Robust Test-Time Adaptation for Zero-Shot Prompt Tuning

Ding-Chu Zhang[*], **Zhi Zhou**[*], **Yu-Feng Li**[†]

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{zhangdc,zhouz,liyf}@lamda.nju.edu.cn

Abstract

CLIP has demonstrated remarkable generalization across diverse downstream tasks. By aligning images and texts in a shared feature space, they enable zero-shot classification via hand-crafted prompts. However, recent studies have shown that hand-crafted prompts may be unsuitable in practical applications. Specifically, choosing an appropriate prompt for a given task requires accurate data and knowledge, which may not be obtainable in practical situations. An inappropriate prompt can result in poor performance. Moreover, if there is no training data, tuning prompts arbitrarily through unlabeled test data may lead to serious performance degradation when giving hand-crafted prompts. Our study reveals that the aforementioned problems are mainly due to the biases in testing data (*Data Bias*) and pre-trained CLIP model (*Model Bias*). The *Data Bias* makes it challenging to choose an appropriate prompt, while *Model Bias* renders some predictions inaccurate and biased, which leads to error accumulation. To address these biases, we propose robust test-time **Ada**ptation for zero-shot **Prompt** tuning (ADAPROMPT). Specifically, we ensemble multiple prompts to avoid the worst-case results and dynamically tune prompts to adapt to *Data Bias* during testing. Furthermore, we adopt a confidence-aware buffer to store balanced and confident unlabeled test data to tune prompts in order to overcome *Model Bias*. Our extensive experiments on several benchmarks demonstrate that ADAPROMPT alleviates model bias, adapts to data bias and mostly outperforms the state-of-the-art methods at a small time cost. Moreover, our experimental results reveal that ADAPROMPT hardly encounters any performance degradation on these datasets.

Introduction

Benefited from recent advances in computer vision (Radford et al. 2021; Jia et al. 2021) and natural language processing (Kenton and Toutanova 2019; Brown et al. 2020; Shi, Wei, and Li 2024), large pre-trained vision-language models like CLIP (Radford et al. 2021) have shown the outstanding generalization on numerous downstream tasks. These models align visual and textual contents within a common feature space through training with millions of noisy image-text pairs. This enables zero-shot classification (Wei et al. 2022) using appropriately hand-crafted prompts, greatly reducing the cost of deploying models in real-world applications. However, the appropriate prompt, which is challenging to choose in practical applications, plays a crucial role in downstream tasks.

Prompt tuning, a method that optimizes the prompt by using data from downstream tasks, is an effective way to tackle the previous problems. Some studies (Zhou et al. 2022b,a) optimize the prompt with the help of training data, which can eliminate the need for us to manually select prompts. However, we need to collect accurate data for training, which may be difficult or expensive in practical situations. Recent studies (Shu et al. 2022) propose to fine-tune the prompt by using unlabeled test data, solving the problem of unavailable training data (Guo, Zhou, and Li 2020; Zhu et al. 2023; Tian et al. 2023; Jia et al. 2024; Guo and Li 2024). However, they encounter performance degradation on certain domains. In addition, they also use a large amount of data augmentation, which requires the model to predict at a long time cost.

Therefore, it is urgent to study a zero-shot classification method that does not require us to manually choose the optimal prompts and can also solve the robustness of prompt tuning at a small time cost. We demonstrate that existing problems are caused by two biases, *Data Bias* and *Model Bias*, through experimental results. Specifically, *Data Bias* causes a problem that the performance of different prompts can vary across datasets, resulting in the difficulty of selecting an optimal prompt for downstream tasks. *Model Bias* causes prediction biases towards specific classes, leading to error accumulation. And the errors accumulated by *Model Bias* will finally result in performance degradation problem.

To this end, we propose robust test-time adaptation for zero-shot prompt tuning, which updates the efficient and reliable prompts for CLIP model at a small time cost by using unlabeled test data. To tackle the *Data Bias*, we propose an ensemble-tuning method for prompts optimization during the testing. Specifically, we ensemble multiple hand-crafted prompts, such as "an image of a", "a colorful picture of a" and "a noisy image of a", to avoid the worst-case prediction. Meanwhile, we fine-tune all prompts with unlabeled test data to adapt to *Data Bias*. Then, a confidence-aware data buffer is proposed to eliminate the problem of *Model Bias* during the updating process. Specifically, we store high-confidence, class-balanced samples, which ensures robust

[*]These authors contributed equally.
[†]Corresponding author
Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

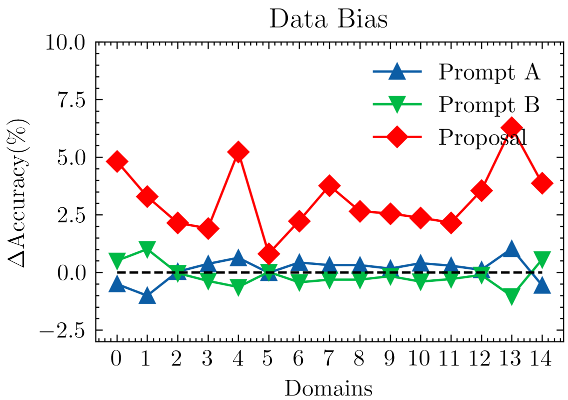

Figure 1: Relative performance compared to the average performance of prompts evaluated on CIFAR10-C in 15 domains with corruption level 3 using different initial prompts.

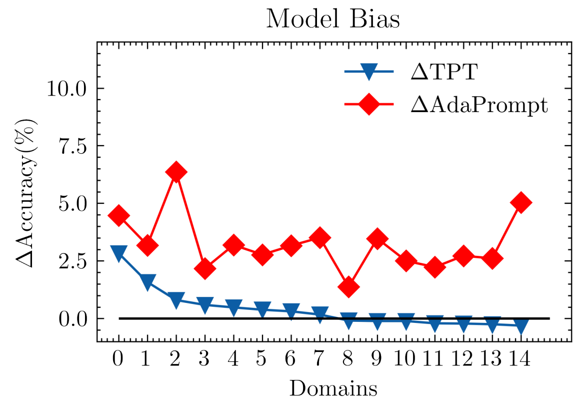

Figure 2: Relative performance compared to baseline evaluated on CIFAR10-C in 15 different domains with corruption level 3. The black line represents the baseline.

updates in a balanced and confident way as much as possible. The experiments show that ADAPROMPT mostly outperforms the state-of-the-art methods and hardly encounters any performance degradation on several benchmarks spending a small amount of time.

Our main contributions are highlighted as follows:

(a) We empirically analyze existing prompt tuning methods by using unlabeled test data. Based on our analysis, we point out the *Data Bias* and *Model Bias* issues. Existing methods cannot address these two issues effectively at a small time cost.

(b) We propose the novel ADAPROMPT, containing Prompt Ensembling, Test-time Prompt Tuning, and Confidence-aware Buffer, which effectively tackles the previously proposed *Data Bias* and *Model Bias* issues.

(c) We evaluate our framework on multiple benchmark datasets. Our experiment results show that the proposed ADAPROMPT mostly outperforms the state-of-the-art test-time prompt tuning methods consuming a small amount of time.

Problem and Analysis

This section provides an overview of the problems and the notations used. We describe zero-shot classification with CLIP model and then introduce problems of tuning prompts through using unlabeled test data. Specifically, we analyze the two major problems in previous studies (Wei et al. 2022; Shu et al. 2022), i.e. *Data Bias* and *Model Bias*.

Problem Formulation

We focus on the multi-class classification with input space $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ and $\mathcal{Y} = \{y_1, ..., y_K\}$ for a K-class classification task. We denote a CLIP (Radford et al. 2021) model as $\mathcal{F} = \{\mathbf{E}_{visual}, \mathbf{E}_{text}\}$, with $\mathbf{E}_{visual}$ and $\mathbf{E}_{text}$ being the image and text encoders.

In the zero-shot classification task, we are given a CLIP model $\mathcal{F}$ and a single test sample $\mathbf{x}_t \in \mathcal{X}$ of class y_t, where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$. Then, we prepend a hand-crafted prompt prefix, such as **p=**"a photo of a", to every $y_i \in \mathcal{Y}$ to form the category-specific text inputs $\{\mathbf{p}; y_i\}$. We pass these category-specific text inputs to the text encoder to get the text features $\{\mathbf{t}_1, ..., \mathbf{t}_K\}$, where $\mathbf{t}_i = \mathbf{E}_{text}(\{\mathbf{p}; y_i\})$. Each text feature $\mathbf{t}_i$ is paired with the image feature $\mathbf{v}_t = \mathbf{E}_{visual}(\mathbf{x}_t)$ to compute a similarity score $\mathbf{s}_i = \text{sim}(\mathbf{t}_i, \mathbf{v}_t)$, where $\text{sim}(\cdot,\cdot)$ denotes the cosine similarity. The prediction probability on $\mathbf{x}_t$ can be denoted by $f(y_i|\mathbf{x}_t; \mathbf{p}) = \frac{\exp(\mathbf{s}_i\cdot\tau)}{\sum_{j=1}^{K}\exp(\mathbf{s}_j\cdot\tau)}$, where τ is the pre-defined temperature of the softmax function.

In test-time prompt tuning, we apply the CLIP model $\mathcal{F}$ to downstream tasks with a hand-crafted prompt $\mathbf{p}_0$, which is a learnable vector. The probability of zero-shot prediction is denoted as $f(y_i|\mathbf{x}; \mathbf{p}) : \mathcal{X} \rightarrow [0, 1]$. At each timestamp t, the model adaptively evolves its parameter $\mathbf{p}_{t-1} \rightarrow \mathbf{p}_t$ using unlabeled test data and gives the predictions. The goal of ADAPROMPT is to adaptively update the prompt in a single domain for better performance.

Problem Analysis

The empirical results presented in Figure 1 illustrate that the optimal prompt varies across domains. Specifically, we define "an image of a" as the Prompt A and use "a noisy picture of a" as the Prompt B. The black dashed line indicates the average performance of these two prompts on each domain. The relative performance of Prompt A and Prompt B rises and falls on different domains, which demonstrates that the certain prompt may perform well in a given domain, while it may perform badly in other domains. The results indicates that it is difficult to choose a prompt that is optimal for all domains. Without any knowledge or data from the specific downstream task, it is impossible to choose an effective prompt for zero-shot classification. We named this phenomenon described above *Data Bias*. This phenomenon requires the zero-shot classification method to adaptively optimize prompts for different data.

Recent studies, e.g., TPT (Shu et al. 2022), propose to tune the prompt at test time, which tries to solve the problem

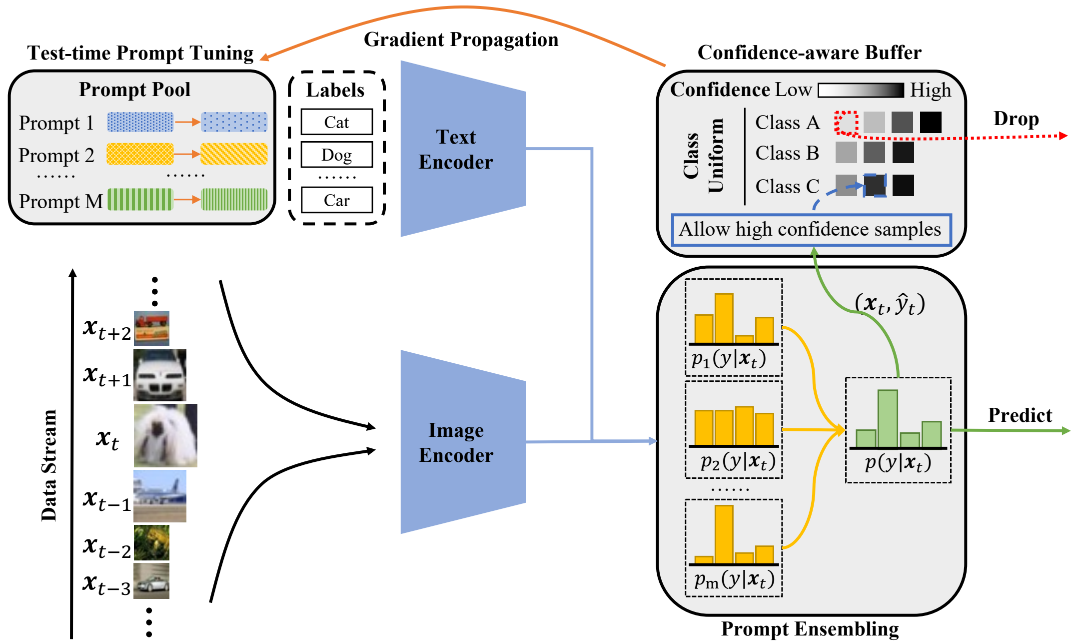

Figure 3: ADAPROMPT for image classification. We select confident samples from unlabeled test data stream by ensembling multiple prompts and push them into the confidence-aware buffer, which is used to store confident and balanced samples. Then, ADAPROMPT extracts samples from the buffer and adaptively updates all the prompts, which adapts prompts to current data.

of *Data Bias*. They optimize prompt using only unlabeled test data by minimizing the following marginal entropy:

$$\mathcal{L}(\mathbf{x}) = -\sum_{k=1}^{K} \tilde{f}(y_k|\mathbf{x};\mathbf{p}) log \tilde{f}(y_k|\mathbf{x};\mathbf{p}) \tag{1}$$

They additionally adopt test-time augmentation (Shanmugam et al. 2021) and confidence-based sample selection methods to enhance robustness of test-time prompt tuning:

$$\tilde{f}(y|\mathbf{x};\mathbf{p}) = \frac{1}{\rho N}\sum_{i=1}^{N} \mathbb{1}[\mathbf{H}(f_i) \leq \alpha] f(y|\mathcal{A}_i(\mathbf{x});\mathbf{p}) \tag{2}$$

where α is the entropy threshold and the ρ is a cutoff percentile parameter on N augmentation functions $\mathcal{A}(\mathbf{x})$ across augmented views. Although TPT adopts test-time augmentation to enhance the robustness of prompt tuning, they need 63 random data augmentation for each image and then reset their prompt state, which consumes a lot of time to predict. Futhermore, our experimental results in Figure 2 indicate that it still faces performance degradation compared to baseline across multiple domains. We claim that this phenomenon is caused by *Model Bias*, i.e., the pre-trained CLIP model has prediction bias in different domains. Test-time prompt tuning accumulates these errors further into the optimized prompt and ultimately leads to performance degradation. Moreover, *Model Bias* disables TPT method to continuously tune the prompts, leading to the collapse of the model as demonstrated in Table 1.

Methodology

Existing studies (Shu et al. 2022; Wei et al. 2022) using the pre-trained CLIP model face two serious problems: *Data Bias* and *Model Bias*. In this section, we propose ADAPROMPT with three modules:

(a) Prompt Ensembling: We use multiple prompts to ensemble the output results, alleviating the negative impact of *Data Bias* on a single prompt and avoiding the worst-case prediction.

(b) Test-time Prompt Tuning: We use unlabeled test data to tune the prompts in order to adapt them to the *Data Bias* and improve the accuracy of prediction.

(c) Confidence-aware Buffer: Due to *Model Bias*, imbalanced prompt tuning can lead to error accumulation. Therefore, we use a confidence-aware buffer to store confident and balanced samples and use them to update prompts to alleviate the *Model Bias*.

Note that ADAPROMPT is independent on TPT. Test-time augmentation, used in TPT, also can enhance robustness of ADAPROMPT. However, a lot of data augmentations(63 for each image in TPT) will consume a lot of time through CLIP model, which may not be a good choice for test data stream in practical situations. Below we present a detailed description of three modules in ADAPROMPT. The overall illustration of ADAPROMPT framework is presented in Figure 3.

Prompt Ensembling

As described in Section **Problem Analysis**, performance of different prompts can vary across domains. So we use different hand-crafted prompts and ensemble their predictions

Algorithm 1: Confidence-aware Buffer

Input: sample $\mathbf{x}_t$, pseudo label $\hat{y}(\mathbf{x}_t)$, confidence $c(\mathbf{x}_t)$
Parameter: threshold τ

if $c(\mathbf{x}_t) > \tau$ **then**
  if buffer is not full **then**
    Add($\mathbf{x}_t$,$\hat{y}(\mathbf{x}_t)$,$c(\mathbf{x}_t)$)
  else
    M $\leftarrow$ majority class(es) in buffer
    if $\hat{y}(\mathbf{x}_t) \notin M$ **then**
      Randomly select a class and discard one instance ($\mathbf{x}_i$,$\hat{y}(\mathbf{x}_i)$,$c(\mathbf{x}_i)$) with the lowest confidence in that class where $\hat{y}(\mathbf{x}_i) \in M$
      Add($\mathbf{x}_t$,$\hat{y}(\mathbf{x}_t)$,$c(\mathbf{x}_t)$)
    else
      $c(\mathbf{x}_j) \leftarrow$ the minimum confident value in class $\hat{y}(\mathbf{x}_t)$
      if $c(\mathbf{x}_j) < c(\mathbf{x}_t)$ **then**
        Discard the instance ($\mathbf{x}_j$,$\hat{y}(\mathbf{x}_j)$,$c(\mathbf{x}_j)$) in buffer
        Add($\mathbf{x}_t$,$\hat{y}(\mathbf{x}_t)$,$c(\mathbf{x}_t)$)
      end if
    end if
  end if
end if

to alleviate the negative effects of *Data Bias* and avoid the worst-case results. Let M denote the number of prompts we use. We obtain the ensembled probability of different prompts in the following formulation:

$$\hat{f}(y|\mathbf{x}_t;\mathbf{p}) = \frac{1}{M}\sum_{i=1}^{M} f(y|\mathbf{x}_t;\mathbf{p}^i) \tag{3}$$

Specifically, we use a common small number of hand-crafted prompts, such as ”an image of a”, ”a colorful image of a” and ”a noisy picture of a”. And we obtain pseudo label and confidence for each sample in following formulation:

$$\begin{aligned} \hat{y}(\mathbf{x}_t) &= argmax_k \hat{f}(y_k|\mathbf{x}_t;\mathbf{p}) \\ c(\mathbf{x}_t) &= max_k \hat{f}(y_k|\mathbf{x}_t;\mathbf{p}) \end{aligned} \tag{4}$$

Test-time Prompt Tuning

In order to adapt all prompts to test data stream, we optimize all prompts using unlabeled test data by cross-entropy loss. The optimization objective is formalized as follows:

$$L(\mathbf{x}_t) = -\sum_{k=1}^{K} \hat{y}_k(\mathbf{x}_t) log\ \hat{f}(y_k|\mathbf{x}_t;\mathbf{p}) \tag{5}$$

where K represents the number of classes and $\hat{y}$ represents the pseudo label obtained by Eq. (4). The purpose of minimizing cross-entropy loss is to make the model more confident in the predicted samples, which can adapt prompts to *Data Bias* and improve the accuracy of predictions.

Confidence-aware Buffer

The temporal incoming batch of samples is random and the predictions may be biased and inaccurate, and thus adaptation with a batch of biased and inaccurate unlabeled test samples by Eq. (5) may exacerbate the bias of model predictions and degrade the performance of the model. To alleviate the problem of *Model Bias*, we propose a confidence-aware buffer that uses a small buffer with confidence as the priority and pseudo label balanced to store unlabeled samples from test data stream. For confidence as the priority, the higher confidence of the sample, the more accurate the prediction will be, making it less likely to cause erroneous updates. For pseudo label balance, we first compute the majority class(es) in the buffer and then replace the lowest confident sample of the majority class(es) with a new one. In addition, to ensure the accuracy of the samples entering the buffer, we use a threshold τ to filter out samples with low confidence. We detail the algorithm of confidence-aware buffer as a pseudo-code in Algorithm 1. Through the mechanism of confidence-aware buffer, we can ensure robust updates in a balanced and confident way by using samples in buffer, which can alleviate the *Model Bias*.

Experiments

In this section, we conduct experiments to answer the following questions:

RQ1: Does our proposed method perform better than existing test-time prompt tuning methods?

RQ2: Whether our proposed method alleviate the problem of *Data Bias*?

RQ3: Does ADAPROMPT relieve the problem of *Model Bias* on CLIP model?

Experimental Setup

Datasets. We conduct experiments on two standard benchmarks: CIFAR10-C and CIFAR100-C (Hendrycks and Dietterich 2019), which contain 15 corrupt testing sets. Each corrupt testing set has 10000 32×32 test images associated with 10/100 classes. Different from the previous methods that require training on the training set, we directly update prompts with unlabeled test data and then predict on them. We report the results evaluated on two different corruption level, 3 and 5.

Compared Methods. We compare our ADAPROMPT with the existing test-time prompt tuning methods that are designed for CLIP. CLIP (Radford et al. 2021), as our baseline, proposes to use contrastive loss to pull together images and their textual descriptions while pushing away unmatched pairs in the feature space. TPT (Shu et al. 2022) optimizes the prompt with a single test sample to encourage consistent predictions across augmented views by minimizing the marginal entropy and introduce confidence selection to filter out noisy augmentations. TPT-Continual continuously updates prompt with each sample in a single domain. In addition, all compared methods use a default prompt ”a photo of a”.

Implementation Details. We adopt the pre-trained CLIP model where visual model is ViT-B/16 (Dosovitskiy et al.

Dataset		CIFAR10-C(s=3)			CIFAR10-C(s=5)			CIFAR100-C(s=3)			CIFAR100-C(s=5)		
Methods		Source	TPT	Ours	Source	TPT	Ours	Source	TPT	Ours	Source	TPT	Ours
Noise	Gauss.	50.03	52.86	**54.50**	38.00	40.08	**42.48**	27.81	25.54	**28.61**	19.60	17.31	**21.92**
	Shot	61.74	63.32	**64.92**	43.14	44.74	**47.89**	33.81	32.22	**35.30**	21.36	19.04	**23.95**
	Impul.	78.59	78.87	**81.36**	56.70	59.08	**60.59**	47.30	47.63	**50.51**	25.31	25.65	**30.06**
Blur	Defoc.	85.46	85.25	**87.69**	72.88	72.10	**74.98**	60.10	**60.55**	60.54	42.52	42.73	**43.07**
	Glass	54.26	53.95	**59.29**	42.59	43.19	**47.51**	29.35	29.21	**30.38**	20.06	19.97	**20.91**
	Motion	77.15	77.06	**78.52**	70.96	70.14	**72.54**	48.69	48.86	**49.69**	**43.15**	42.63	42.46
	Zoom	81.57	81.35	**84.29**	74.66	74.89	**78.30**	56.08	55.96	**57.22**	47.89	48.12	**48.72**
Weather	Snow.	81.01	81.18	**84.52**	74.74	75.32	**78.26**	53.90	55.41	**56.34**	48.35	**49.19**	48.95
	Frost	81.13	81.02	**84.60**	78.40	78.33	**80.19**	53.12	53.89	**55.05**	49.72	50.43	**50.89**
	Fog	86.60	86.49	**89.10**	71.66	72.54	**73.14**	60.77	**61.64**	61.33	41.64	**42.71**	42.45
	Brit.	88.92	88.67	**91.53**	85.00	85.12	**88.06**	64.88	65.39	**66.64**	57.02	57.58	**59.07**
Digital	Contr.	87.11	87.70	**89.28**	63.00	**70.80**	67.95	59.77	61.18	**61.58**	34.54	**38.06**	36.84
	Elastic	80.27	80.75	**83.46**	55.40	57.10	**58.88**	52.53	53.43	**55.01**	29.21	30.05	**30.56**
	Pixel	75.18	75.98	**81.54**	48.09	52.24	**57.21**	51.09	51.94	**53.29**	23.94	25.15	**27.50**
	JPEG	69.51	69.82	**72.67**	60.30	61.55	**63.83**	39.68	40.17	**42.40**	32.46	32.43	**34.29**
Avg.		75.90	76.29	**79.15**	62.37	63.81	**66.12**	49.26	49.54	**50.93**	35.78	36.07	**37.44**

Table 1: Comparison with state-of-the-art test-time prompt tuning methods on CIFAR10-C and CIFAR100-C benchmarks with corruption level 3 and 5. We conduct separate tests on 15 different domains for each benchmark. We omit std in this table due to space issues. The best results are indicated in bold. Our method outperforms comparison methods in almost all cases. The best performance is in bold.

Method	CIFAR10-C(s=3)	CIFAR10-C(s=5)
P_A	75.91 ± 0.00	62.37 ± 0.00
P_B	76.21 ± 0.00	62.77 ± 0.00
P_C	72.98 ± 0.00	59.25 ± 0.00
P_{best} + UP.	77.72 ± 0.24	65.32 ± 0.18
P_e	75.38 ± 0.00	61.75 ± 0.00
P_e + UP.	$\mathbf{79.15 \pm 0.23}$	$\mathbf{66.12 \pm 0.43}$

Table 2: Evaluation of each module on CIFAR10-C with corruption level 3 and 5. The average accuracy of different modules on 15 different domains is shown.

2021) as the backbone and don't involve any training process. For baseline, we use a non-updated CLIP with a hand-crafted prompt, which is "a photo of a", to predict the results. For TPT, we use their original hyperparameters in their paper. For TPT-Continual, we use the same hyperparameters as TPT. For ADAPROMPT , we set 64 as our buffer size and three different hand-crafted prompts for ensembling, which are "an image of a","a colorful image of a" and "a noisy picture of a". Moreover, we set the batch size to 64 following previous studies (Boudiaf et al. 2022; Niu et al. 2022). The AdamW optimizer optimizes all the prompts with a learning rate of 0.005. We report mean $\pm$ std accuracy over five runs with random seed setting to 0, 1, 2, 3, 4.

Experimental Results

RQ1: Does our proposed method perform better than existing test-time prompt tuning methods?

To demonstrate the effectiveness of ADAPROMPT, we compare ADAPROMPT with the existing test-time prompt tuning methods to answer the question. Table 1 gives the detailed results on CIFAR10-C and CIFAR100-C datasets with corruption level 3 and 5. We evaluate each method on a single domain in order. The results show that ADAPROMPT consistently outperforms existing test-time prompt tuning methods on almost every domain. Especially on the CIFAR10-C dataset with corruption level 3, our method achieves optimal results in each domain and 2.86% average accuracy improvement compared to the SOTA method TPT.

RQ2: Whether our proposed method alleviate the problem of *Data Bias*?

To validate that our method alleviates *Data Bias* on the pre-trained CLIP model, we add a set of ablation experiments. The detailed results are shown in Table 2. Table 2 gives the average results on CIFAR10-C dataset with corruption level 3 and 5. The first three rows show the average performance of using three different prompts separately. Then, we select the best prompt from the first three rows and update it in a single domain, which is shown in the fourth row. Moreover, we ensemble the predictions of three prompts without updates in the fifth row. Finally, we update all prompts and ensemble their outputs in the last row. We can find that although the performance of ensembling without updates may not be as good as a certain good prompt, we do avoid the worst prediction results of a single prompt. Furthermore, we adapt the prompts to test data stream by updating all prompts, which improves performance and thereby alleviates *Data Bias*. To further explain how the hand-craft prompts affect performance, we present the performance of different hand-crafted prompts.

Methods		Source	TPT	TPT-C	Ours
Noise	Gauss.	15.72	16.29	0.52	**17.52**
	Shot	23.44	23.86	0.52	**26.47**
	Impul.	17.47	17.58	0.52	**20.76**
Blur	Defoc.	32.43	32.65	0.58	**34.39**
	Glass	11.88	12.51	0.52	**14.45**
	Motion	31.97	32.31	0.54	**33.98**
	Zoom	30.99	31.57	0.54	**33.32**
Weather	Snow.	29.69	30.90	0.55	**32.82**
	Frost	32.98	33.25	0.58	**36.30**
	Fog	35.81	36.36	0.58	**37.97**
	Brit.	43.95	43.62	0.60	**46.80**
Digital	Contr.	22.56	23.00	0.52	**25.52**
	Elastic	38.14	38.74	0.58	**40.78**
	Pixel	26.38	27.72	0.55	**29.42**
	JPEG	37.54	37.56	0.64	**40.72**
Avg.		28.73	29.20	0.55	**31.42**

Table 3: Comparison with SOTA test-time prompt tuning methods on TinyImageNet-C with corruption level 3. ADAPROMPT outperforms them in all domains.

RQ3: Does ADAPROMPT relieve the problem of *Model Bias* on CLIP model?

To verify ADAPROMPT's effectiveness in relieving *Model Bias*, we conduct experiments in CIFAR100-C *contrast* domain with corruption level 3. The details are shown in Figure 4. We can see that performance of TPT-Continual becomes very bad, which validates that *Model Bias* has a significant impact on model updates. Compared with baseline, our method achieves performance improvement, which proves that ADAPROMPT relieves *Model Bias*.

Ablation Study

We investigate the contribution of each module on the CIFAR10-C dataset with corruption level 3 and 5. The results are shown in the Table 4. The first row gives the performance of a non-updated CLIP with the best prompt among three prompts we set. Then, in the second row, M_e is added to ensemble all prompts. We can see that using only multiple prompts for ensembling without updates may not improve performance. Moreover, M_u is added to update the prompt in a balanced way by using unlabeled test data in buffer. We can see that balanced updates can adapt to *Data Bias* in the third row. Finally, we show the performance of all modules in the fourth row. We can see that updating multiple prompts together and then ensembling can adapt to current test data stream better, which verifies the effectiveness of ADAPROMPT and validates that the two modules, i.e., M_u and M_e, are crucial to our framework.

More Discussion

Different visual backbones. We show the average performance of different visual backbones on CIFAR10-C with corruption level 3, such as RN50 and ViT-B/32. In Table 5, ADAPROMPT can also achieve performance improvement in different backbones.

Component		CIFAR10-C(s=3)	CIFAR10-C(s=5)
M_e	M_u		
		76.21 ± 0.00	62.37 ± 0.00
✓		75.38 ± 0.00	61.75 ± 0.00
	✓	77.72 ± 0.24	65.32 ± 0.18
✓	✓	**79.15 ± 0.23**	**66.12 ± 0.43**

Table 4: Ablation study of ADAPROMPT on CIFAR10-C dataset with corruption level 3 and 5. The average accuracy on 15 different domains is reported.

Acc(%)	Source	TPT	Ours
RN50	47.70 ±0.00	51.44 ±0.02	**55.44 ± 0.30**
ViT-B/32	71.30 ±0.00	73.77 ±0.03	**75.81 ± 0.33**

Table 5: Average accuracy of CIFAR10-C in different 15 domains with corruption level 3 on different backbones.

Running time consumption. We explore the consumption of running time for different methods. In Table 6, it can be seen that ADAPROMPT consumes much less time than TPT. When using CLIP model with longer inference time than traditional models (such as CNN) for predictions, the running time may also be a factor to consider.

Results on TinyImageNet-C and ImageNet-R. From Table 3, we present the performance of ADAPROMPT on TinyImageNet-C with corruption level 3, which contains 200 prediction classes and 15 different domains. We can see that ADAPROMPT achieves optimal performance on 15 different domains and achieves 2.22% average accuracy improvement compared to the SOTA method TPT. The results on ImageNet-R, which contains 200 prediction classes and 30000 testing images with artistic renditions, are shown in Table 6. Although ADAPROMPT do not outperform TPT, TPT uses 63 random data augmentations for each image, which greatly consumes time and storage costs. Moreover, to make a more holistic comparison with TPT, we present accuracy and time cost on other datasets in the appendix.

The effect of confidence selection. We present confidence threshold as a component of ADAPROMPT, which is used to select confident samples pushed into the buffer. In Figure 6, we provide the performance at different confidence thresholds on CIFAR100-C with corruption level 3. We can see that different thresholds have little impact on performance.

The trade-off between storage cost and accuracy. We analyze the impact of buffer size on performance. In Figure 5, we show the average performance of 15 domains in the CIFAR100-C with corruption level 3. When the capacity of buffer increases, more samples are used for updating, resulting in better performance and more storage cost.

Related Work

Test-time Adaptation. Test-time adaptation (Zhou et al. 2023a,b, 2021; Zhou, Jin, and Li 2024) aims to adapt a

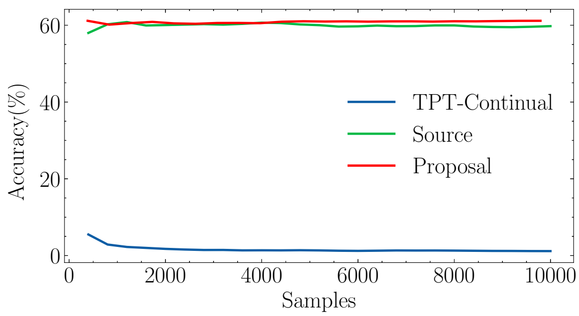

Figure 4: Comparison with three different methods in CIFAR100-C contrast domain with corruption level 3.

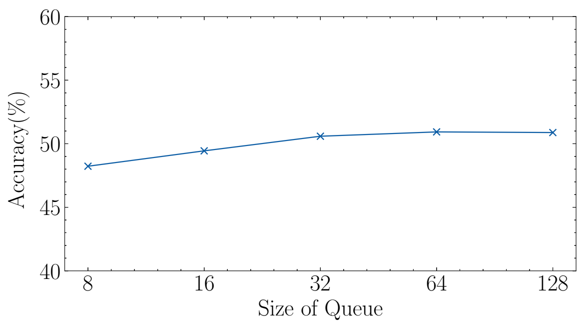

Figure 5: Average accuracy of ADAPROMPT with different buffer size on CIFAR100-C with corruption level 3.

source model to the distribution shift in testing data without using any source data. TENT (Wang et al. 2021) introduces entropy minimization to update the BN (Ioffe and Szegedy 2015) layers at test time. EATA (Niu et al. 2022) additionally proposes the sample selection and weighting strategies for efficiency. NOTE (Gong et al. 2022) adopts instance-aware batch normalization and prediction-balanced reservoir sampling to ensure robustness under non-i.i.d. scenarios. However, for these TTA methods that update the BN layer parameters of the model, the vision-language model uses LN (Xu et al. 2019) instead of BN. CoTTA (Wang et al. 2022), another way to update model parameters, adopts the weight-averaged model, augmentation-averaged prediction, and stochastically restores to enable the continual adaptation ability in changing environments, which updates all parameters of the model. However, the vision language model has a large number of parameters, and updating the entire model is time-consuming and may not necessarily improve performance due to the small size of the dataset. Therefore, traditional TTA methods cannot be directly transferred to vision-language models. In this work, we propose test-time prompt tuning that works on a single domain in the vision-language model. Our work does not involve any training process and can directly work with the zero-shot classification.

Prompt tuning. Prompt tuning (Hossain et al. 2021; Li and Liang 2021) is first proposed in natural language processing(NLP), hoping to adapt pre-trained visual-language models to various downstream tasks. Recently, the

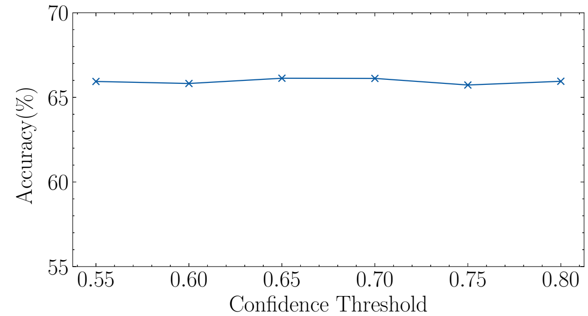

Figure 6: Performance of ADAPROMPT with different confidence threshold on CIFAR100-C with corruption level 3.

Dataset	Metrics	Source	TPT	Ours
CIFAR10-C	Acc(%)	62.37	63.81	66.12
	Time cost(s)	393.15	41257.35	2143.8
ImageNet-R	Acc(%)	70.86	74.19	73.98
	Time cost(s)	98.11	9875.10	531.30

Table 6: The time consumption and accuracy in CIFAR10-C with corruption level 5 and ImageNet-R with ViT-B/16.

idea of prompt has been transferred to some multi-modal tasks. CoOp (Zhou et al. 2022b) applies prompt tuning to CLIP (Radford et al. 2021), which proposes to use contrastive loss to pull together images and their textual descriptions while pushing away unmatched pairs in the feature space. CoOp effectively improves CLIP's performance on the corresponding downstream tasks by tuning the prompt on a collection of training data. However, the learning of these prompts requires training data, which may not be available in practical situations. Recently, TPT (Shu et al. 2022) proposes test-time prompt tuning that works on a single test sample. However, TPT encounters performance degradation on certain domains and requires a significant time cost for data augmentation. Our paper focuses on solving the problem of performance degradation and alleviating *Model Bias* and *Data Bias*, which further adapts the model to the current data at a small time cost.

Conclusion

In this paper, we study the problems of zero-shot classification based on the pre-trained CLIP model. We show that existing methods suffer from two fundamental issues: *Data Bias* and *Model Bias*. These issues significantly weaken the robustness of existing methods and lead to performance degradation problems. Therefore, we propose robust test-time adaptation for zero-shot prompt tuning. For *Data Bias*, we ensemble multiple hand-crafted prompts and fine-tune all prompts with unlabeled test data. For *Model Bias*, we store high-confidence, class-balanced samples in a confidence-aware buffer, which ensures robust updates in a balanced and confident way. Extensive experiments on multiple benchmark datasets demonstrate our method mostly achieves SOTA performance at a small time cost.

Acknowledgements

This research was supported by the National Key R&D Program of China (2022ZD0114803), the National Science Foundation of China (62176118, 61921006).

References

Boudiaf, M.; Mueller, R.; Ben Ayed, I.; and Bertinetto, L. 2022. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8344–8353.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, 1877–1901.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 9th International Conference on Learning Representations*.

Gong, T.; Jeong, J.; Kim, T.; Kim, Y.; Shin, J.; and Lee, S.-J. 2022. NOTE: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*, 27253–27266.

Guo, L.-Z.; and Li, Y.-F. 2024. Robust Pseudo-Label Selection for Holistic Semi-Supervised Learning. *Science CHINA Information Science*.

Guo, L.-Z.; Zhou, Z.; and Li, Y.-F. 2020. Record: Resource constrained semi-supervised learning under distribution shift. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1636–1644.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proceedings of the 7th International Conference on Learning Representations*.

Hossain, M. A.; Yin, D.; Gao, X.; Zhang, J.; Shen, S.; Guo, H.; Tang, J.; and Xu, Y. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 5963–5977.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 448–456.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 4904–4916.

Jia, L.-H.; Guo, L.-Z.; Zhou, Z.; and Li, Y.-F. 2024. LAMDA-SSL: a comprehensive semi-supervised learning toolkit. *Science CHINA Information Science*, 67.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 4582–4597.

Niu, S.; Wu, J.; Zhang, Y.; Chen, Y.; Zheng, S.; Zhao, P.; and Tan, M. 2022. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, 16888–16905.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 8748–8763.

Shanmugam, D.; Blalock, D. W.; Balakrishnan, G.; and Guttag, J. V. 2021. Better Aggregation in Test-Time Augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1194–1203.

Shi, J.-X.; Wei, T.; and Li, Y.-F. 2024. Residual Diverse Ensemble for Long-Tailed Multi-Label Text Classification. *Science CHINA Information Science*.

Shu, M.; Nie, W.; Huang, D.-A.; Yu, Z.; Goldstein, T.; Anandkumar, A.; and Xiao, C. 2022. Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models. In *Advances in Neural Information Processing Systems*, volume 35, 14274–14289.

Tian, Q.; Sun, H.-Y.; Peng, S.; and Ma, T.-H. 2023. Self-adaptive label filtering learning for unsupervised domain adaptation. *Frontiers of Computer Science*, 17.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *Proceedings of the 9th International Conference on Learning Representations*.

Wang, Q.; Fink, O.; Van Gool, L.; and Dai, D. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7201–7211.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models are Zero-Shot Learners. In *Proceedings of the 10th International Conference on Learning Representations*.

Xu, J.; Sun, X.; Zhang, Z.; Zhao, G.; and Lin, J. 2019. Understanding and Improving Layer Normalization. In *Advances in Neural Information Processing Systems*, 4383–4393.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

Zhou, Z.; Guo, L.-Z.; Cheng, Z.-Z.; Li, Y.-F.; and Pu, S.-L. 2021. STEP: Out-of-Distribution Detection in the Presence of Limited In-Distribution Labeled Data. In *Advances in Neural Information Processing Systems*, 29168–29180.

Zhou, Z.; Guo, L.-Z.; Jia, L.-H.; Zhang, D.-C.; and Li, Y.-F. 2023a. ODS: test-time adaptation in the presence of open-world data shift. In *Proceedings of the 40th International Conference on Machine Learning*, 42574–42588.

Zhou, Z.; Jin, Y.-X.; and Li, Y.-F. 2024. Rts: Learning Robustly from Time Series Data with Noisy Label. *Frontiers of Computer Science*, 18.

Zhou, Z.; Zhang, D.-C.; Li, Y.-F.; and Zhang, M.-L. 2023b. Towards Robust Test-Time Adaptation for Open-Set Recognition. *Journal of Software*, 35(4).

Zhu, Y.; Wu, X.-D.; Qiang, J.-P.; Yuan, Y.-H.; and Li, Y. 2023. Representation learning via an integrated autoencoder for unsupervised domain adaptation. *Frontiers of Computer Science*, 17.