

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}

¹Mohamed bin Zayed University of Artificial Intelligence ²Nanyang Technological University
³University of Ottawa ⁴Carnegie Mellon University

Abstract

Test-time adaptation with pre-trained vision-language models has attracted increasing attention for tackling distribution shifts during the test time. Though prior studies have achieved very promising performance, they involve intensive computation which is severely unaligned with test-time adaptation. We design TDA, a training-free dynamic adapter that enables effective and efficient test-time adaptation with vision-language models. TDA works with a lightweight key-value cache that maintains a dynamic queue with few-shot pseudo labels as values and the corresponding test-sample features as keys. Leveraging the key-value cache, TDA allows adapting to test data gradually via progressive pseudo label refinement which is super-efficient without incurring any backpropagation. In addition, we introduce negative pseudo labeling that alleviates the adverse impact of pseudo label noises by assigning pseudo labels to certain negative classes when the model is uncertain about its pseudo label predictions. Extensive experiments over two benchmarks demonstrate TDA’s superior effectiveness and efficiency as compared with the state-of-the-art. The code has been released in <https://kdiaaa.github.io/tda/>.

1. Introduction

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

*Equal Contribution.

†Corresponding Author.

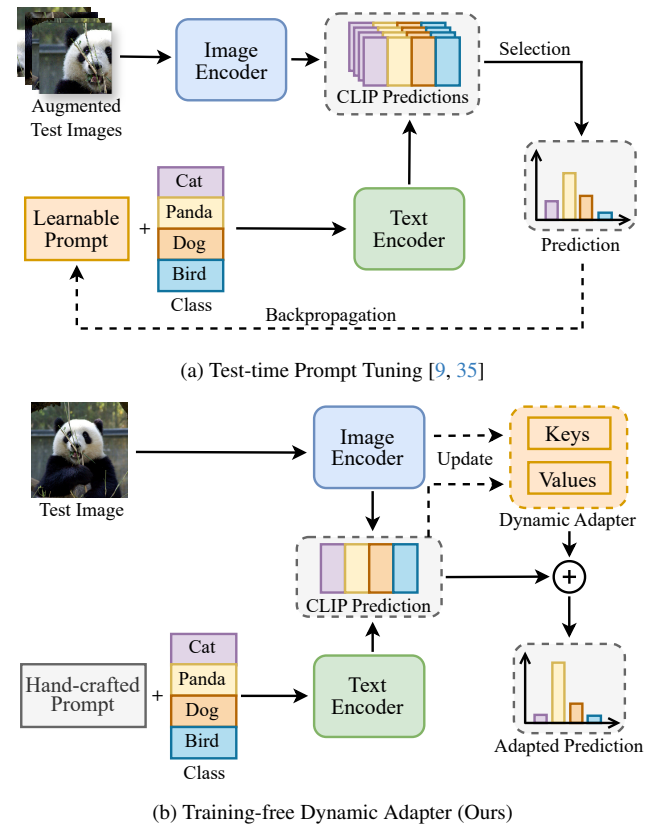


Figure 1. Comparison of our proposed *Training-free Dynamic Adapter* (TDA) with *Test-time Prompt Tuning* TPT [35] and its enhancement DiffTPT [9]: both TPT and DiffTPT require significant computational resources to optimize the *learnable prompt* via backpropagation; TDA is a dynamic cache that is training-free without any backpropagation, making it efficient for test-time adaptation in various real-world scenarios.

with the CLIP training images.

Several recent studies [2, 3, 40] introduce a new paradigm called test-time adaptation for mitigating the domain shift. The idea of test-time adaptation is well aligned with real-world scenarios where a model needs to adapt to new environments quickly. Despite its great research and

application values, test-time adaptation of vision-language models has been largely neglected. Recently, Test-time Prompt Tuning, as introduced in TPT [35] and its enhancement DiffTPT [9], attempts to adapt vision-language models by learning domain-specific prompts from test data. As illustrated in Fig. 1 (a), both TPT and DiffTPT train a learnable prompt for each test sample by feeding its augmented views into the CLIP model for generating predictions and minimizing the marginal entropy of confident predictions. Despite its decent performance, the prompt optimization in both TPT [50] and DiffTPT [9] is computationally intensive which hinders its applications in various real-world scenarios.

We design a training-free dynamic adapter (TDA) that allows efficient and effective test-time adaptation of vision-language models without requiring any backpropagation in test time. As Fig. 1 (b) shows, TDA constructs a lightweight *Dynamic Adapter* that keeps a dynamic queue with the pseudo labels of a stream of test samples as values and the corresponding CLIP-extracted features as keys. TDA has two desirable features that make its test-time adaptation highly applicable in real-world scenarios. First, TDA is highly effective, as it improves the quality of pseudo labels via progressive incorporation of test predictions of lower entropy [11, 40]. Second, TDA is very efficient as the key-value cache is non-parametric and requires no backpropagation during testing. Beyond that, TDA cache is lightweight due to the few-shot setup and it can be computed with simple matrix multiplications [12, 21, 29, 47].

Note that the performance of TDA depends heavily on the pseudo labels of unlabelled test samples which are often noisy with prediction errors. Inspired by the idea of negative learning [22, 23, 33], we introduce negative pseudo labeling to reduce the impact of noisy estimated labels. Traditional pseudo labeling methods identify the presence of particular classes in unlabeled data, which may result in erroneous pseudo labels being assigned when comparable high probabilities are observed. In contrast, our designed negative pseudo labeling determines the absence of certain classes and can provide more accurate pseudo labels as the probabilities of these complementary classes are very low. Concretely, we construct an additional TDA cache that stores negative pseudo labels to complement the positive TDA cache. By combining positive and negative caches, TDA is more tolerant to noisy pseudo labels and can better generalize to testing data. Extensive experiments over two widely adopted test-time adaptation benchmarks show that TDA outperforms the state-of-the-art by large margins while significantly reducing the testing time from over 12 hours to 16 minutes on the ImageNet dataset.

In summary, the contributions of this work are threefold. *First*, we design a training-free dynamic adapter (TDA) that can achieve test-time adaptation of vision-language models

efficiently and effectively. To the best of our knowledge, this is the first work that investigates the efficiency issue of test-time adaptation of vision-language models. *Second*, we introduce negative pseudo labeling to alleviate the adverse impact of pseudo label noises which makes TDA more robust to pseudo label noises and generalizable to testing data. *Third*, we evaluate TDA extensively over two benchmarks, and experiments show that TDA achieves superior accuracy and efficiency compared with the state-of-the-art.

2. Related Work

Vision-language models [6, 17, 19, 31, 43] have demonstrated significant potential in learning semantic representations effectively by undergoing extensive training on image-text data. CLIP [31] stands out among these models for its ability to establish links between visual and textual representations, which enables it to achieve impressive zero-shot results on various downstream tasks. To enhance the transfer learning capability of the CLIP model in the downstream classification tasks, researchers have proposed integrating language prompt learners such as CoOp [50] and CoCoOp [49], as well as vision adapters such as CLIP-Adapter [10], Tip-Adapter [47], CaFo [48], TaskRes [44] and GraphAdapter [25]. Although these methods have shown considerable performance improvements, they typically require a large amount of training data in downstream tasks, making them less practical for real-world scenarios. On other hand, this work focuses on a new paradigm named test-time adaptation without accessing the original training data.

Test-time adaptation refers to the process of adapting models to testing data that may have distributional differences from the training data. It is particularly beneficial for real-world applications that require models to be deployed in diverse environments, such as autonomous driving in various weather conditions, medical diagnosis in different hospitals, and etc. Several recent works utilize each batch of testing samples to update partial weights [18, 37, 38], normalization statistics [34], or a combination of both [40, 45]. To avoid updating models with multiple testing samples, MEMO [46] proposes enforcing the invariant predictions from different augmentations of each sample in the testing data stream. TPT [35] tackles the same challenge with vision-language models by fine-tuning a learnable prompt with each testing sample. DiffTPT [9] innovates test-time prompt tuning by leveraging pre-trained diffusion models to augment the diversity of test data samples used in TPT. Although TPT [35] and DiffTPT [9] are effective in addressing test-time adaptation of vision-language models, prompt learning is computationally expensive and time-consuming. This paper aims to mitigate the computational efficiency challenges of TPT and DiffTPT through the introduction of

a cache model.

Cache model benefits the adaptation techniques by providing efficient inference and non-parametric processing without requiring parameter updates. Unbounded Cache [12] and PSMM [27] have shown promising results in the text generation task by storing a large amount of the training dataset to capture long-term dependencies, however, this approach poses a challenge of memory efficiency during the test-time adaptation. A different technique [20] was proposed to mitigate this limitation by reducing the size of the cache memory through the search find application. Alternatively, Tip-Adapter [47] solves the cache memory problem by only storing few-shot samples per class to create a cache model for vision-language models. Inspired by this work, our dynamic adapter use the same architecture for the test-time adaptation setting, where there is no access to the source data, and testing samples can only be accessed one by one. To address the lack of access to source data during testing, our adapter collects the most reliable test samples and their pseudo labels to form cache model.

3. Method

3.1. Preliminaries

CLIP [31] is a vision-language model that performs a proxy task of predicting the correct pairings of text and image. Consider an N -class classification problem where the CLIP’s objective in the zero-shot setting is to match images with their most relevant textual descriptions using an image encoder E_v and a text encoder E_t . To obtain the textual descriptions, N -class names are concatenated with hand-crafted prompts and then mapped into the c -channeled text embeddings \mathbf{W}_c using the text encoder E_t .

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:

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

where \mathbf{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.

Tip-Adapter [47] provides a training-free solution that uses a key-value cache model and integrates knowledge from the pre-trained CLIP model with few-shot labeled samples. The cache model is created using a set of k -shot labeled samples x_k from N classes and their corresponding ground-truth labels y_N . It can be conceptualized as a linear two-layer model, where the first layer contains train image features $\mathbf{F}_{\text{train}} = E_v(x_k)$ and the second layer consists of one-hot vectors $\mathbf{L}_{\text{train}}$ encoded from the labels y_N . Given test image features f_{test} generated from the CLIP’s image encoder E_v , the prediction from the cache model can be calculated as follows:

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

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: $P_{\text{TA}}(f_{\text{test}}) = P_{\text{cache}}(f_{\text{test}}) + f_{\text{test}} \mathbf{W}_c^T$.

3.2. Training-free Dynamic Adapter

During testing, pre-trained vision-language models like CLIP may encounter distribution shifts that degrade the classification performance. To address this issue, existing test-time prompt tuning methods train a learnable prompt by enforcing consistency across different image augmentations during testing. However, it requires a large number of augmentation operations on each test image and computationally-heavy optimization steps to learn the prompt, limiting its applicability in various real-world settings.

In this paper, our motivation is to design an efficient method for test-time adaptation of the pre-trained vision-language models like CLIP. Inspired by the concept of Tip-Adapter, we propose a training-free dynamic adapter (TDA) to enable efficient and effective test-time adaptation with CLIP. As shown in Figure 2, TDA includes two lightweight key-value caches, where each cache stores a dynamic queue of few-shot test features as keys and the corresponding pseudo labels as values. The first cache is intended for positive learning and it dynamically updates key-value pairs with high-confidence predictions to improve the accuracy. The second cache is designed for negative learning and it aims to address the adverse effects of noisy pseudo labels by introducing negative pseudo labeling to identify class absence rather than presence. By combining the positive cache and negative cache, the proposed TDA can achieve superior performance in terms of both speed and accuracy.

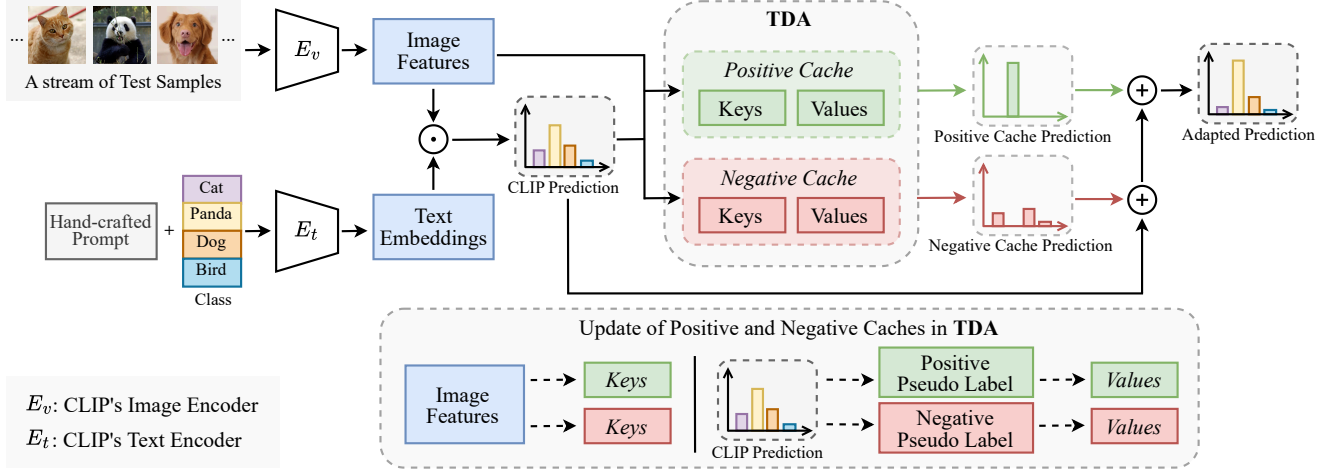


Figure 2. Overview of the proposed Training-free Dynamic Adapter (TDA). TDA constructs and updates two key-value caches to store the knowledge of a stream of test samples, and uses the two caches to generate positive and negative predictions which are combined with CLIP predictions to produce the final 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.

Our goal is to conduct test-time adaptation by gathering adequate knowledge from the testing data stream and improving the predictions through the adaptation of the image features. To accomplish this goal, we create positive and negative caches that capture particular characteristics of the testing data stream. In the upcoming parts, we will present the process of collecting data for each cache and define the conditions in which we can utilize cache information to adapt features and enhance model predictions.

Positive Cache. The positive cache in TDA is a key-value cache, in which keys and values are represented as a dynamic queue. It aims to collect high-quality few-shot pseudo labels $\hat{\mathbf{L}}_p$ as positive values and the corresponding features \mathbf{Q}_p as keys. The key-value cache is initially empty and then accumulates a sufficient number of key-value pairs during the test-time adaptation. To maintain high-quality pseudo labels, TDA progressively incorporates test predictions with lower entropy while limiting shot capacity¹ in the positive cache. Different from a normal queue like a FIFO queue with a fixed size, the dynamic queue in our method expands in size during testing. Besides, the dynamic queue operates similarly to a priority queue, using entropy as the criterion for prioritization. Note that, each class has its own queue to maintain the order and correct data structure of each class in the cache.

Given a pre-trained CLIP model that consists of a text encoder E_t and an image encoder E_v , E_t processes class

¹The shot capacity refers to the maximum number of pairs per class.

names with pre-defined prompts to generate c -channeled text embeddings \mathbf{W}_c and E_v processes each test image x_{test} to produce image features f_{test} . To build the positive cache, TDA first generates a pseudo label \hat{l} , a one-hot encoded vector of a categorical distribution, for each test sample x_{test} by applying the softmax function to the prediction $f_{\text{test}} \mathbf{W}_c^T$. We establish two conditions to ascertain whether and how to include the pseudo label \hat{l} and its corresponding image features f_{test} into the positive cache. The first condition is defined as: if the shot number (the number of collected pairs per class) of $\hat{\mathbf{L}}_p$ is less than the maximum shot capacity (the maximum capacity number of pairs per class) k , TDA will add \hat{l} and f_{test} as a new value and a new key to $\hat{\mathbf{L}}_p$ and \mathbf{Q}_p respectively. Meanwhile, the second condition is defined as: if the shot number of $\hat{\mathbf{L}}_p$ has reached the maximum shot capacity k , TDA will replace an ‘uncertain’ key-value pair $\{\mathbf{q}^{\text{ent}}, \hat{l}^{\text{ent}}\}$ with $\{f_{\text{test}}, \hat{l}\}$ when $H(f_{\text{test}} \mathbf{W}_c^T) < H(\mathbf{q}^{\text{ent}} \mathbf{W}_c^T)$. Here, H denotes the entropy function and the term ‘uncertain’ indicates that the entropy of a particular key-value pair is the highest compared to the entropy of all other key-value pairs of the same class in the cache model. By applying these two conditions, TDA can gradually integrate test predictions with lower entropy while controlling the shot capacity, which helps to ensure the collection of high-quality pseudo labels in positive cache.

During test-time adaptation, the positive cache can quickly retrieve relevant information by treating the image features f_{test} generated from a test example as a query and searching the stored key-value pairs $\{\mathbf{Q}_p, \hat{\mathbf{L}}_p\}$ for match-

ing information. The adapted predictions using the positive cache can then be obtained as:

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

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

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 \mathbf{Q}_n and $\hat{\mathbf{L}}_n$, respectively. It aims to gather CLIP-generated image features to \mathbf{Q}_n and the corresponding negative pseudo labels to $\hat{\mathbf{L}}_n$. Unlike the pseudo labels in the positive cache, the negative pseudo labels are obtained by applying negative mask on the class probabilities as:

$$\hat{\mathbf{L}}_n = -\mathbb{1}[p_l < P(\mathbf{Q}_n)], \quad (4)$$

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 represents a threshold in negative pseudo labeling, and $\hat{\mathbf{L}}_n$ denotes a negative pseudo label, which is a vector whose elements larger than p_l have a value -1 and otherwise 0. Different from existing negative learning methods [22, 23, 33] that select negative labels from all noisy labels, TDA selects negative pseudo labels from uncertain predictions to avoid bias to the data with certain predictions.

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 :

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

This condition is designed to mitigate the risk of prediction errors due to high entropy or be biased to certain predictions (characterized by very low entropy), by incorporating test samples that exhibit a moderate degree of prediction uncertainty. Once the $\gamma(f_{\text{test}})$ check is completed, the remaining steps for collecting uncertain samples in the negative cache follow the same two conditions designed in the positive cache. Similar to the positive cache, TDA also limits the shot capacity \tilde{k} in the negative cache.

During test-time adaption, the testing features f_{test} can be quickly adapted to target domains by retrieving the knowledge from $\{\mathbf{Q}_n, \hat{\mathbf{L}}_n\}$ in the negative cache and the adapted prediction can be obtained as:

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

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:

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

3.3. Relationship with TPT and Tip-Adapter

Both TPT and TDA are designed to handle the challenge of adapting models to test data that have distributional discrepancies from the training data. TPT trains a learnable prompt \mathbf{p}_c in Eq. (1) for each test sample with a large number of augmentations and such training process requires back-propagation and is computationally intensive. In contrast, our TDA is training-free as both positive cache $\{\mathbf{Q}_p, \hat{\mathbf{L}}_p\}$ and negative cache $\{\mathbf{Q}_n, \hat{\mathbf{L}}_n\}$ are non-parametric, which is super-efficient without incurring any backpropagation.

Our TDA employs a cache model that shares similarities with the Tip-Adapter, where features and labels are stored as key-value pairs in a memory cache. One notable distinction between the Tip-Adapter and TDA lies in the type of cache used. Specifically, the Tip-Adapter relies on a static cache $\{\mathbf{F}_{\text{train}}, \mathbf{L}_{\text{train}}\}$ because it is designed for a supervised adaptation setting, where the ground-truth labels $\mathbf{L}_{\text{train}}$ are predetermined and readily available. In contrast, our TDA introduces a new dynamic cache $\{\mathbf{Q}_p, \hat{\mathbf{L}}_p\}$ designed for a test-time adaptation setting, where the pseudo labels $\hat{\mathbf{L}}_p$ are generated on-the-fly from a stream of test samples. Furthermore, TDA incorporates a novel negative cache model $\{\mathbf{Q}_n, \hat{\mathbf{L}}_n\}$ that enhances testing predictions by utilizing indirect knowledge that a test image does not belong to certain negative classes. By combining positive and negative caches, our proposed TDA is more robust to noisy pseudo labels and can generalize well to testing data.

4. Experiments

4.1. Experimental Setup

Benchmarks. We conducted main experiments on two benchmarks: out-of-distribution (OOD) benchmark and cross-domain benchmark, both applied in the previous work [35] that adapts vision-language models in test time. The OOD benchmark serves as a measure of the robustness of our approach by involving assessment on 4 out-of-distribution datasets derived from ImageNet [5]: ImageNet-A [16], ImageNet-V2 [32], ImageNet-R [15], and ImageNet-S [41]. This benchmark is specifically designed to evaluate a model’s capacity to generalize to new and unseen data. The cross-domain benchmark, on the other hand, is involved to evaluate the model’s performance across 10 diverse image classification datasets, each from a distinct domain with different classes: Aircraft [26], Caltech101 [8], Cars [24], DTD [4], EuroSAT [14], Flower102 [28], Food101 [1], Pets [30], SUN397 [42], and UCF101 [36]. This benchmark provides a comprehensive evaluation of the model’s adaptability during test time across various 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

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.

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.

Implementation details. All the models in our experiments are built upon the pre-trained CLIP model [31] that consists of an image encoder and a text encoder. The image encoder can be either ResNet [13] or Vision Transformer [7], while the text encoder is Transformer [39]. Test-time adaptation is set for single-image scenarios, using a batch size of 1. We conduct a search for all our hyperparameters using a single ImageNet validation set. The threshold p_l for negative pseudo-labeling in Eq 4 is set as 0.03. The upper and lower thresholds $[\tau_l, \tau_h]$ for testing feature selection in Eq 5 are set as [0.2, 0.5]. Once searched, these hyperparameters are fixed and evaluated across various new datasets. To avoid incurring backpropagation when using learnable prompt, we follow [31] to use hand-crafted

prompts. We use top-1 accuracy (%), a standard classification criterion, as our evaluation metric. All the experiments are conducted using a single NVIDIA Quadro RTX 6000 GPU.

4.2. Comparisons with State-of-the-art

In this section, we compare our proposed TDA with several state-of-the-art methods, including CLIP [31], three train-time adaptation methods, *i.e.*, CoOp [50], CoCoOp [49], and Tip-Adapter [47], as well as two existing test-time adaptation methods TPT [35] and DiffTPT [9], all of which are designed for vision-language models. Specifically, CLIP is evaluated using an ensemble of 80 hand-crafted prompts as in [31]. All train-time adaptation methods are trained on the ImageNet train set with 16 shots per class and tested on other datasets as in [35]. CoOp utilizes a learnable context module of size 4 that is fine-tuned for specific downstream tasks, while CoCoOp is the generalized version of CoOp with the input-conditional token for image features. Tip-Adapter employs the optimal hyperparameters obtained from the ImageNet validation set during evaluation. We would like to note that Tip-Adapter is unable to handle new classes during testing, limiting its implementation to OOD benchmark evaluation where the training classes encompass all the testing classes. Different from train-time adaptation methods, test-time adaptation methods (*i.e.*, TPT, DiffTPT, and our TDA) do not utilize the ImageNet train set. Instead, they are fine-tuned with target datasets using a stream of unlabeled test samples. Follow-

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.

ing TPT and DiffTPT, we compare TDA with the state-of-the-art over two public benchmarks: OOD benchmark and cross-domain benchmark.

Results on the OOD Benchmark. We first compare TDA with state-of-the-art methods over the OOD benchmark. Table 1 presents the experimental results, highlighting the superior performance of the proposed TDA compared to both TPT and DiffTPT across various OOD datasets derived from ImageNet. Specifically, TDA outperforms TPT on both ResNet-50 and ViT-B/16 architectures, improving OOD accuracy by 2.74% and 3.08% on average, respectively. Furthermore, compared to DiffTPT, TDA exhibits an average accuracy improvement of 0.94% and 3.37% in the OOD benchmark for ResNet-50 and ViT-B/16, respectively. These results validate the effectiveness of TDA in enhancing test-time adaptation performance on various OOD test datasets.

In order to provide a more comprehensive evaluation of our proposed method’s efficiency and effectiveness, we compared it with the baseline CLIP-ResNet-50 and two existing test-time adaptation methods (*i.e.*, TPT and DiffTPT). This comparison encompasses both testing time and testing accuracy, and the corresponding results are shown in Table 2. This evaluation is performed on the ImageNet validation dataset, which consists of 50,000 images, using a single NVIDIA Quadro RTX 6000 GPU. When compared to the baseline CLIP-ResNet-50, the proposed TDA demonstrates a significant improvement in testing accuracy (+1.54%), with only a minimal sacrifice in testing efficiency (requiring an additional 4min). In comparison to TPT and DiffTPT, the proposed TDA demonstrates not only superior

testing accuracy but also significantly improved efficiency. It reduces the testing time dramatically from 12h 50min by TPT and even more from 34h 45min by DiffTPT, down to just 16 minutes. Without including the image generation time, DiffTPT consumes clearly more test time than TPT as it involves a time-consuming multi-step prompt updating process whereas TPT requires a single step only. The experimental results strongly validate the effectiveness and efficiency of our proposed method, establishing its suitability for real-world applications.

We then compare TDA with state-of-the-art methods over cross-domain benchmark. The results, presented in Table 3, demonstrate that TDA not only surpasses the performance of the TPT method but also shows a significant advantage over its improvement method DiffTPT. Specifically, when utilizing CLIP-ResNet-50 and CLIP-ViT-B/16 as the backbone, TDA achieves an improvement in average accuracy over TPT by 3.37% and 2.43%, respectively. These improvements, along with a 1.18% and 2.06% gain over DiffTPT for the respective backbones, further verify the effectiveness of TDA in adapting to diverse class datasets during test time. This attribute holds significant value for vision-language models such as CLIP, as it enables them to classify arbitrary classes in image classification without the need for additional training.

4.3. Ablation Studies

In this section, we perform ablation studies to examine the effectiveness of our designs. All the ablation studies are conducted over the ImageNet dataset, where TDA can achieve an accuracy of 61.35% under default settings. TDA consists of a *Positive Cache* and a *Negative Cache*, which

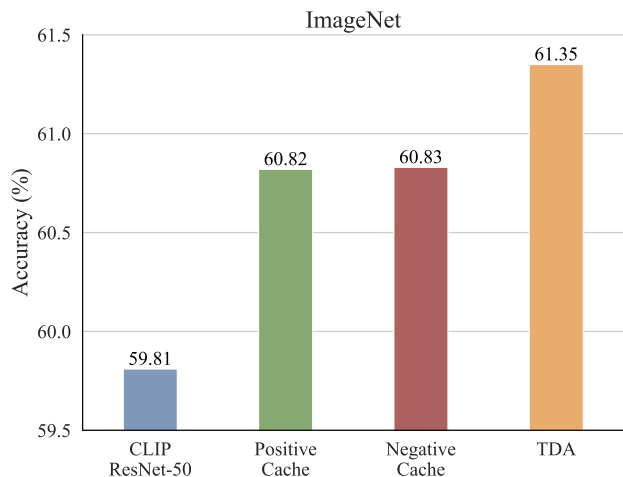


Figure 3. Ablation studies on two cache designs in TDA: *Positive Cache* and *Negative Cache*. All the models are built upon the baseline model CLIP-ResNet-50.

perform positive and negative pseudo labeling within our designed dynamic adapter, respectively. We first assess the efficacy of the two cache designs in TDA. As shown in Figure 3, both *Positive Cache* and *Negative Cache* significantly surpass the baseline model CLIP, demonstrating that test-time adaptation can be improved by introducing a dynamic adapter with either positive pseudo labeling or negative pseudo labeling. Besides, the two cache designs in TDA can complement each other as TDA (*i.e.*, the combination of the two designs) clearly outperforms either *Positive Cache* or *Negative Cache* on the challenging ImageNet dataset. Moreover, we extend our ablation studies to the cross-domain benchmark, and results show that the positive cache achieved 60.38% accuracy, the negative cache 60.11%, while their combination yielded a 61.03% accuracy, thereby highlighting the significance of each type of cache in enhancing TDA’s performance.

We proceed to perform parameter studies on the shot capacity, which refers to the maximum number of key-value pairs per class, in both *Positive Cache* and *Negative Cache* models. These studies aim to find the optimal balance between the diversity and accuracy of the key-value pairs. Figure 4 shows that the performance of both cache models is significantly affected when the shot capacity is either too low or too high. We find that the shot capacity is set as 3 for the *Positive Cache* and 2 for the *Negative Cache* yields the best performance. This is because an appropriate shot capacity ensures both high-quality pseudo labels (paired values) and diverse image features (paired keys) in the cache models. The negative cache is sensitive to shot numbers due to its role in storing probabilities for various negative pseudo labels, each representing a class to be excluded from the model. Contrary to the intuition that a larger negative

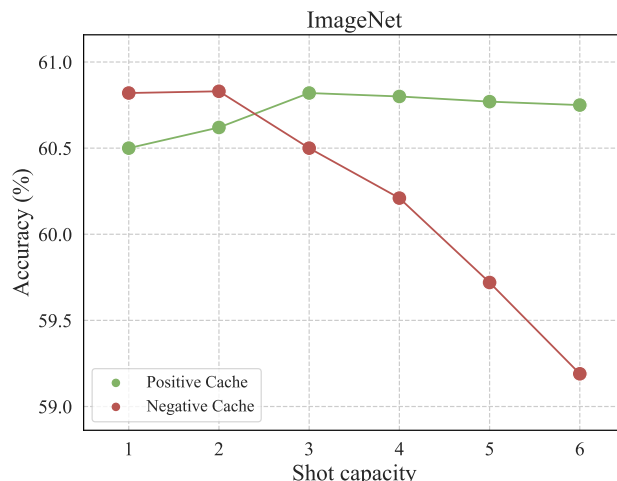


Figure 4. Parameter studies on the *Shot Capacity* in *Positive Cache* and *Negative Cache*.

cache might be beneficial, a larger negative cache leads to more high-entropy, noisier pseudo labels, as highlighted in self-training studies like [11], thereby lowering the performance. Conversely, the positive cache, which stores single and high-confidence prediction, is less affected by variations in shot capacity, thereby maintaining consistent accuracy across different shot capacities. To facilitate practical applications, the shot capacity settings are fixed and directly applied to new datasets without the need for additional parameter adjustments.

5. Conclusion

In this work, we have presented TDA, a dynamic adapter for efficient and effective test-time adaptation of vision-language models. The proposed method employs a key-value cache, which maintains a dynamic queue with test-sample features as keys and corresponding few-shot pseudo labels as values, allowing for gradual adaptation to test data through progressive pseudo label improvement. Moreover, TDA introduces a negative cache to mitigate the undesirable effects of noisy pseudo labels by assigning negative pseudo labels to certain classes when the model is uncertain about its predictions. The results of extensive experiments over two benchmarks demonstrate that TDA outperforms state-of-the-art test-time adaptation methods while significantly reducing testing time. This work contributes to the research and application values of test-time adaptation and presents a promising solution to the efficiency issue of test-time adaptation of vision-language models.

Acknowledgement

This study is supported under the Mohamed bin Zayed University of Artificial Intelligence.

References

- [1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 5
- [2] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022. 1
- [3] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022. 1
- [4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5
- [6] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*, 2021. 2
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6
- [8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshops*, 2004. 5
- [9] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023. 1, 2, 6
- [10] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2
- [11] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 2, 8
- [12] Edouard Grave, Moustapha M Cisse, and Armand Joulin. Unbounded cache model for online language modeling with open vocabulary. *Advances in neural information processing systems*, 30, 2017. 2, 3
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6
- [14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 2019. 5
- [15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 5
- [16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 5
- [17] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17980–17989, 2022. 2
- [18] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021. 2
- [19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1, 2
- [20] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 3
- [21] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. 2
- [22] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 101–110, 2019. 2, 5
- [23] Youngdong Kim, Juseung Yun, Hyounguk Shon, and Junmo Kim. Joint negative and positive learning for noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2021. 2, 5
- [24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013. 5
- [25] Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. Graphadapter: Tuning vision-language models with dual knowledge graph. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [26] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 5
- [27] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 3
- [28] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 5

- [29] Emin Orhan. A simple cache model for image recognition. *Advances in Neural Information Processing Systems*, 31, 2018. [2](#)
- [30] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. [5](#)
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. [1](#), [2](#), [3](#), [6](#)
- [32] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. [5](#)
- [33] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021. [2](#), [5](#)
- [34] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020. [2](#)
- [35] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, 2022. [1](#), [2](#), [3](#), [5](#), [6](#)
- [36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. [5](#)
- [37] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. [2](#)
- [38] Thomas Varsavsky, Mauricio Orbes-Arteaga, Carole H Sudre, Mark S Graham, Parashkev Nachev, and M Jorge Cardoso. Test-time unsupervised domain adaptation. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 428–436. Springer, 2020. [2](#)
- [39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [6](#)
- [40] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. [1](#), [2](#)
- [41] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. [5](#)
- [42] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. [5](#)
- [43] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liquan Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15671–15680, 2022. [1](#), [2](#)
- [44] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. [2](#)
- [45] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678, 2021. [2](#)
- [46] Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, 2022. [2](#)
- [47] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022. [2](#), [3](#), [6](#)
- [48] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222, 2023. [2](#)
- [49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. [2](#), [6](#)
- [50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. [2](#), [6](#)