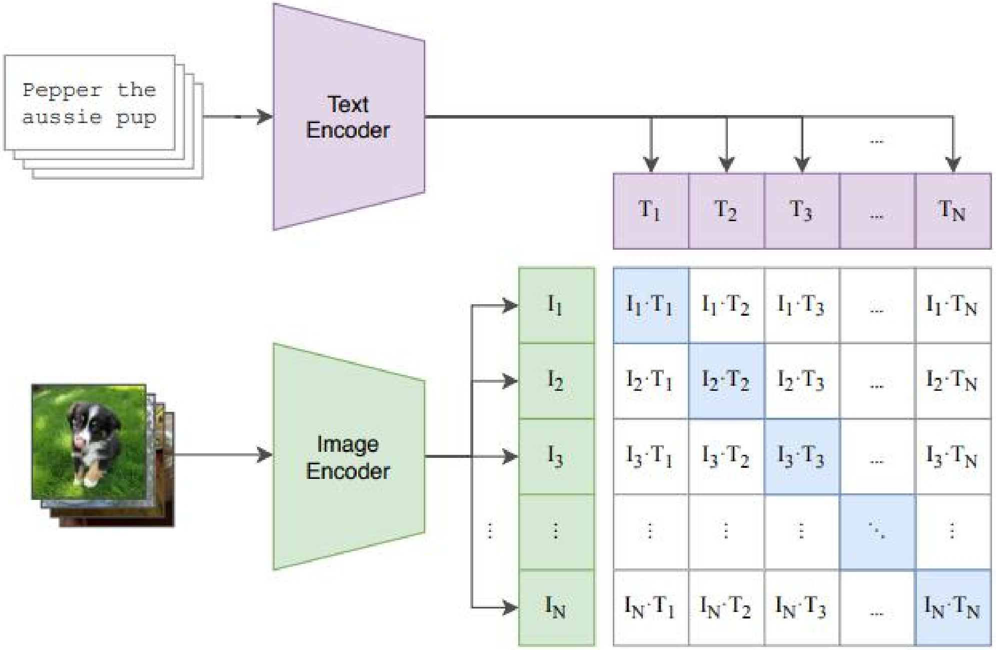
Robust Test-Time Adaptation for Zero-Shot Prompt Tuning

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

Background

- CLIP aligns visual and textual contents within a common feature space through training with millions of noisy image-text pairs and has demonstrated remarkable generalization across diverse downstream tasks [1].
- However, the appropriate prompt, which is challenging to choose in practical applications, plays a crucial role in downstream tasks.

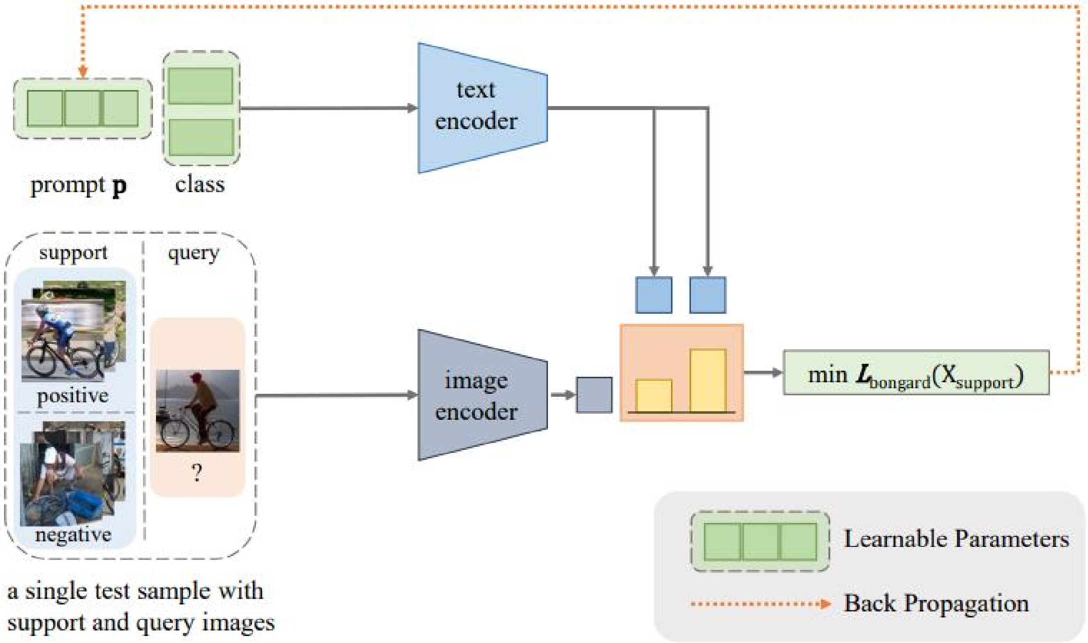


	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

[1] 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.

Background

- Prompt tuning, a method that optimizes the prompt by using data from downstream tasks, is an effective way to tackle selection problems.
- Different from using training data, recent studies[1] propose to fine-tune the prompt by using unlabeled test data to reduce human labeling pressure.
- However, they encounter performance degradation on certain domains and too much data augmentation leads to a high time cost.



[1] 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.

Background

We demonstrate that existing problems are caused by two biases, Data Bias and Model Bias.

I. Data bias: It is difficult to select an optimal prompt for some downstream task.

II. Model Bias: Prediction biases lead to error accumulation and will finally result in performance degradation.

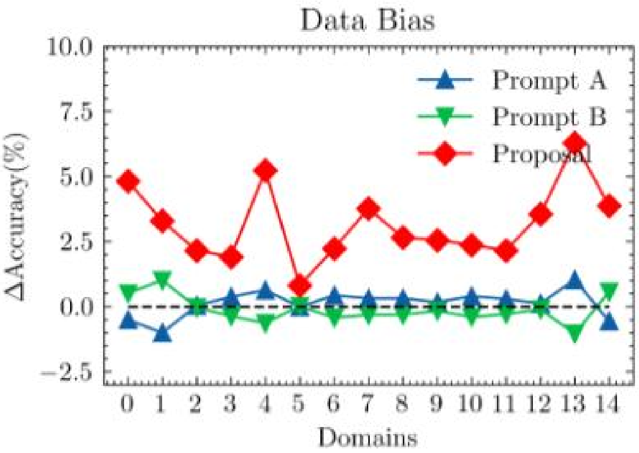


Figure 1: Relative performance compared to the average performance of prompts evaluated on CIFAR10-C in 15 domains with corruption level 3 using different initial prompts.

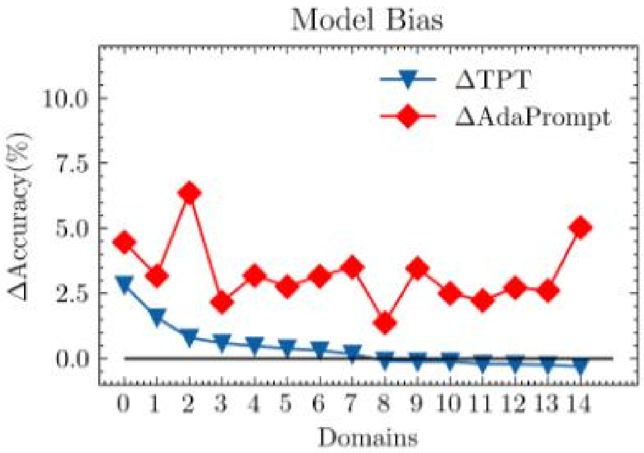


Figure 2: Relative performance compared to baseline evaluated on CIFAR10-C in 15 different domains with corruption level 3. The black line represents the baseline.

Method

Prompt Ensembling

Because performance of different prompts can vary across domains. We use different hand-crafted prompts and ensemble their predictions to alleviate the negative effects of Data Bias and avoid the worst-case results.

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

Based on ensembling results above, we can obtain pseudo label and confidence for each sample, which can help us select confident samples to update the model's prompts.

$$\hat{y}(\mathbf{x}_t) = argmax_k \hat{f}(y_k|\mathbf{x}_t;\mathbf{p})$$
$$c(\mathbf{x}_t) = max_k \hat{f}(y_k|\mathbf{x}_t;\mathbf{p})$$

Method

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.

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

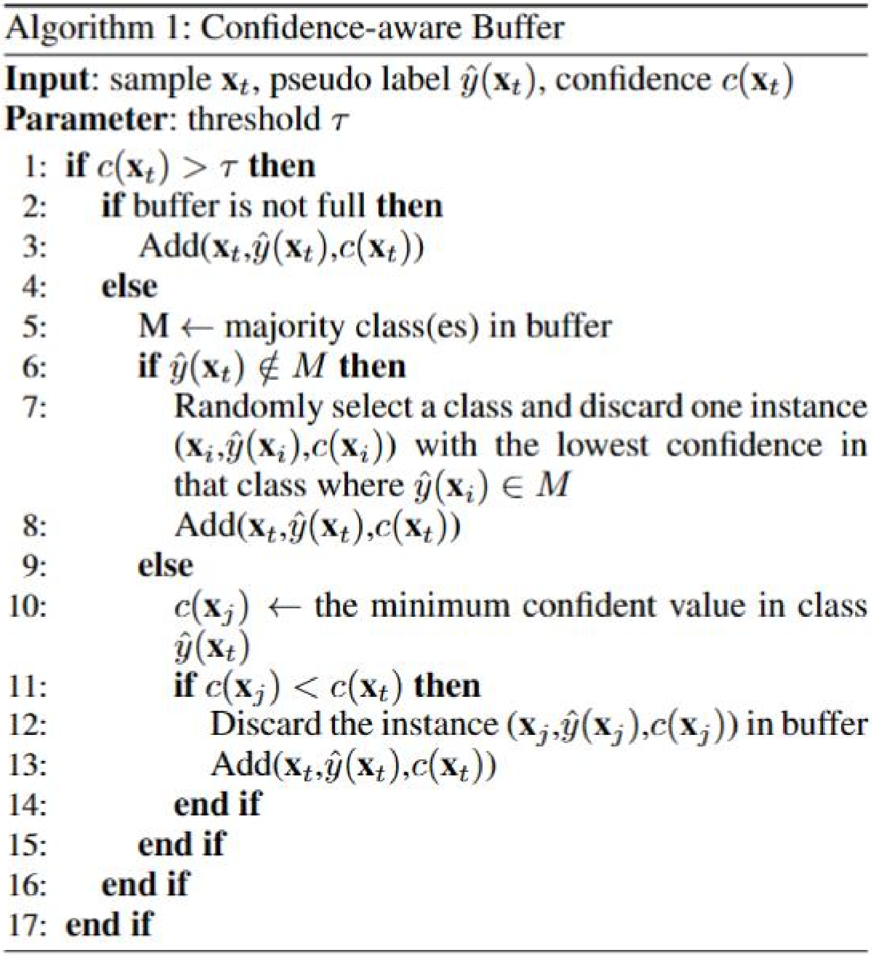
where we can obtain the pseudo label from the results of ensembling. The purpose of minimizing cross-entropy loss is to make the model more confident in the predicted samples, which can adapt prompts to Data Bias and improve the accuracy of predictions.

Method

Confidence-aware Buffer

To alleviate the problem of Model Bias, we use 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, we set confidence as priority of buffer, making it less likely to cause erroneous updates.
- For pseudo label balance, we count the number of samples in each class to ensure buffer balance.
- In addition, to ensure the accuracy of the samples entering the buffer, we use a threshold τ to filter out samples with low confidence.



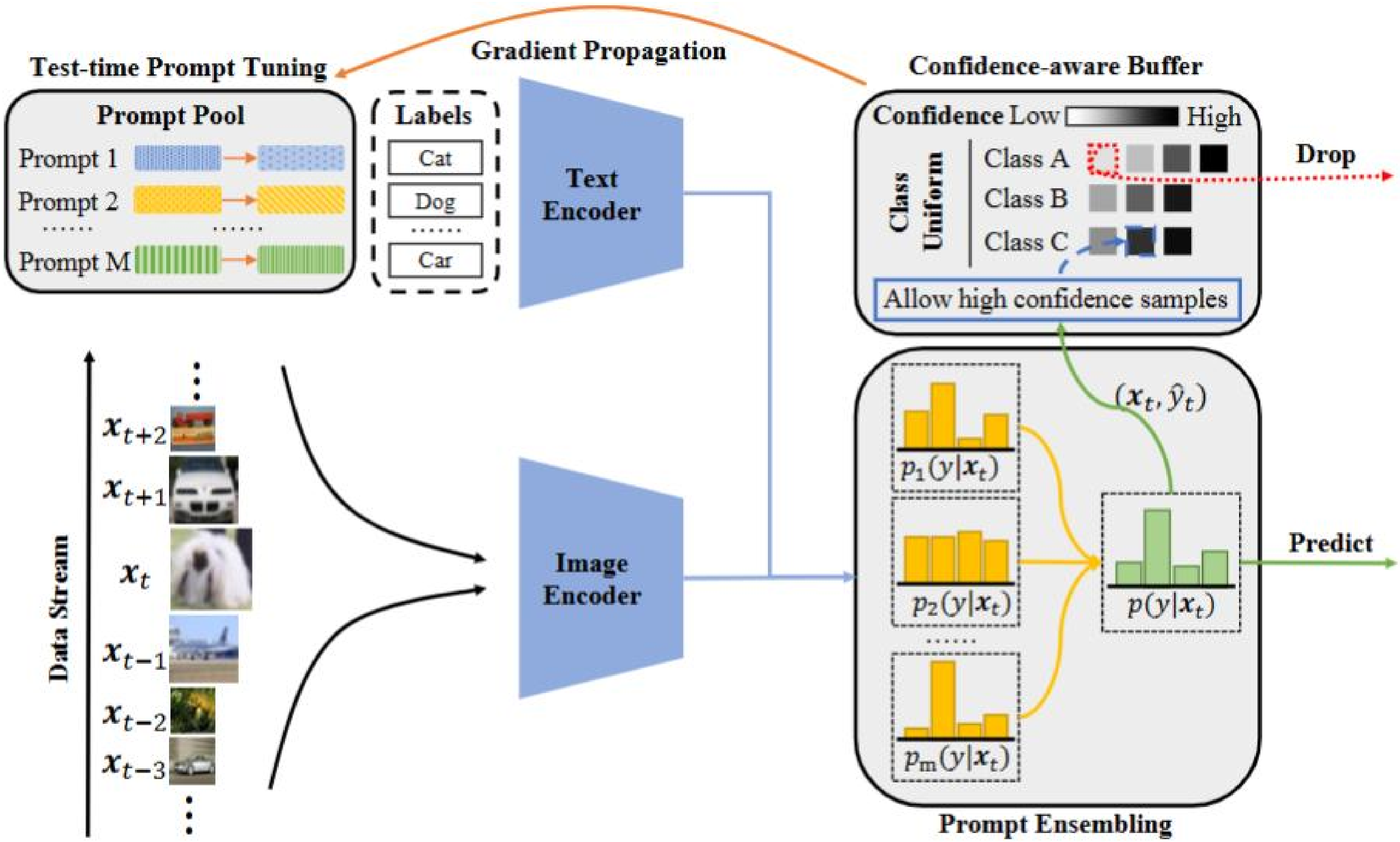
Algorithm 1: Confidence-aware Buffer

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 ← 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
```

Method

Overall Framework



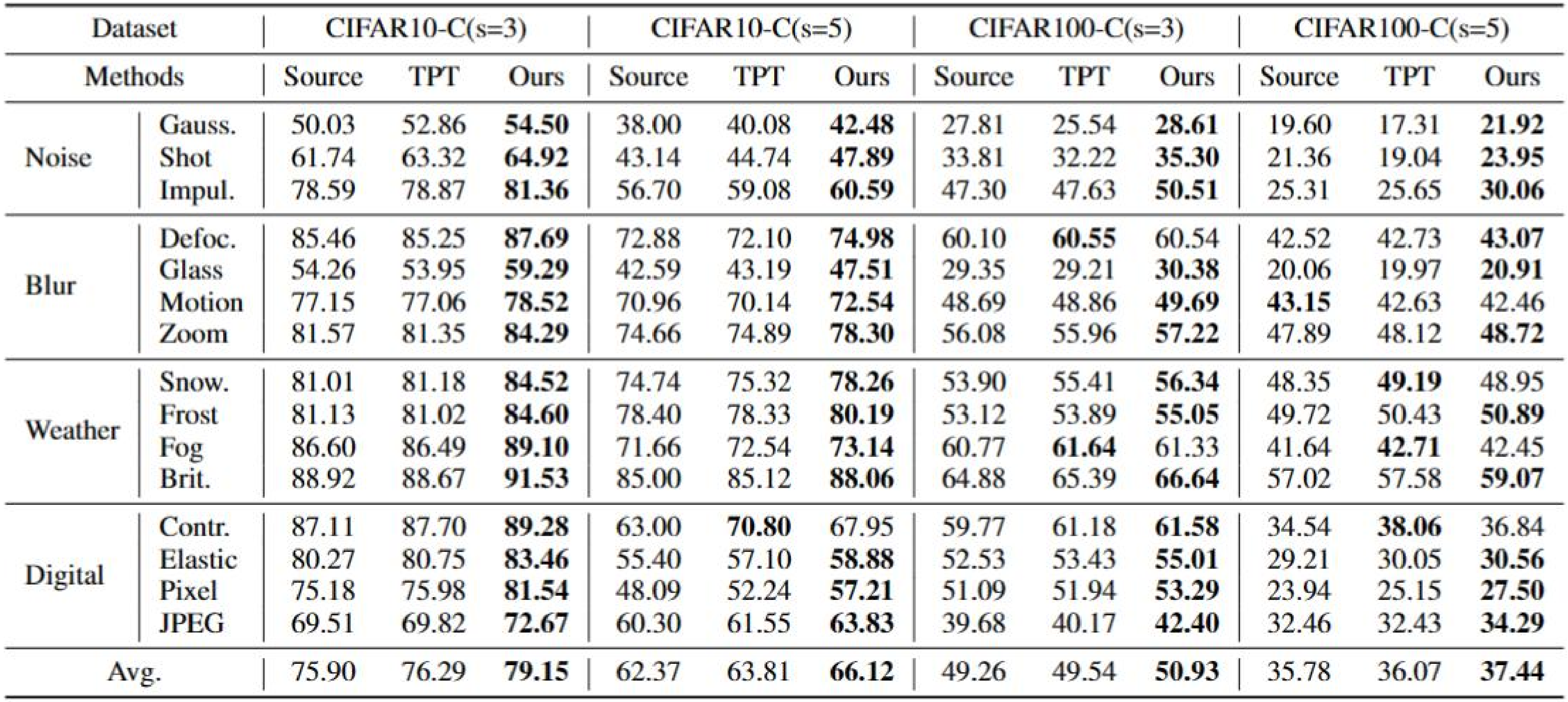
Experiments

We try to give answers to three 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?

Experiments

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



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.

Experiments

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

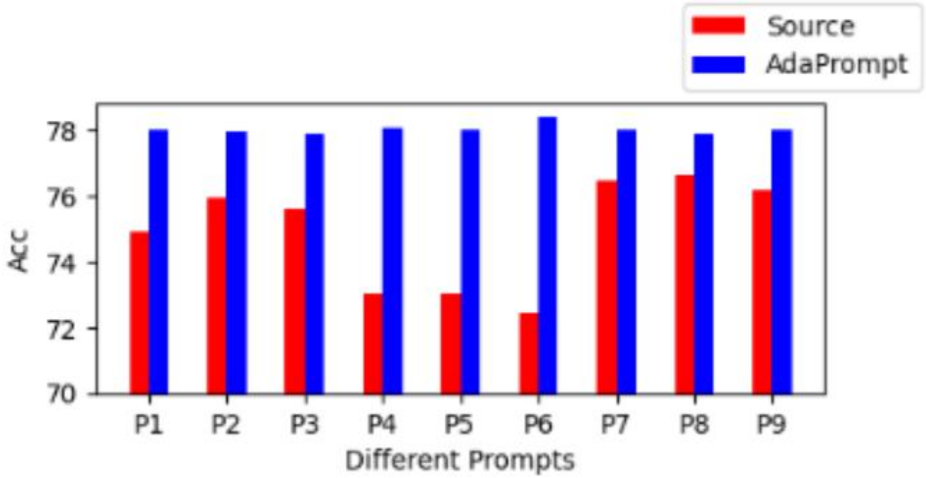
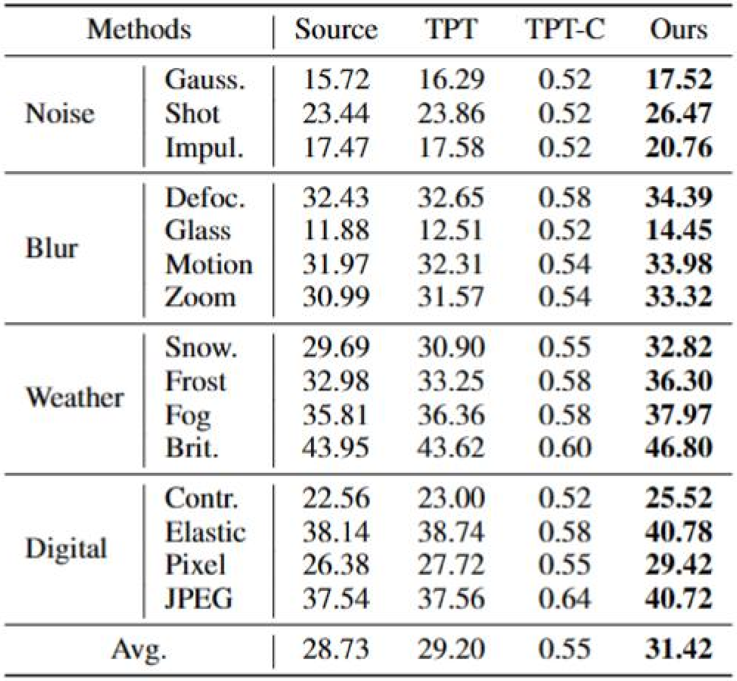


Figure 1: Average performance of different hand-crafted prompts on the CIFAR-10-C dataset.

Experiments

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



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.

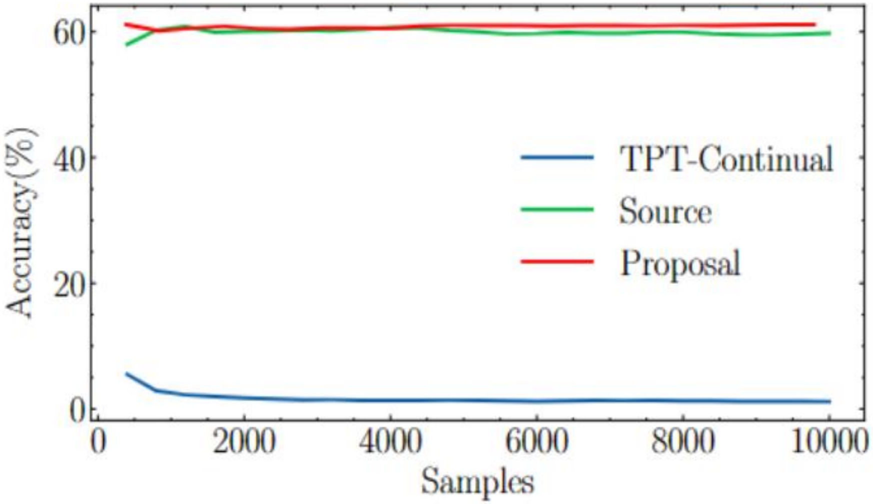
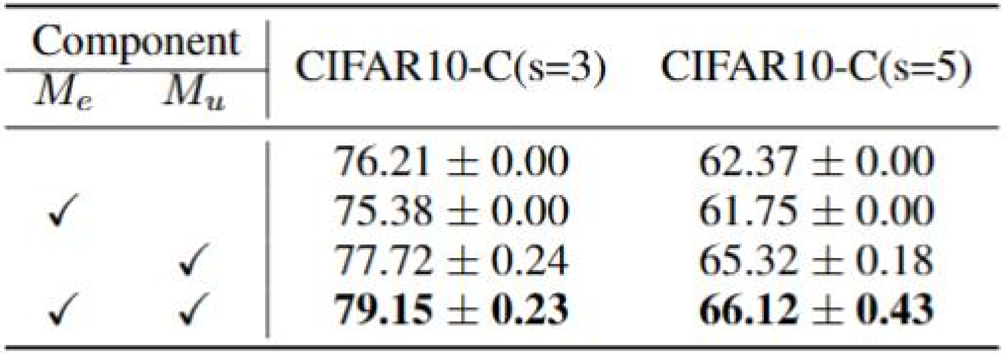


Figure 4: Comparison with three different methods in CIFAR100-C contrast domain with corruption level 3.

Experiments

Ablation studies show that updating multiple prompts together and then ensembling can adapt to current test data stream better, i.e., these two modules are crucial to our framework.

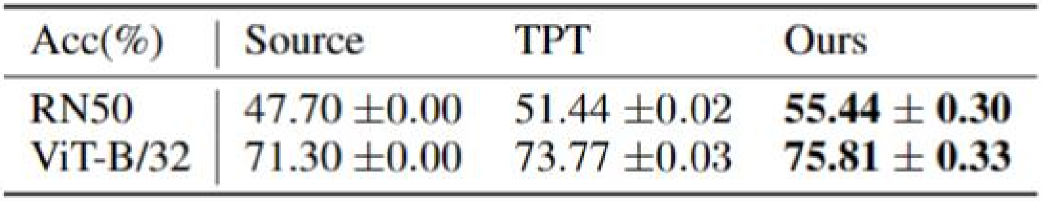


Component		CIFAR10-C(s=3)	CIFAR10-C(s=5)
M_e	M_u		
		76.21 ± 0.00	62.37 ± 0.00
✓		75.38 ± 0.00	61.75 ± 0.00
	✓	77.72 ± 0.24	65.32 ± 0.18
✓	✓	$\mathbf{79.15 \pm 0.23}$	$\mathbf{66.12 \pm 0.43}$

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.

More Discussion

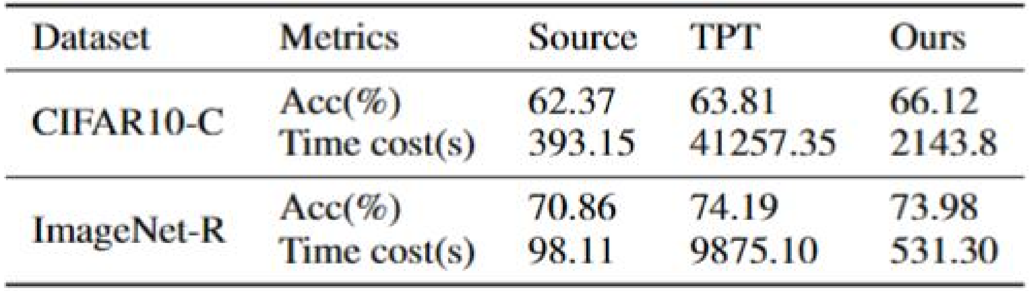
Different Visual Backbones



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



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.

More Discussion

Hyparameter Experiments

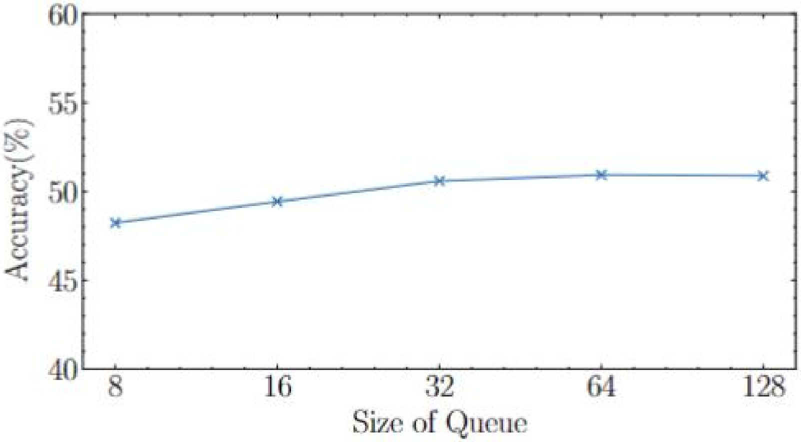


Figure 5: Average accuracy of ADAPROMPT with different buffer size on CIFAR100-C with corruption level 3.

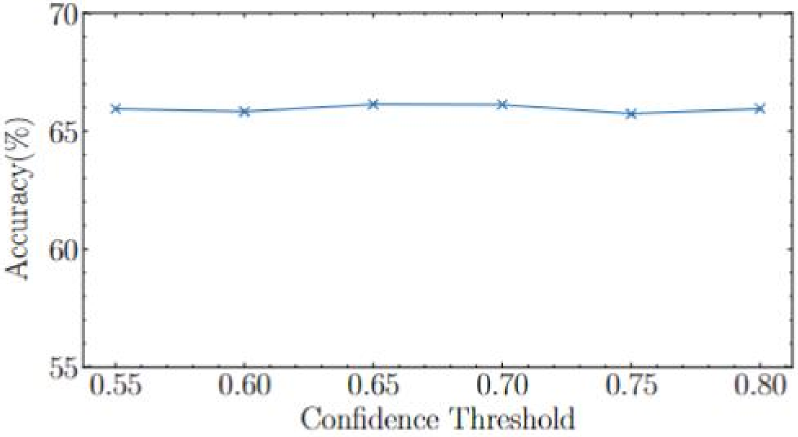


Figure 6: Performance of ADAPROMPT with different confidence threshold on CIFAR100-C with corruption level 3.

Thanks for listening !