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

# Efficient Test-Time Adaptation of Vision-Language Models

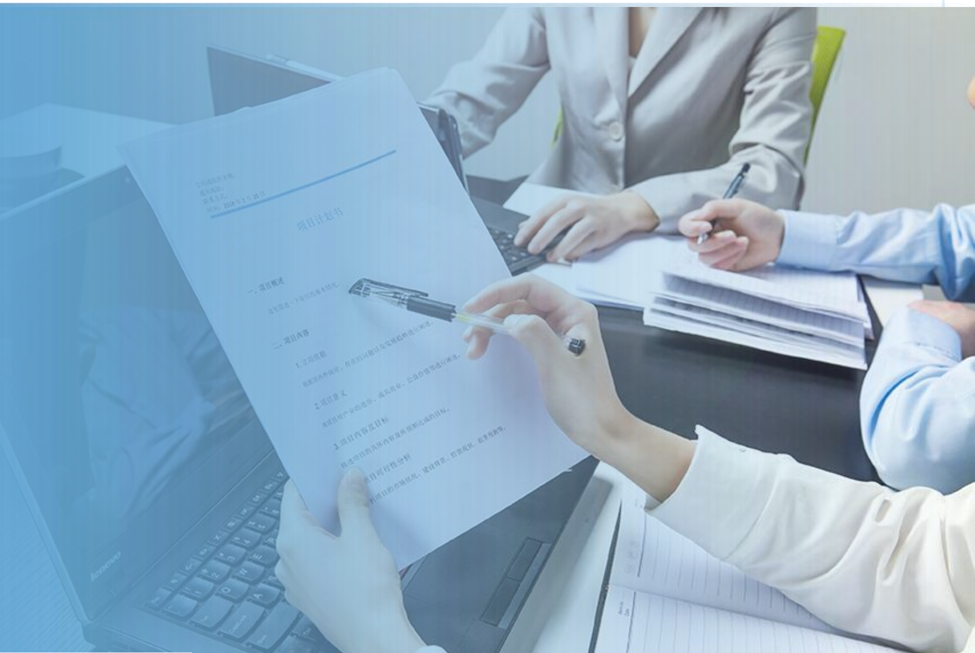
Adilbek Karmanov[1]* Dayan Guan[2]* Shijian Lu[1,2]† Abdulmotaleb El Saddik[1,3] Eric Xing[1,4]
[1]Mohamed bin Zayed University of Artificial Intelligence [2]Nanyang Technological University
[3]University of Ottawa [4]Carnegie Mellon University

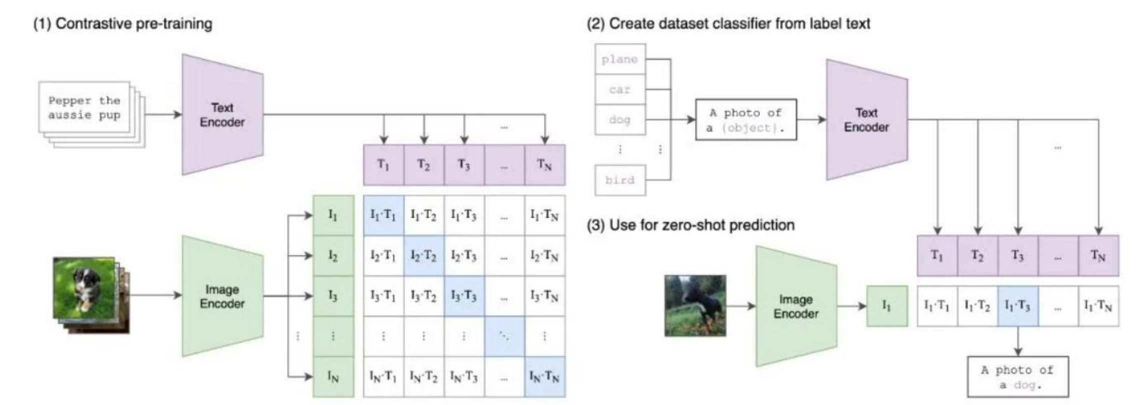
汇报人：蒋明忠

时间：2025.05

# Background

Recent advances in vision-language models [19, 31, 43] have opened a new door for integrating human language into various computer vision tasks. Take CLIP [31] as an example. It can enable zero-shot image classification by leveraging a shared embedding space that is learnt from web-scale image-text pairs. Within this shared space, images can be directly recognized by matching their features with the text embeddings of CLIP classes. At the other end, CLIP often faces challenges while handling various specific downstream images, especially when the downstream images have clear domain and distribution shifts as compared with the CLIP training images.

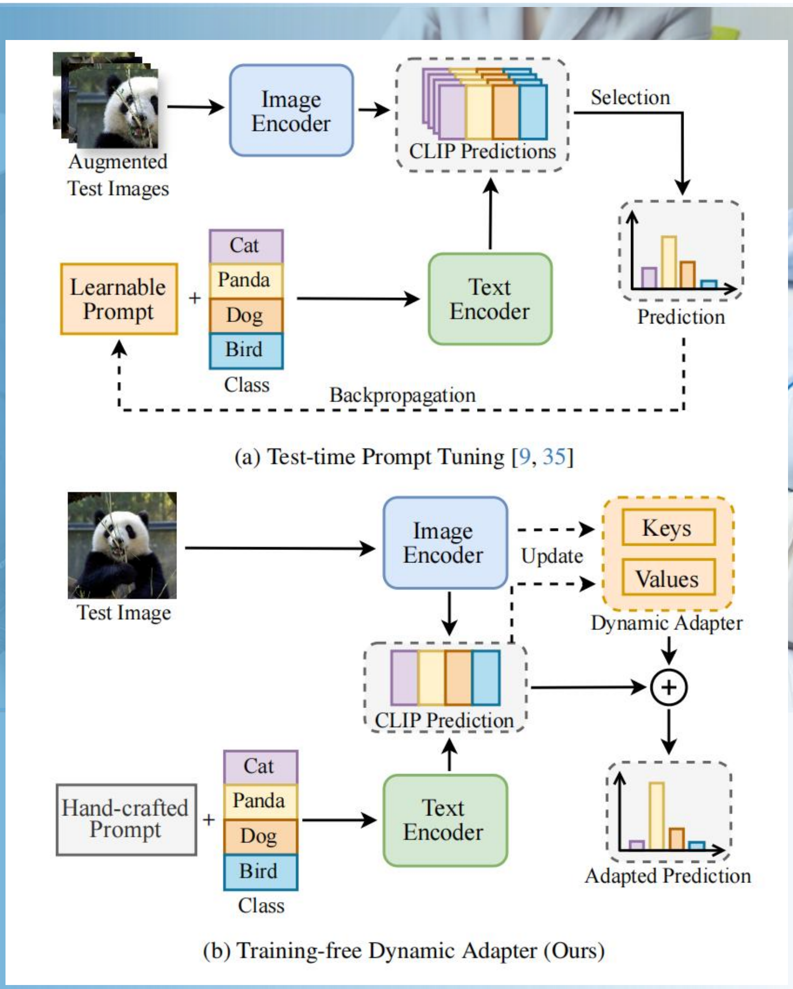



(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
$T_1$ $T_2$ $T_3$ ... $T_N$
$I_1$ $I_2$ $I_3$ ... $I_N$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
$T_1$ $T_2$ $T_3$ ... $T_N$
$I_1$
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

# Background

(1) Test-time adaptation (TTA) is a new approach designed to handle domain shift by allowing models to adapt to new environments during inference. Although promising, its application in vision-language models is still underexplored. TPT and DiffTPT address this gap by learning domain-specific prompts from test data to guide models like CLIP. They optimize a prompt for each test sample using augmented views to reduce prediction uncertainty. However, the high computational cost of this prompt optimization limits their practical use.

(2) We propose a training-free Dynamic Adapter (TDA) for efficient and effective test-time adaptation of vision-language models like CLIP, without requiring backpropagation. As shown in Fig. 1(b), TDA maintains a lightweight key-value cache, storing CLIP image features as keys and pseudo labels as values. It is both effective, by progressively refining pseudo labels with low-entropy predictions, and efficient, as it uses simple matrix operations and requires no parameter updates.

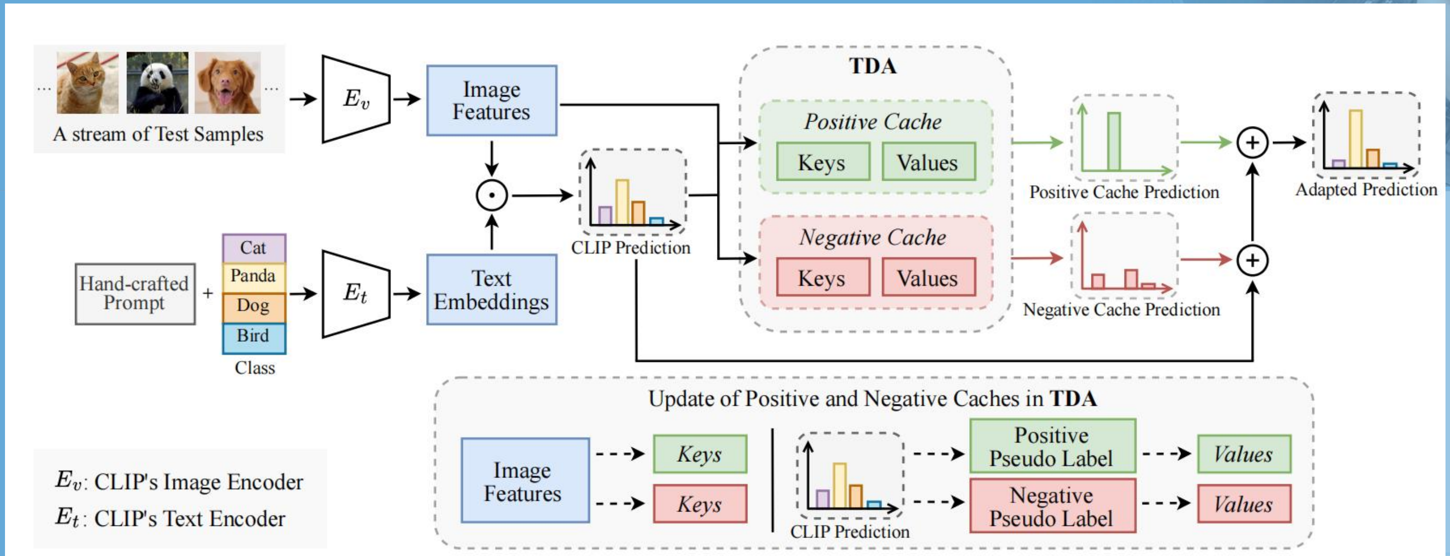


(a) Test-time Prompt Tuning [9, 35]

(b) Training-free Dynamic Adapter (Ours)

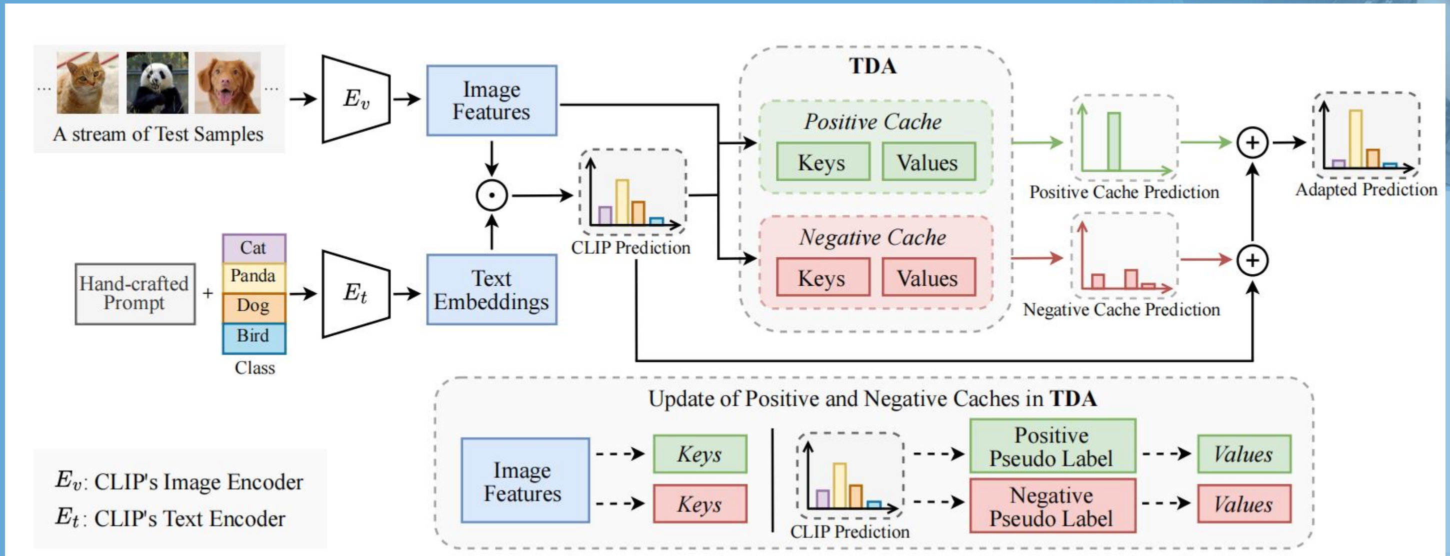
# Method

We propose a training-free dynamic adapter (TDA) for efficient test-time adaptation of pre-trained vision-language models like CLIP. TDA consists of two lightweight key-value caches: one for positive learning and the other for negative learning. The positive cache stores high-confidence predictions to improve accuracy, while the negative cache handles noisy pseudo labels by identifying class absence. Combining both caches, TDA achieves superior performance in speed and accuracy during test-time adaptation.



# Method

Positive Cache: TDA uses a key-value cache in the form of a dynamic queue to store high-quality pseudo labels (values) and their corresponding features (keys) during test-time adaptation. The cache starts empty and grows as more confident predictions (with lower entropy) are added. Unlike a fixed-size FIFO queue, this queue dynamically expands and acts like a priority queue, prioritizing entries by entropy. Each class maintains its own separate queue to ensure proper structure and ordering.



# Method

TDA builds a positive cache by generating pseudo labels for test images using a pre-trained CLIP model. For each test sample, it applies softmax to the prediction to get a one-hot pseudo label and decides whether to add it to the cache based on two conditions:

(1) If the current class has fewer cached pairs than the max capacity, the pseudo label and its image features are added directly.

(2) If the class cache is full, the new pair replaces the one with the highest entropy (i.e., the least confident prediction), but only if the new prediction has lower entropy.

This ensures that TDA selectively keeps confident (low-entropy) predictions and maintains a balanced cache size across classes.

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

# Method

Negative Cache: Similar to the positive cache in our TDA, the negative cache is also a dynamic queue structure with negative keys and negative values denoted as  and  , respectively. It aims to gather CLIP-generated image features to and the corresponding negative pseudo labels to  . Unlike the pseudo labels in the positive cache, the negative pseudo labels are obtained by applying negative mask on the class probabilities as.

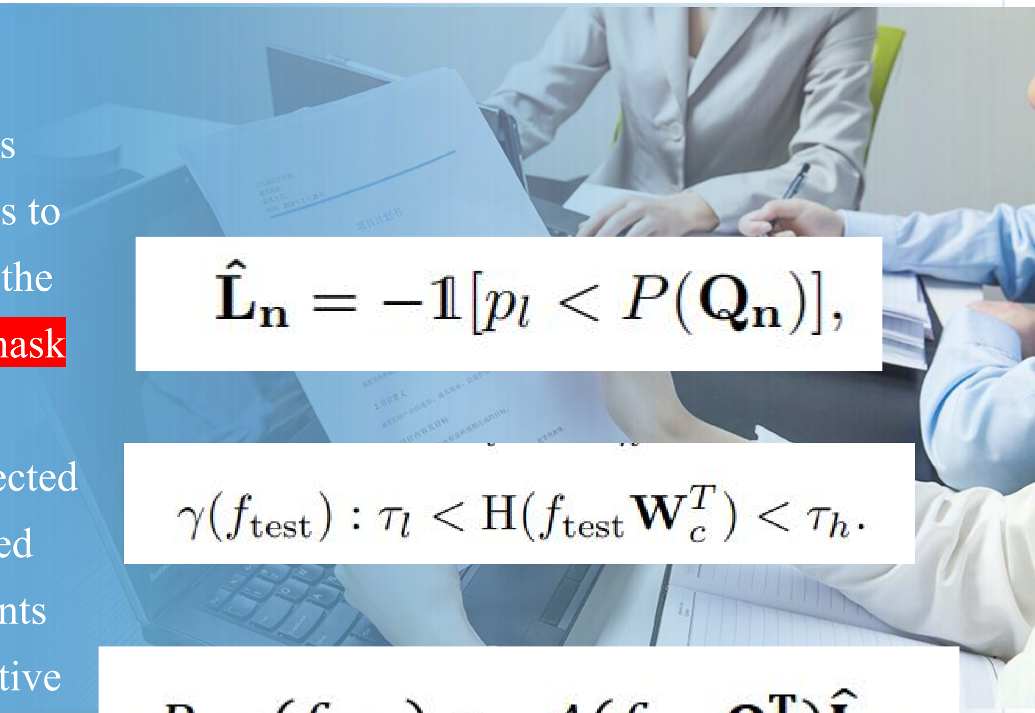
$$\hat{\mathbf{L}}_{\mathbf{n}} = -\mathbb{1}[p_l < P(\mathbf{Q}_{\mathbf{n}})],$$

In negative pseudo labeling, higher probabilities than a threshold are selected as negative pseudo labels from uncertain predictions, with uncertainty measured by the entropy of predictions. A negative pseudo label is a vector where elements greater than the threshold are set to -1, and others to 0. Unlike traditional negative learning methods, TDA selects negative pseudo labels from uncertain predictions to avoid bias towards certain predictions. When constructing the negative cache, a testing feature will be included if its prediction entropy falls within a specified interval.

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

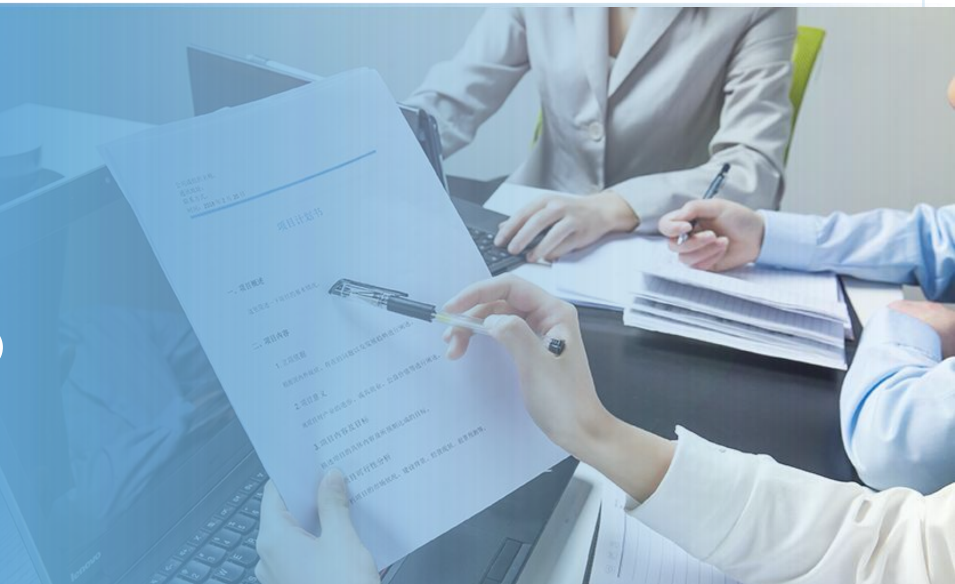
$$P_{\text{neg}}(f_{\text{test}}) =- A(f_{\text{test}}\mathbf{Q}_{\mathbf{n}}^{\mathbf{T}})\hat{\mathbf{L}}_{\mathbf{n}},$$

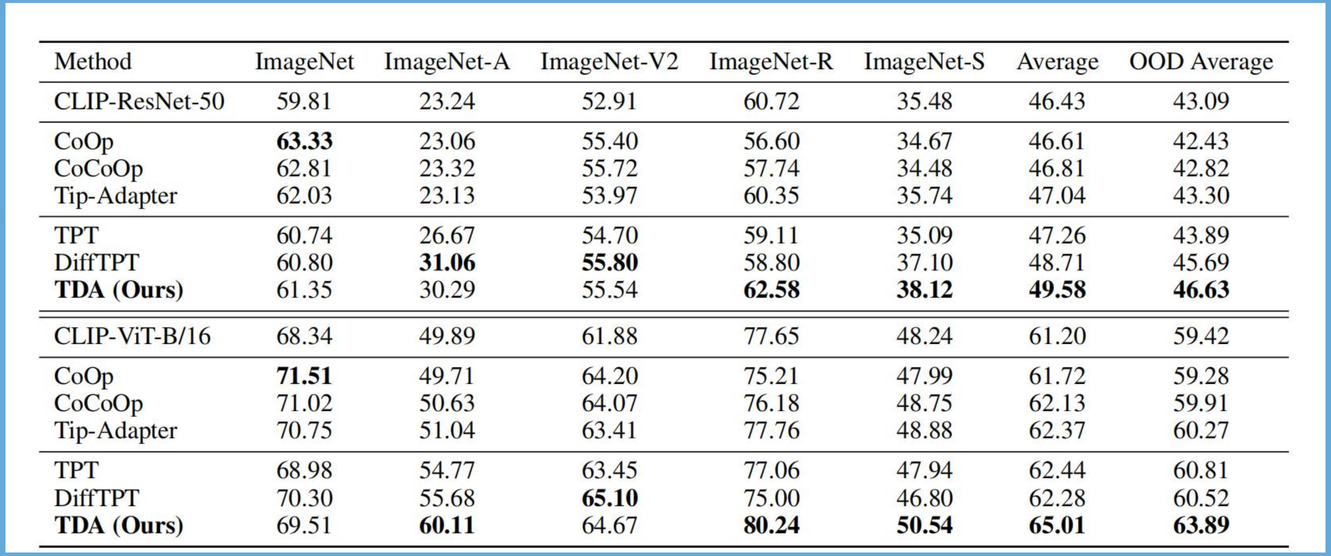
$$P_{\text{TDA}}(f_{\text{test}}) = f_{\text{test}}\mathbf{W}_c^T + P_{\text{pos}}(f_{\text{test}}) + P_{\text{neg}}(f_{\text{test}})$$



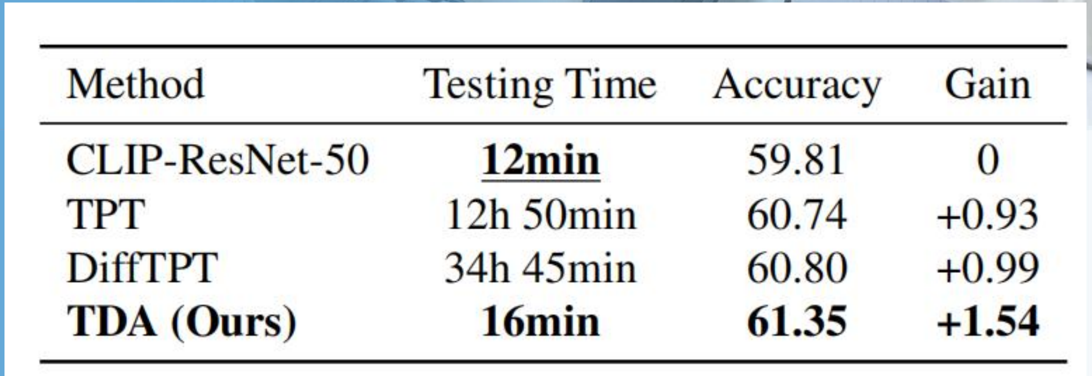
# Experiments

Benchmarks: We evaluate our method on two benchmarks: an out-of-distribution (OOD) benchmark and a cross-domain benchmark, following [35]. The OOD benchmark includes four ImageNet variants (ImageNet-A, V2, R, S) to assess model robustness on unseen data. The cross-domain benchmark tests generalization across 10 diverse datasets (e.g., Aircraft, Caltech101, EuroSAT, etc.) to evaluate adaptability across different domains and class spaces.





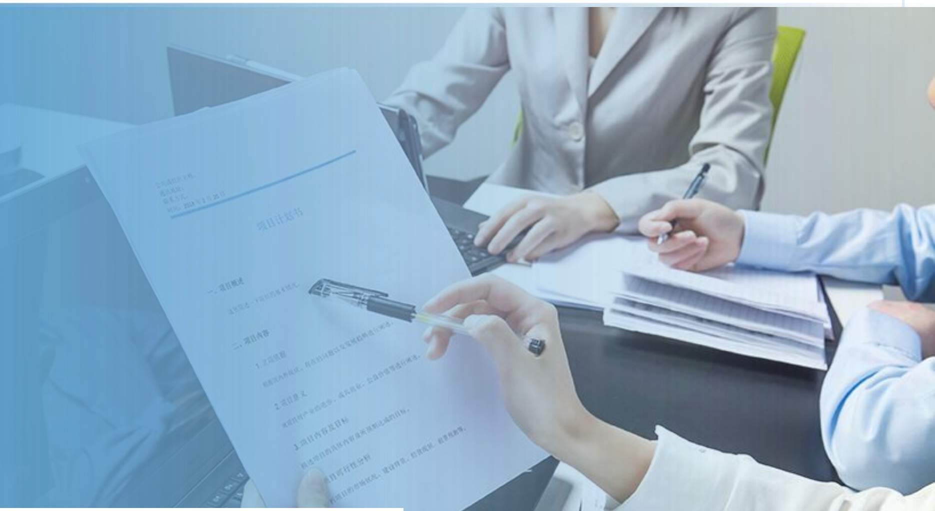
| Method | ImageNet | ImageNet-A | ImageNet-V2 | ImageNet-R | ImageNet-S | Average | OOD Average |
|---|---|---|---|---|---|---|---|
| CLIP-ResNet-50 | 59.81 | 23.24 | 52.91 | 60.72 | 35.48 | 46.43 | 43.09 |
| CoOp | **63.33** | 23.06 | 55.40 | 56.60 | 34.67 | 46.61 | 42.43 |
| CoCoOp | 62.81 | 23.32 | 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** |

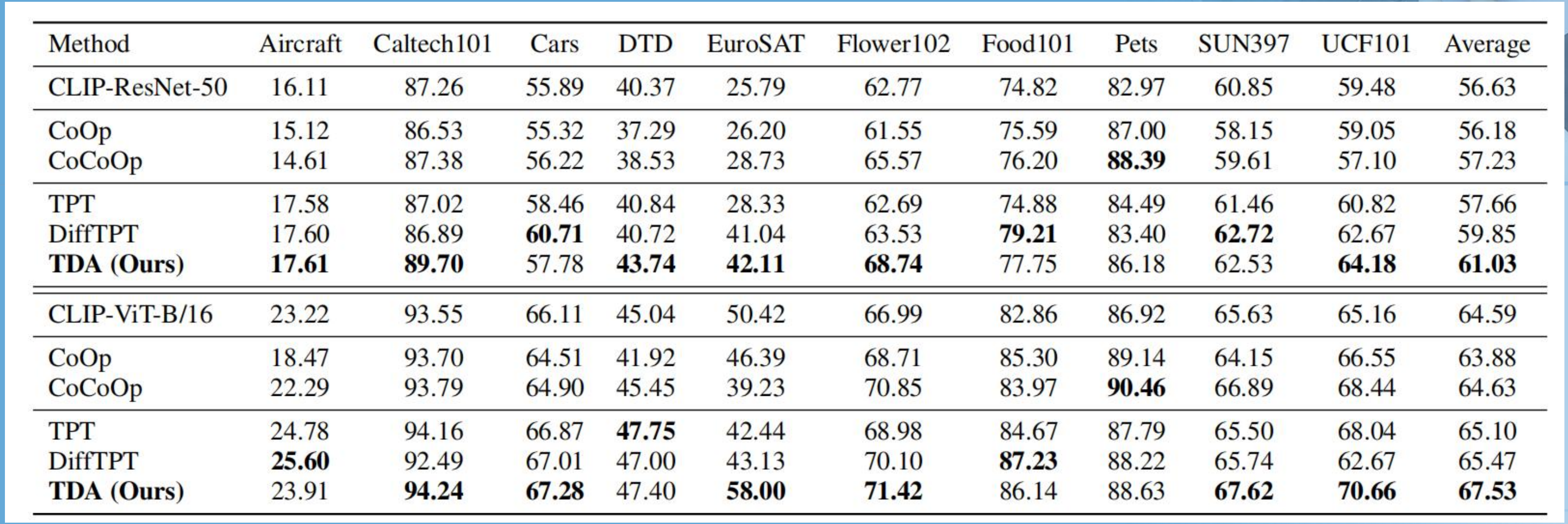


| Method | Testing Time | Accuracy | Gain |
|---|---|---|---|
| CLIP-ResNet-50 | **12min** | 59.81 | 0 |
| TPT | 12h 50min | 60.74 | +0.93 |
| DiffTPT | 34h 45min | 60.80 | +0.99 |
| **TDA (Ours)** | **16min** | **61.35** | **+1.54** |

# Experiments

Cross-Domain Benchmark Results. We compare TDA with top methods on the cross-domain benchmark. As shown in Table 3, TDA outperforms both TPT and DiffTPT. Using CLIP-ResNet-50 and CLIP-ViT-B/16, TDA improves average accuracy over TPT by 3.37% and 2.43%, respectively. TDA also shows clear gains over DiffTPT, confirming its strong test-time adaptability across diverse datasets. This is especially valuable for models like CLIP that aim to classify new classes without extra training.



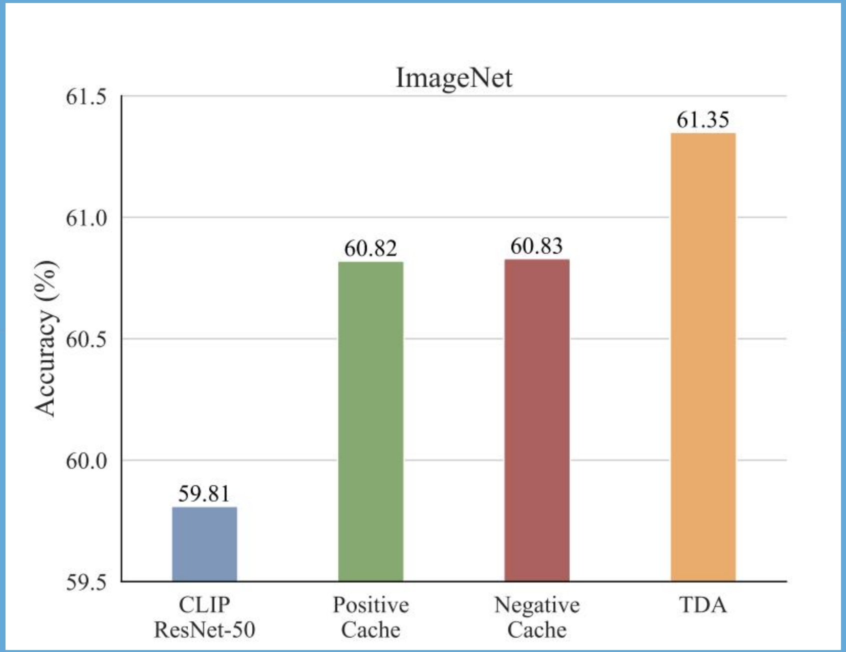


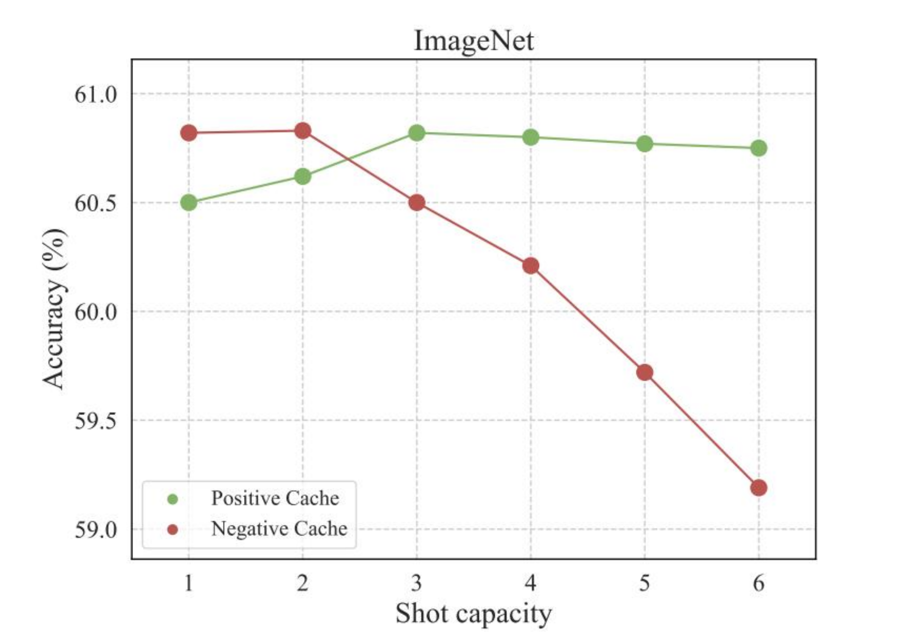
| Method | Aircraft | Caltech101 | Cars | DTD | EuroSAT | Flower102 | Food101 | Pets | SUN397 | UCF101 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CLIP-ResNet-50 | 16.11 | 87.26 | 55.89 | 40.37 | 25.79 | 62.77 | 74.82 | 82.97 | 60.85 | 59.48 | 56.63 |
| CoOp | 15.12 | 86.53 | 55.32 | 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** |

# Experiments

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.

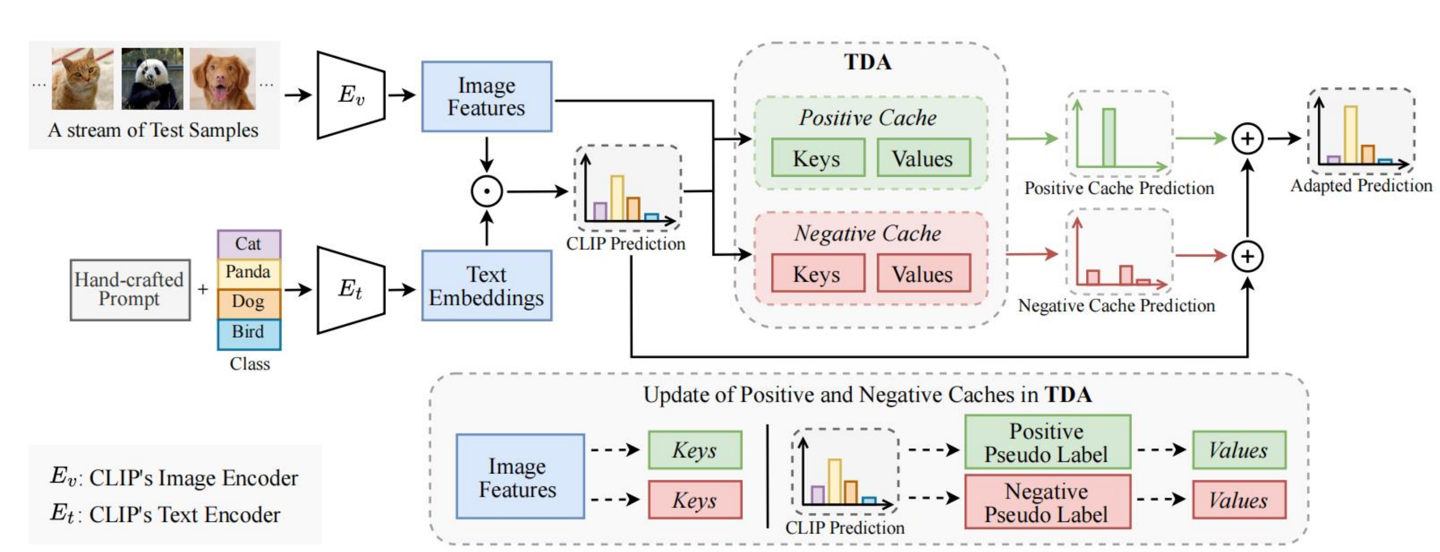
Parameter studies on the Shot Capacity in Positive Cache and Negative Cache.





# Experiments

We propose TDA, a dynamic adapter for efficient test-time adaptation of vision-language models. TDA uses a key-value cache to store test features and pseudo labels, enabling progressive adaptation. It also introduces a negative cache to reduce the impact of noisy predictions by excluding uncertain classes. Experiments on two benchmarks show that TDA outperforms existing methods while being more efficient, offering a practical and effective solution for adapting vision-language models at test time.



# *Thanks*