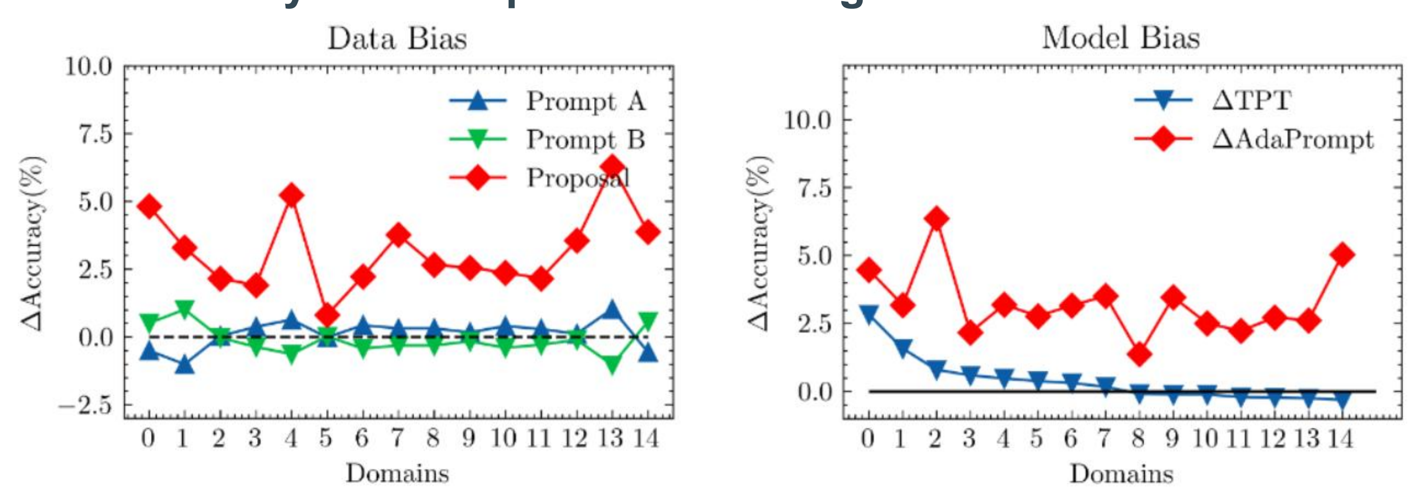
Robust Test-Time Adaptation for Zero-Shot Prompt Tuning

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

Motivation

Prompt tuning is a method that optimizes the prompt by using data from downstream tasks, which adapts CLIP models to various downstream task. However, prompt tuning without any training data will result in two issues, i.e. **data bias and model bias:**

- **Data bias: It is difficult to select an optimal prompt for some downstream task.**
- **Model Bias: Prediction biases lead to error accumulation and will finally result in performance degradation.**



I. We empirically analyze existing prompt tuning methods by using unlabeled test data and point out Data Bias and Model Bias.

II. We propose the test-time prompt tuning method ADAPROMPT, which effectively tackles the previously proposed Data Bias and Model Bias issues.

III. We evaluate our methods 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.

ADAPROMPT Method

Prompt Ensembling

We use different hand-crafted prompts and ensemble their predictions to alleviate negative effects of Data Bias and avoid worst-case results.

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

Test-time Prompt Tuning

We optimize all prompts using unlabeled test data to adapt prompts to Data Bias.

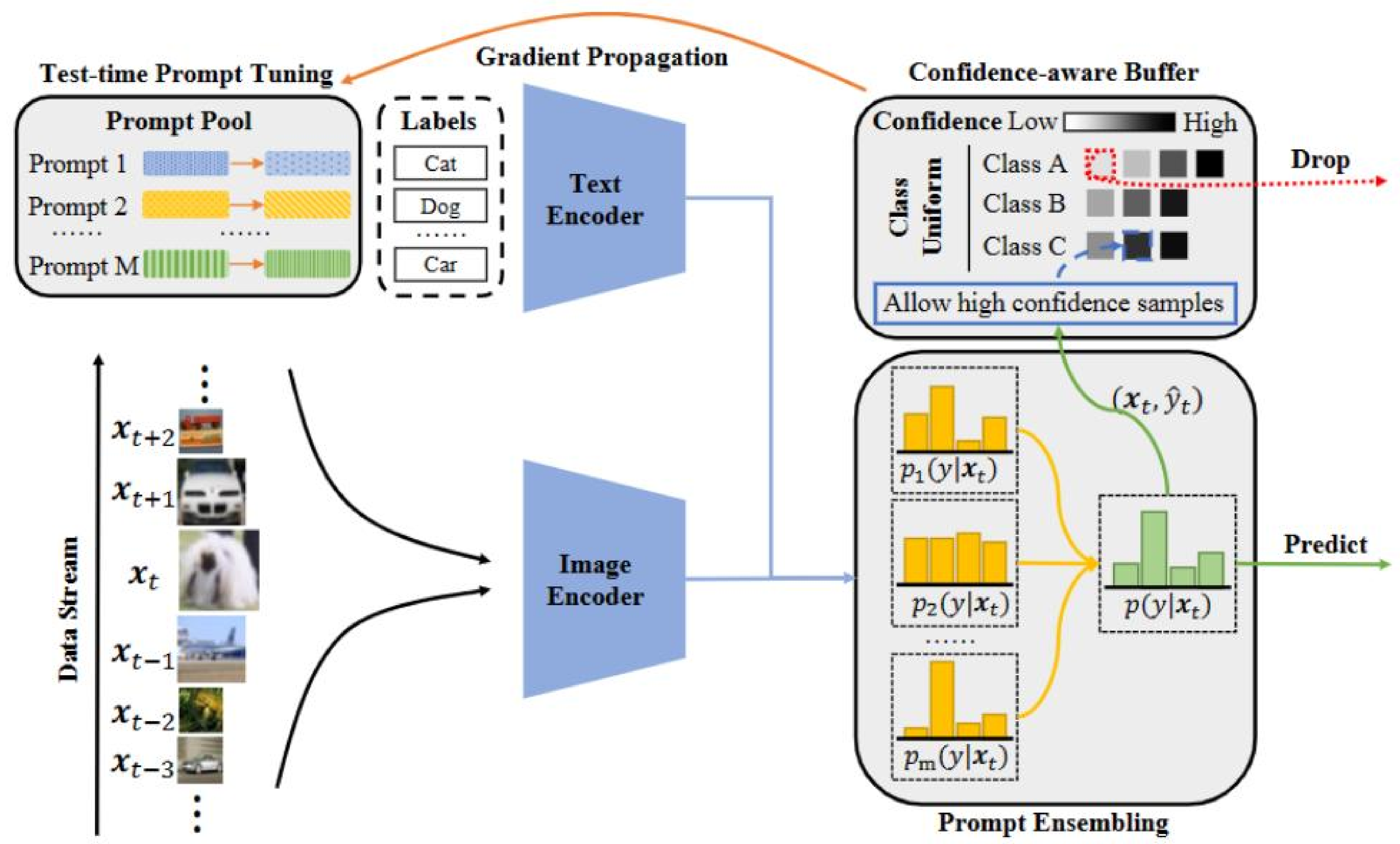
$$L(\mathbf{x}_t) = -\sum_{k=1}^{K} \hat{y}_k(\mathbf{x}_t) log\, \hat{f}(y_k|\mathbf{x}_t;\mathbf{p})$$

Confidence-aware Buffer

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.

Overall Framework

Firstly, we obtain the confidence via ensembling the probability of all prompts. Then we push some confident samples into buffer and extract all data from buffer to update the prompts. Finally, we obtain the outputs from the updated 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**

Detailed Results on CIFAR10-C and CIFAR100-C dataset with corruption level 3 and 5. We use CLIP model, whose visual model is vit-b16, as our backbone.

RQ2: Whether our proposed method alleviate Data Bias?

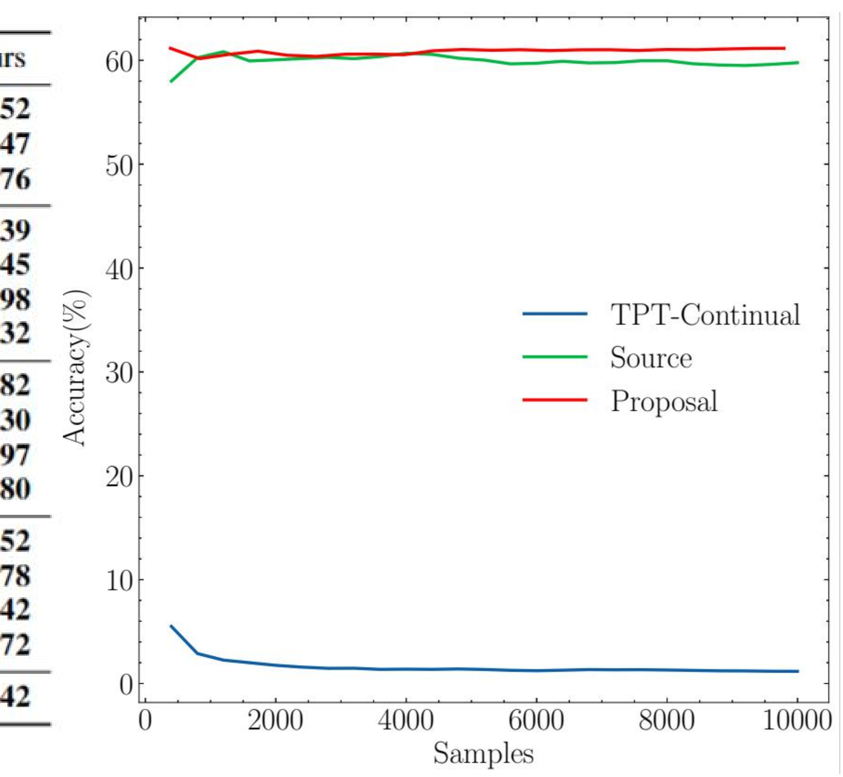
Method	CIFAR10-C(s=3)	CIFAR10-C(s=5)
P_A	75.91 ± 0.00	62.37 ± 0.00
P_B	76.21 ± 0.00	62.77 ± 0.00
P_C	72.98 ± 0.00	59.25 ± 0.00
P_{best} + UP.	77.72 ± 0.24	65.32 ± 0.18
P_e	75.38 ± 0.00	61.75 ± 0.00
P_e + UP.	$\mathbf{79.15 \pm 0.23}$	$\mathbf{66.12 \pm 0.43}$

Average results on CIFAR10-C with different prompts w/o updates

RQ3: Does ADAPROMPT relieve Model Bias?

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**

Detailed results on Tiny-ImageNet-C dataset with corruption level 3



Performance trend with increasing sample size in contrast domain.

Ablation Study

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

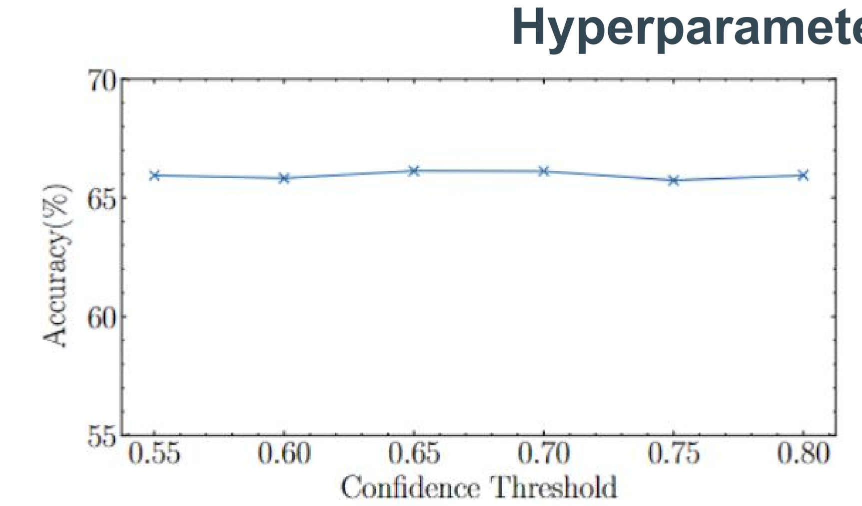
Effectiveness of each module in ADAPROMPT

Accuracy and Time Comparison

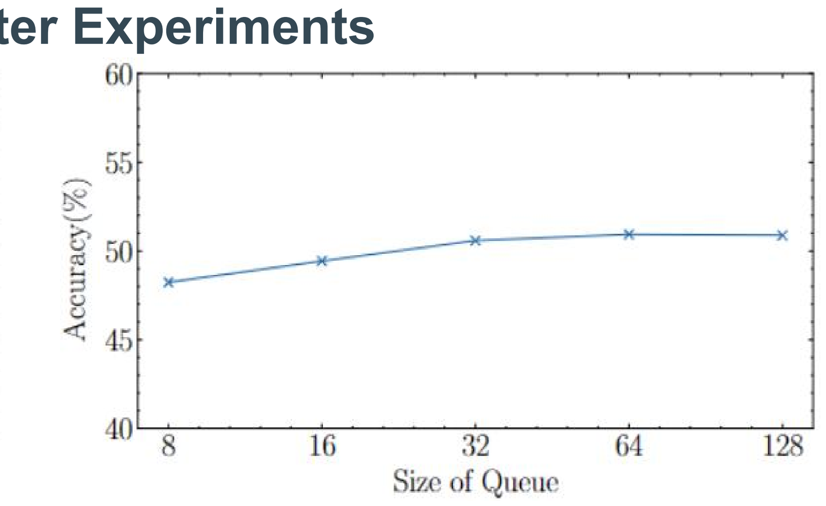
Dataset	Metrics	Source	TPT	Ours
CIFAR10-C	Acc(%)	62.37	63.81	66.12
	Time cost(s)	393.15	41257.35	2143.8
ImageNet-R	Acc(%)	70.86	74.19	73.98
	Time cost(s)	98.11	9875.10	531.30

Accuracy and time comparison in CIFAR10-C and ImageNet-R

Hyperparameter Experiments



Different confidence thresholds on CIFAR100-C dataset



Different queue sizes on CIFAR100-C dataset

*These authors contributed equally. †Corresponding author