is an additional time-consuming factor during the testing phase.

WATT: Weight Average Test-Time Adaptation of CLIP

David Osowiechi* Mehrdad Noori* Gustavo A. Vargas Hakim

Moslem Yazdanpanah Ali Bahri Milad Cheraghalikhani Sahar Dastani

Farzad Beizaee Ismail Ben Ayed Christian Desrosiers

arxiv 2022

Motivation

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

However, a significant challenge remains: swiftly and effectively adapting the model to new domains in real-time while preserving its attractive zero-shot capabilities, thus obviating the need for retraining

To tackle this challenge, we investigate the impact of different text prompt templates on model adaptation. A key observation driving our approach is the varying performance outcomes yielded by different text prompt templates when used for model adaptation.

	T^0	T^1	T^2	T^3	T^4	T^5	T^6	T^7	WATT
Original	89.80	90.37	90.50	88.42	89.93	89.95	90.13	88.54	**91.05**
Gaussian Noise	60.19	61.01	61.17	58.24	58.84	58.35	59.62	61.13	**63.84**
Defocus Blur	77.23	77.07	78.00	75.98	76.39	77.45	77.08	75.59	**78.94**
Snow	76.57	77.36	77.93	75.08	77.45	77.09	77.05	75.57	**79.79**
JPEG Compression	64.65	65.36	65.24	64.16	64.18	64.36	64.78	65.32	**67.36**

Table 1: Comparison of accuracy (%) using cross-entropy (CE) on CIFAR-10 and some corruptions of CIFAR-10-C datasets on different templates (please see Table 2) and the weight average.

Methods

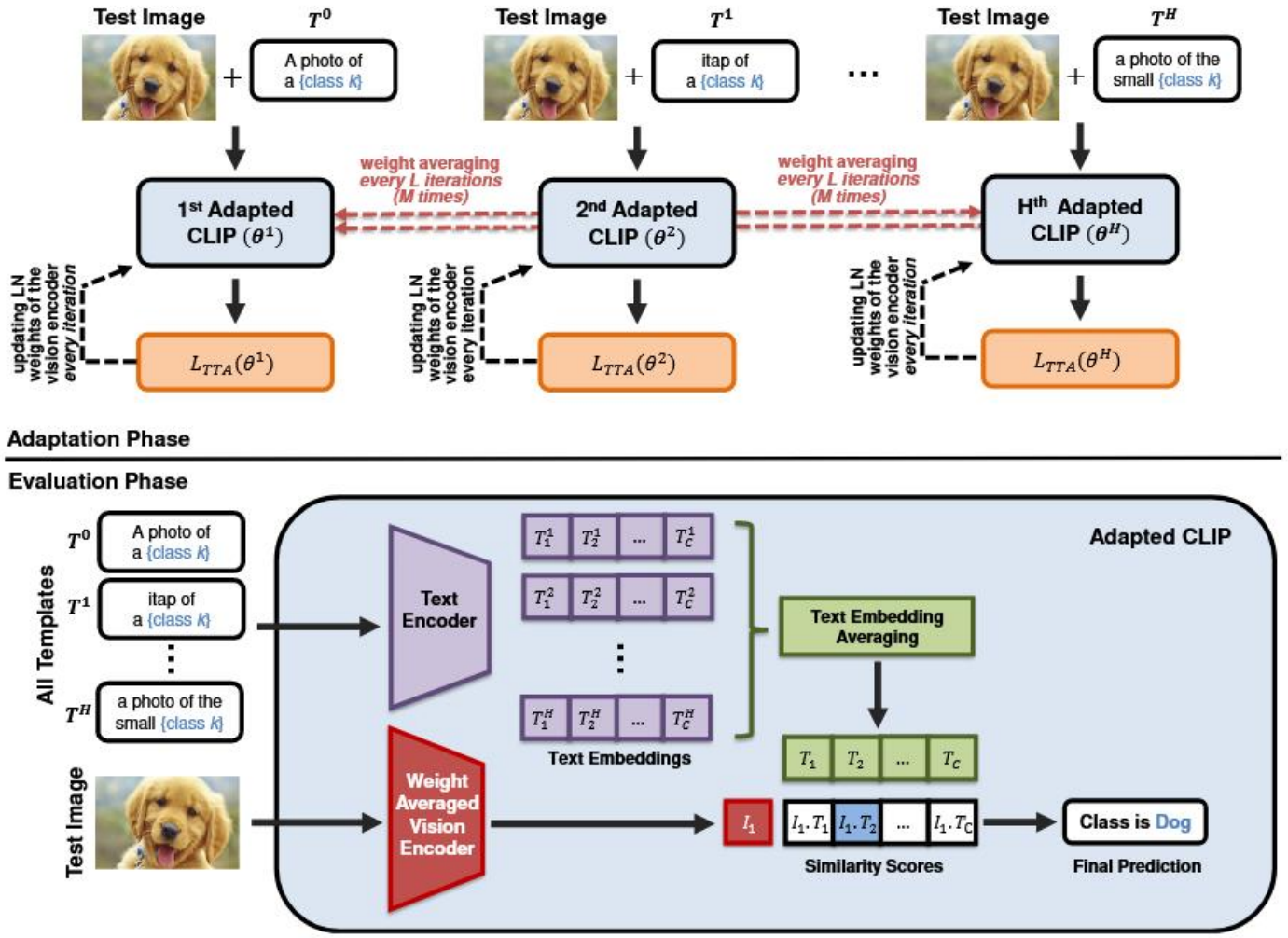
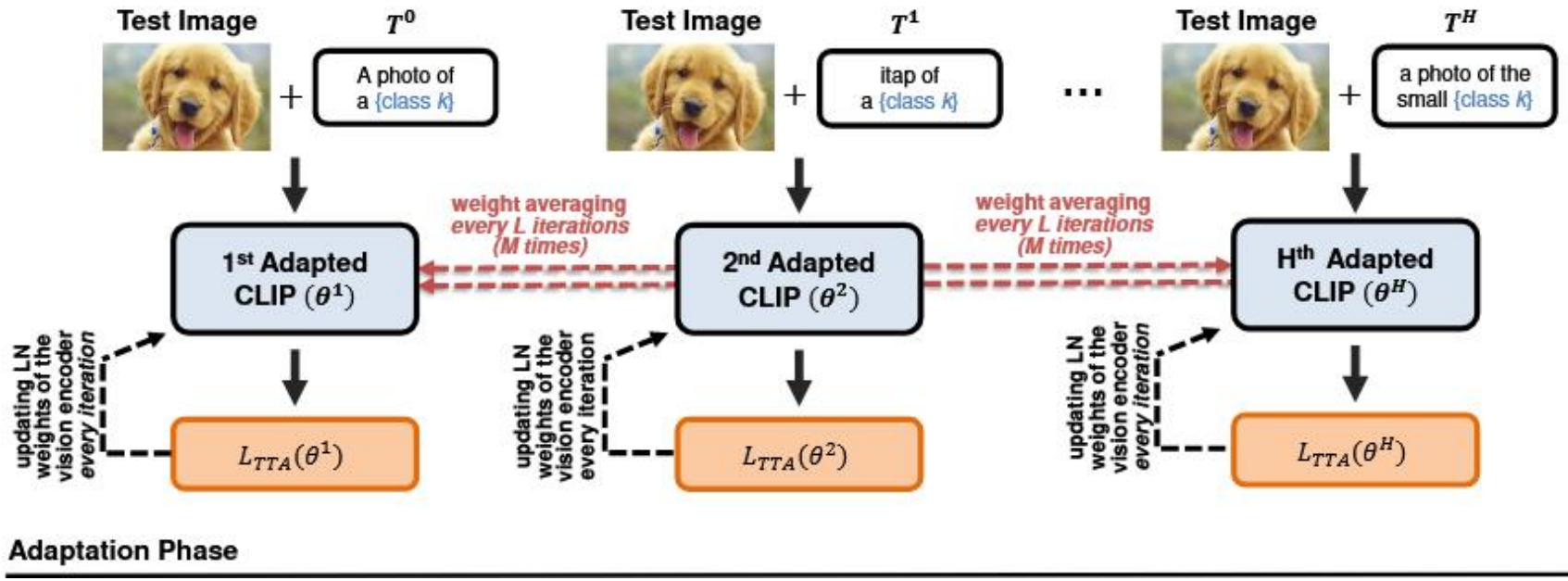


Figure 2: Overview of the proposed WATT method. In the Adaptation Phase, the model is adapted using different text templates (T^0, T^1, ..., T^H), with weight averaging performed periodically. In the Evaluation Phase, the adapted CLIP model uses averaged text embeddings from all templates and the weight averaged model to predict the class of the test image.

	Template
T^0:	"a photo of a {class k}"
T^1:	"itap of a {class k}"
T^2:	"a bad photo of the {class k}"
T^3:	"a origami {class k}"
T^4:	"a photo of the large {class k}"
T^5:	"a {class k} in a video game"
T^6:	"art of the {class k}"
T^7:	"a photo of the small {class k}"

Table 2: The different templates used during the experiments.

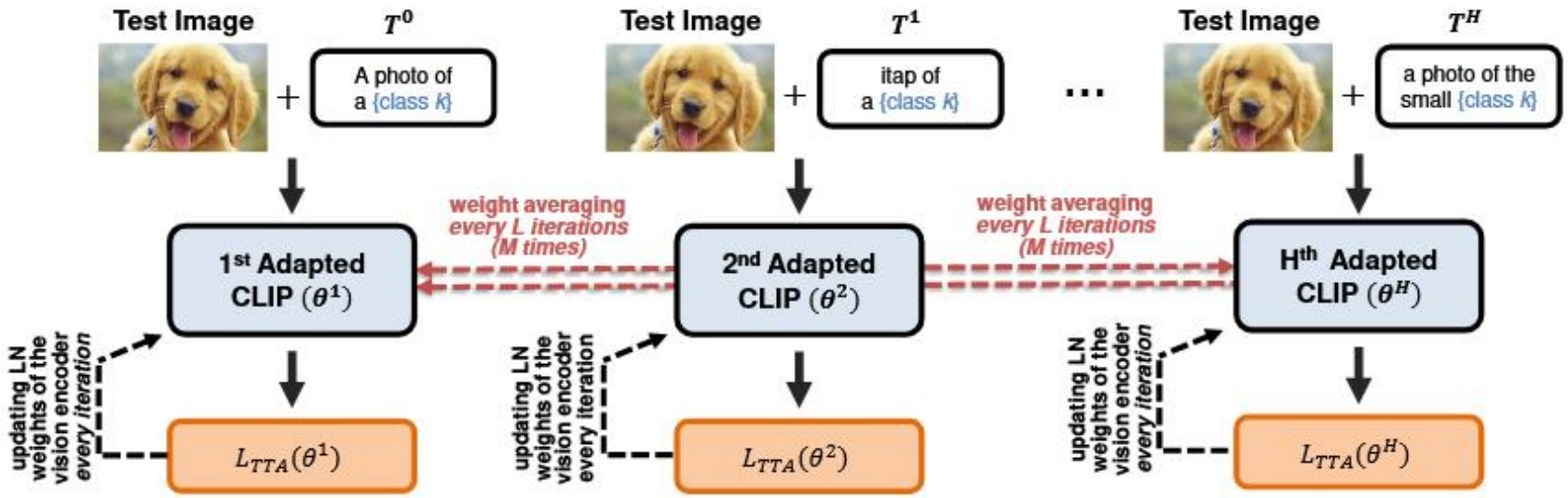
Methods



$$p_{ik} = \frac{\exp\left(\cos(\mathbf{z}_i^v, \mathbf{z}_k^t)/\tau\right)}{\sum_j \exp\left(\cos(\mathbf{z}_i^v, \mathbf{z}_j^t)/\tau\right)}, \quad \cos(\mathbf{z}, \mathbf{z}') = \frac{\mathbf{z}^\top \mathbf{z}'}{\|\mathbf{z}\|_2 \cdot \|\mathbf{z}'\|_2}, \tag{1}$$

where τ is a softmax temperature parameter set to 0.01 is this work. This prediction is then stored to be used as pseudo-labels for the model.

Methods



Adaptation Phase

Denoting the normalized visual embeddings of the samples within the test batch as $\mathbf{Z}^v \in \mathbb{R}^{B\times D}$ and the instance-specific text embeddings as $\mathbf{Z}^t \in \mathbb{R}^{B\times D}$, we compute an image-to-image similarity matrix $\mathbf{S}^v = \mathbf{Z}^v(\mathbf{Z}^v)^\top \in [-1,1]^{B\times B}$ modeling pairwise relationships in terms of image characteristics. Similarly, we construct a text-to-text similarity matrix $\mathbf{S}^t = \mathbf{Z}^t(\mathbf{Z}^t)^\top \in [-1,1]^{B\times B}$, capturing the semantic relationships among text embeddings within the batch. Utilizing the computed pairwise similarity matrices, we generate pseudo-labels $\mathbf{Q} = \text{softmax}\big((\mathbf{S}^v + \mathbf{S}^t)/2\tau\big) \in [0,1]^{B\times B}$ which are used with cross-entropy in our transductive TTA loss:

$$\mathcal{L}_{\text{TTA}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{B} q_{ij}\log p_{ij}. \tag{2}$$

Methods

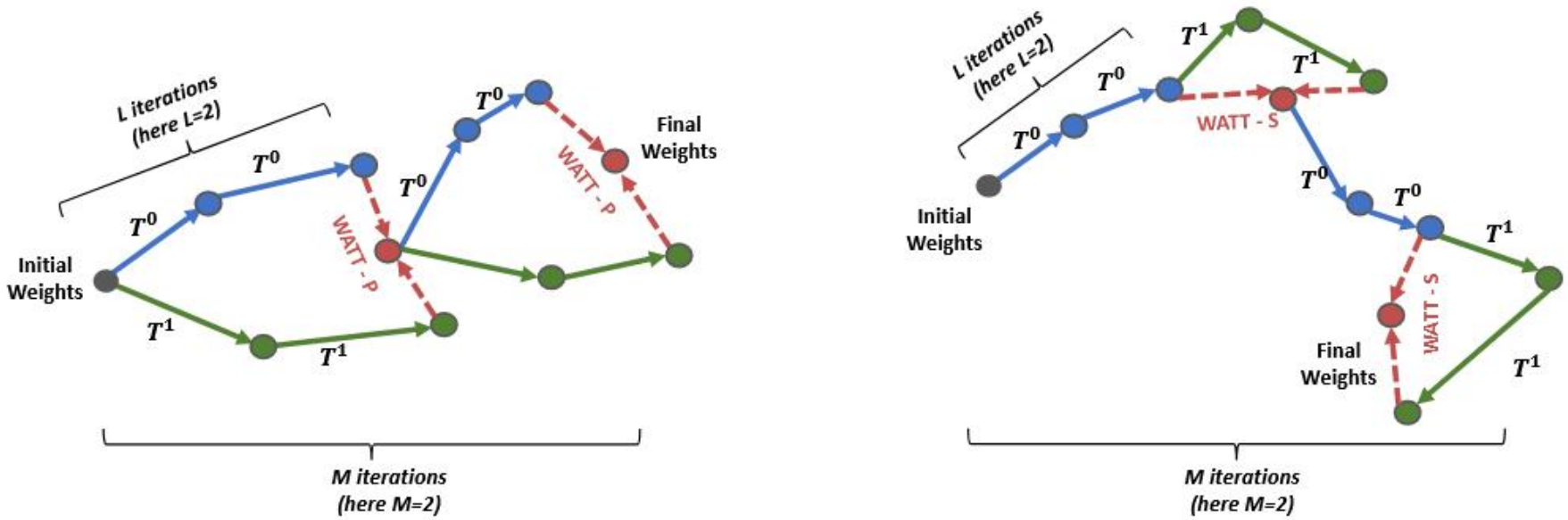


Figure 3: Visual comparison of the Parallel **(left)** and Sequential **(right)** approaches for multi-template weight averaging during adaptation.

Parallel MTWA. This approach optimizes the TTA loss in (2) separately for H different models, each utilizing a distinct template. Starting from the same visual encoder parameters θ, these models are updated in parallel for L iterations, resulting in updated parameters θ'_h, with $h \in \{1, \ldots, H\}$. The parameters are reset after each update, enabling each model to restart the adaptation from the same initial point. Subsequently, we aggregate the weights obtained from these H models by computing their average: $\theta_{\text{avg}} = \frac{1}{H} \sum_{h=1}^{H} \theta'_h$. We repeat this step M times, and denote the overall process as "(after L iter) $\times$ M".

Sequential MTWA. Our Sequential MTWA approach is inspired from the work of [22], where the averaging of weights across various stages of the training process is employed to mitigate variance and enhance generalization capabilities. Instead of resetting parameters for each model, we update

Experiments

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

Comparison of the template used during testing.

Dataset	single_temp	text_avg
CIFAR-10	90.87 ±0.10	**91.08** ±0.06
CIFAR-10.1	86.80 ±0.19	**86.85** ±0.18
CIFAR-10-C	72.08	**72.66**
CIFAR-100	69.79 ±0.20	**70.30** ±0.11
CIFAR-100-C	41.79	**42.24**

Table 3: Accuracy (%) with different text ensembles at test time.

Comparison of the number of templates

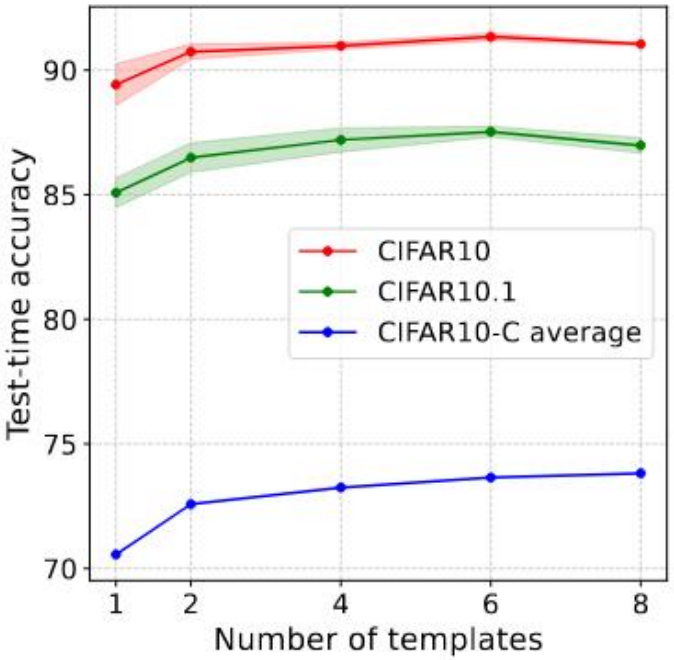


Figure 4: Evolution of the accuracy for different numbers of random template on 5 test-time runs.

Experiments

Text Averaging vs Output Averaging vs Weight Averaging

Dataset	Text avg.	Output avg.	Weight avg. (ours)		
			(after 10 iter)×1	(after 1 iter)×10	(after 2 iter)×5
CIFAR-10	90.58 ±0.03	90.90 ±0.03	91.08 ±0.06	**91.39 ±0.14**	91.05 ±0.06
CIFAR-10.1	85.78 ±0.25	86.77 ±0.08	86.85 ±0.18	**88.02 ±0.18**	86.98 ±0.31
CIFAR-10-C	71.41	72.60	72.66	73.66	**73.82**
CIFAR-100	69.46 ±0.13	70.32 ±0.1	70.3 ±0.11	**70.85 ±0.08**	70.74 ±0.20
CIFAR-100-C	41.37	42.68	42.24	45.32	**45.57**

Table 5: Accuracy (%) obtained with different averaging strategies.

Experiments

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

1、Performance Evaluation in the Presence of Natural or No Domain Shift
2、Performance Evaluation in the Presence of Common Corruptions

Dataset		CLIP	TENT	TPT (BS=32)	CLIPArTT	WATT-P	WATT-S
CIFAR-10		88.74	**91.69** ±0.10	88.06 ±0.06	90.04 ±0.13	91.41 ±0.17	91.05 ±0.06
CIFAR-10.1		83.25	87.60 ±0.45	81.80 ±0.27	86.35 ±0.27	**87.78** ±0.05	86.98 ±0.31
CIFAR-10-C		59.22	67.56	56.80	71.17	72.83	**73.82**
CIFAR-100		61.68	69.74 ±0.16	63.78 ±0.28	69.79 ±0.04	70.38 ±0.14	**70.74** ±0.20
CIFAR-100-C	Gaussian Noise	14.80	14.38 ±0.14	14.03 ±0.10	25.32 ±0.14	31.28 ±0.03	**32.07** ±0.23
	Shot noise	16.03	17.34 ±0.27	15.25 ±0.17	27.90 ±0.05	33.44 ±0.11	**34.36** ±0.11
	Impulse Noise	13.85	10.03 ±0.13	13.01 ±0.13	25.62 ±0.09	29.40 ±0.11	**30.33** ±0.03
	Defocus blur	36.74	49.05 ±0.07	37.60 ±0.17	49.88 ±0.23	52.32 ±0.28	**52.99** ±0.16
	Glass blur	14.19	3.71 ±0.07	16.41 ±0.02	27.89 ±0.03	31.20 ±0.12	**32.15** ±0.30
	Motion blur	36.14	46.62 ±0.27	37.52 ±0.23	47.93 ±0.14	49.72 ±0.15	**50.53** ±0.12
	Zoom blur	40.24	51.84 ±0.15	42.99 ±0.11	52.70 ±0.06	54.72 ±0.04	**55.30** ±0.22
	Snow	38.95	46.71 ±0.21	42.35 ±0.13	49.72 ±0.01	51.79 ±0.04	**52.77** ±0.15
	Frost	40.56	44.90 ±0.27	43.31 ±0.14	49.63 ±0.12	53.04 ±0.08	**53.79** ±0.31
	Fog	38.00	47.31 ±0.04	38.81 ±0.17	48.77 ±0.04	50.78 ±0.24	**51.49** ±0.21
	Brightness	48.18	60.58 ±0.18	50.23 ±0.11	61.27 ±0.08	62.65 ±0.25	**63.57** ±0.21
	Contrast	29.53	45.90 ±0.11	28.09 ±0.09	48.55 ±0.24	51.34 ±0.10	**52.76** ±0.27
	Elastic transform	26.33	33.09 ±0.08	28.12 ±0.15	37.45 ±0.08	39.97 ±0.06	**40.90** ±0.43
	Pixelate	21.98	26.47 ±0.09	20.43 ±0.14	33.88 ±0.14	39.59 ±0.09	**40.97** ±0.16
	JPEG compression	25.91	29.89 ±0.07	28.82 ±0.09	36.07 ±0.32	38.99 ±0.16	**39.59** ±0.08
	Mean	29.43	35.19	30.46	41.51	44.68	**45.57**

Table 6: Accuracy (%) on CIFAR-10, CIFAR-10.1, CIFAR-10-C, CIFAR-100 and CIFAR-100-C datasets. WATT-P refers to our method with Parallel MTWA and WATT-S to the Sequential MTWA variant of WATT.

Experiments

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

3、Performance analysis under simulated and video shifts
4、Performance analysis under texture and style shifts

Dataset	Domain	CLIP	TENT	TPT	CLIPArTT	WATT-P	WATT-S
VisDA-C	3D (trainset)	84.43	84.86 ±0.01	79.35 ±0.04	85.09 ±0.01	**85.42 ±0.03**	85.36 ±0.01
	YT (valset)	84.45	84.68 ±0.01	83.57 ±0.04	84.40 ±0.01	84.57 ±0.00	**84.69 ±0.01**
	Mean	84.44	84.77	81.46	84.75	85.00	**85.03**
OfficeHome	Art	73.75	74.03 ±0.27	**75.76 ±0.27**	73.84 ±0.20	75.65 ±0.27	**75.76 ±0.39**
	Clipart	63.33	63.42 ±0.04	63.08 ±0.31	63.54 ±0.06	**66.23 ±0.13**	65.77 ±0.11
	Product	85.32	**85.51 ±0.08**	84.07 ±0.28	85.23 ±0.16	85.41 ±0.09	85.41 ±0.01
	Real World	87.71	87.74 ±0.05	85.89 ±0.33	87.61 ±0.05	88.22 ±0.15	**88.37 ±0.05**
	Mean	77.53	77.68	77.20	77.56	**78.88**	78.83
PACS	Art	96.34	**96.65 ±0.05**	95.52 ±0.20	96.57 ±0.09	96.31 ±0.00	96.39 ±0.00
	Cartoon	96.08	96.22 ±0.05	94.77 ±0.20	96.00 ±0.02	96.52 ±0.02	**96.62 ±0.02**
	Photo	99.34	99.40 ±0.00	99.42 ±0.06	99.28 ±0.00	99.48 ±0.03	**99.52 ±0.00**
	Sketch	82.85	82.96 ±0.12	83.22 ±0.14	83.93 ±0.14	**86.92 ±0.04**	86.65 ±0.12
	Mean	93.65	93.81	93.23	93.95	**94.81**	94.80
VLCS	Caltech101	**99.51**	**99.51 ±0.00**	99.36 ±0.06	**99.51 ±0.00**	99.43 ±0.00	**99.51 ±0.00**
	LabelMe	68.15	67.89 ±0.13	54.88 ±0.12	67.96 ±0.04	66.67 ±0.21	**68.49 ±0.12**
	SUN09	68.85	69.27 ±0.04	67.30 ±0.49	68.68 ±0.09	72.61 ±0.15	**73.13 ±0.17**
	VOC2007	84.13	**84.42 ±0.15**	76.74 ±0.28	84.09 ±0.02	82.30 ±0.16	83.41 ±0.17
	Mean	80.16	80.27	74.57	80.06	80.25	**81.14**

Table 7: Accuracy (%) on different domains of VisDA-C, OfficeHome, PACS and VLCS datasets.

Thanks