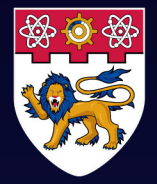


NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Test-Time Model Adaptation with Only Forward Passes

(Oral)

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, Peilin Zhao

Nanyang Technological University, Tencent AI Lab

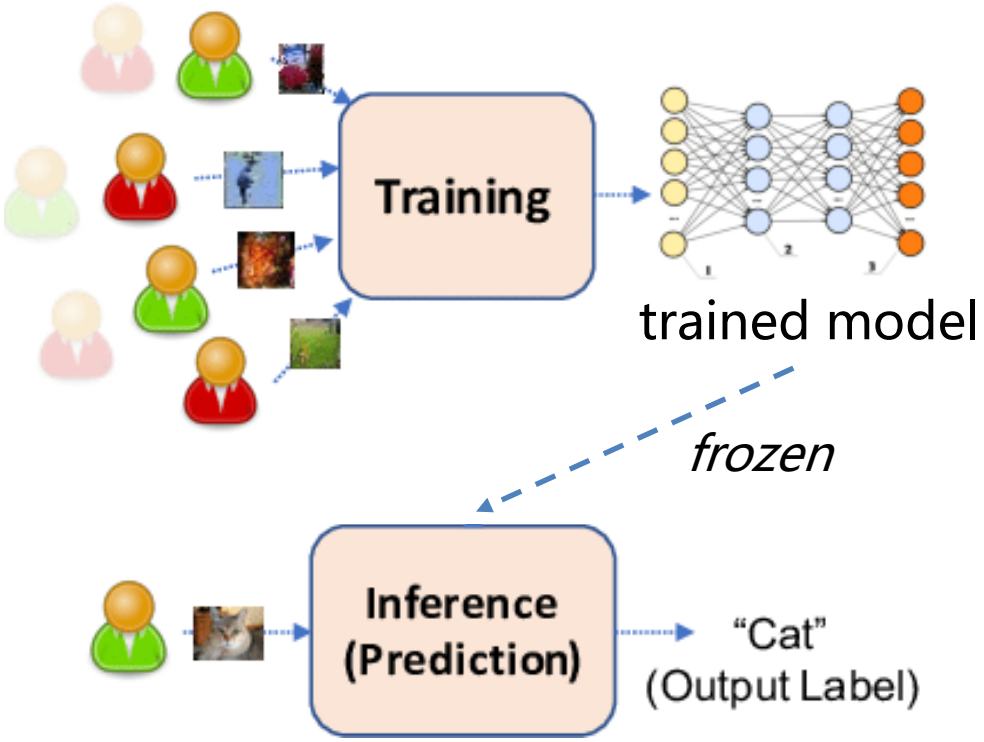
Presenter: Shuaicheng Niu (牛帅程)

International Conference on Machine Learning (ICML) 2024

AI TIME Artificial Intelligence

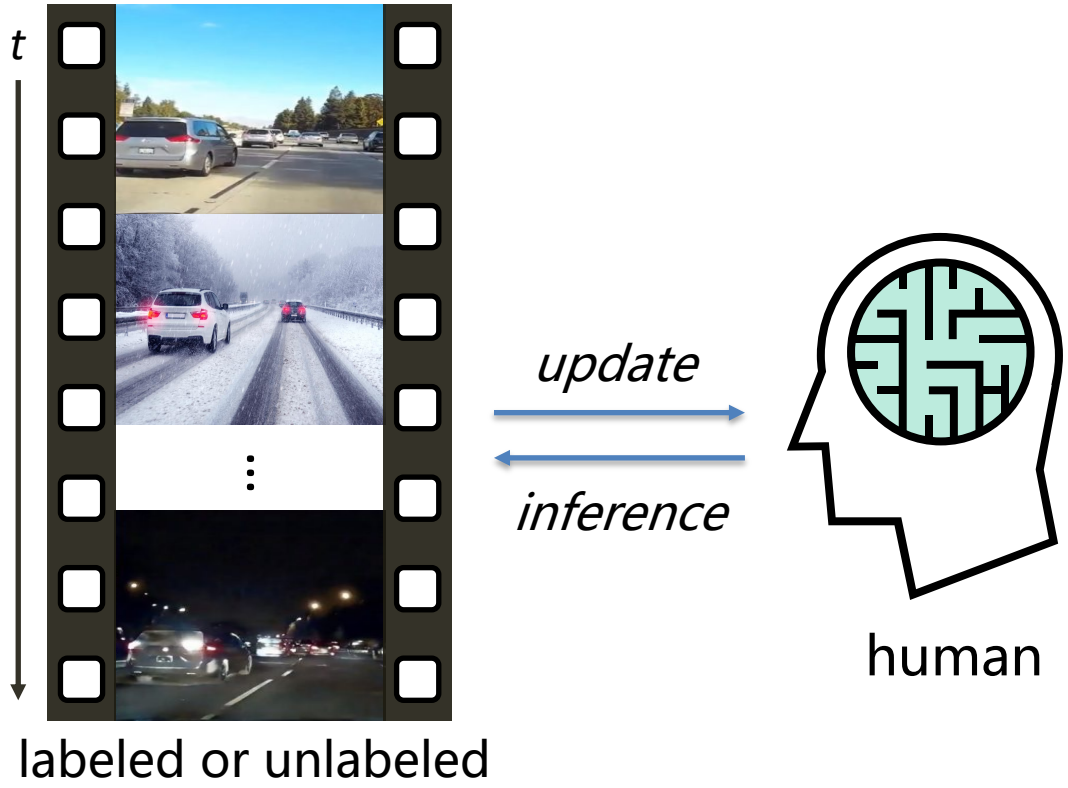
Background: Deep Learning Pipeline

- **conventional deep learning**



Inference with frozen knowledge

- **human intelligence**



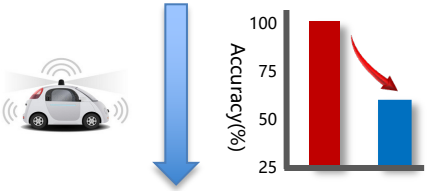
Inference with continuous learning

[1] Deep Learning on Private Data.

Background: Testing Data Shifts

sensor degradation

novel environment



weather change

Rain

Snow

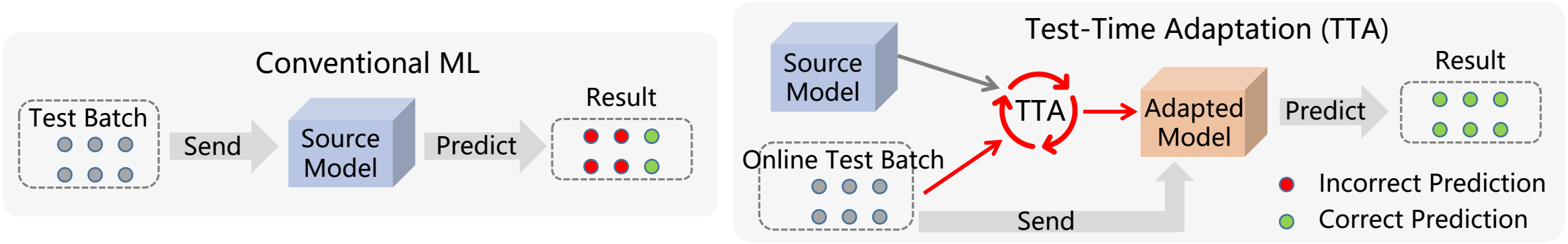
Fog

Models are very sensitive to such shifts, suffer from severe degradation!

ImageNet-C (Hendrycks et al., 2019)

Test-Time Adaptation for Overcoming Domain Shifts

- **Core idea:** (**online**) update model (**self/un-supervised**) before prediction



- **Differences** from prior test-time generalization methods

Setting	Source data	Target data	Training loss	Testing loss	Offline	Online
Fine-tuning[1]	×	x^t, y^t	$\mathcal{L}(\mathrm{x}^t, y^t)$	--	√	×
UDA[2]	x^s, y^s	x^t	$\mathcal{L}(\mathrm{x}^s, y^s) + \mathcal{L}(\mathrm{x}^s, \mathrm{x}^t)$	--	√	×
Test-time training[3] (ICML 20)	x^s, y^s	x^t	$\mathcal{L}(\mathrm{x}^s, y^s) + \mathcal{L}(\mathrm{x}^s)$	$\mathcal{L}(\mathrm{x}^t)$	×	√
Fully TTA[4] (ICLR 21)	×	x^t	×	$\mathcal{L}(\mathrm{x}^t)$	×	√

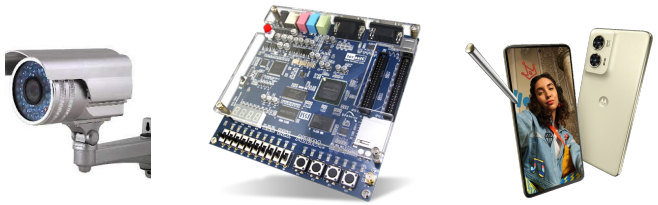
TTA

The Figures are borrowed from *Uncovering Adversarial Risks of Test-Time Adaptation.*

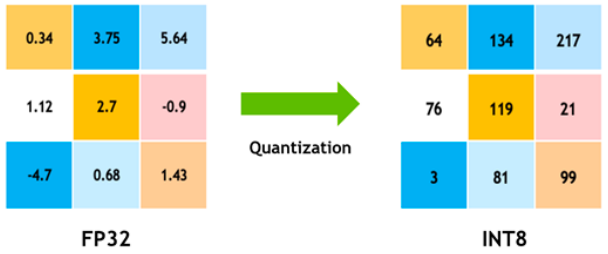
Limitations of Existing TTA

Learning-based TTA
- excellent performance
- rely on backpropagation (BP) at test time
- rely on weights updates for adaptation

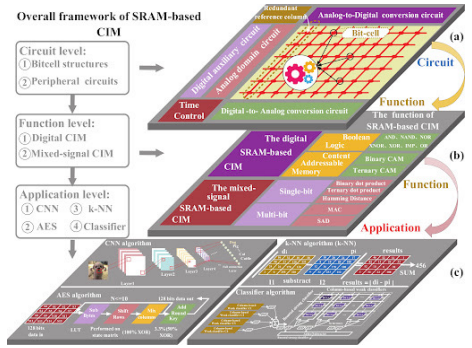
infeasible, hard to deploy 😭



resource-limited/latency-sensitive devices
(*power, memory*)



quantized deep model
(*vanishing gradients*)



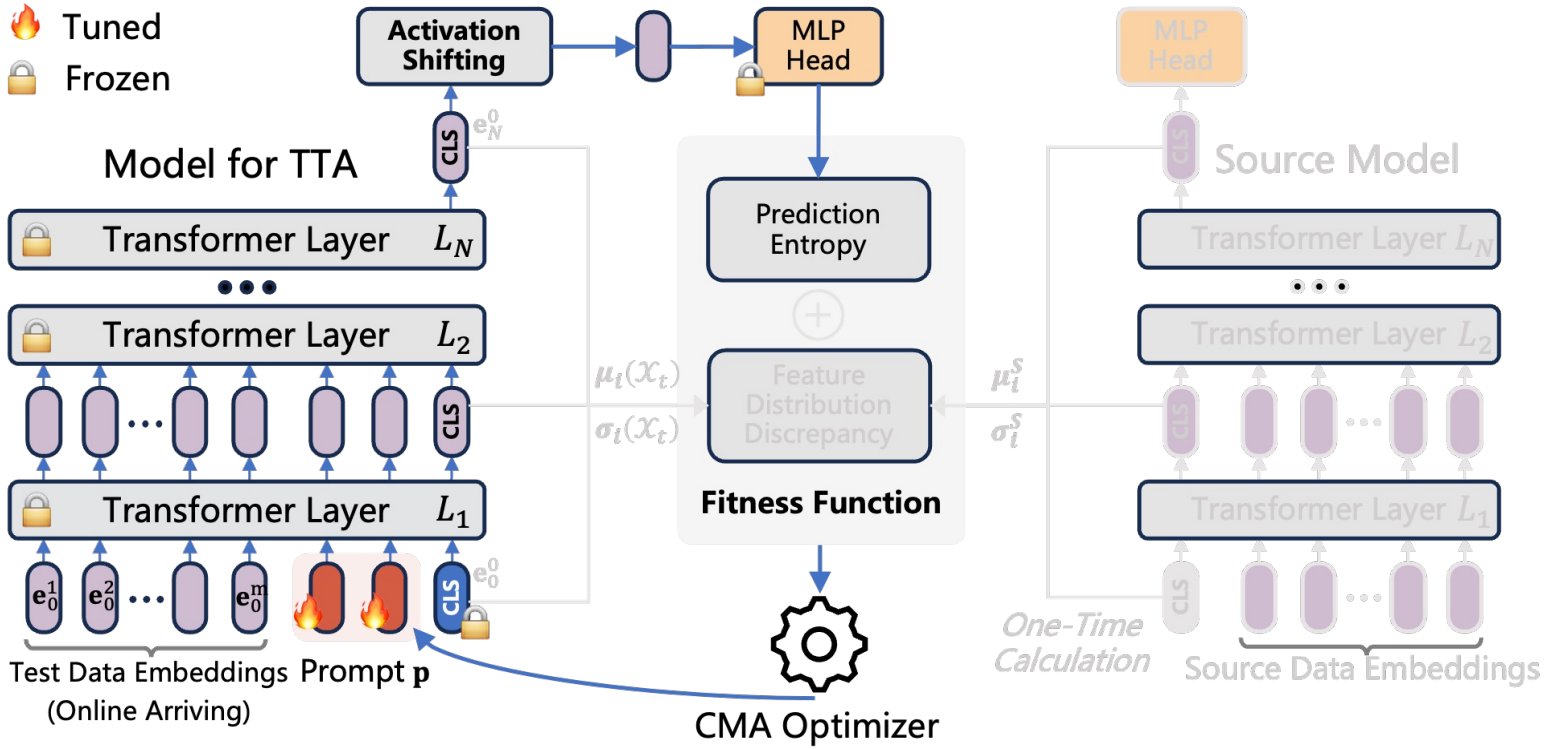
for acceleration, some chips are tailored for specific models with
(*non-modifiable weights*)

Key Motivation

How to **remove BP** and **avoids updating weights** during the TTA process,
while maintaining high adaptation performance?

IDEAs:
replace SGD/Adam with BP-free Optimizers
zeroth-order,
CMA: Covariance Matrix Adaptation (ours), …
reduce the dimension of solution space for easier optimization
input prompt adaptation

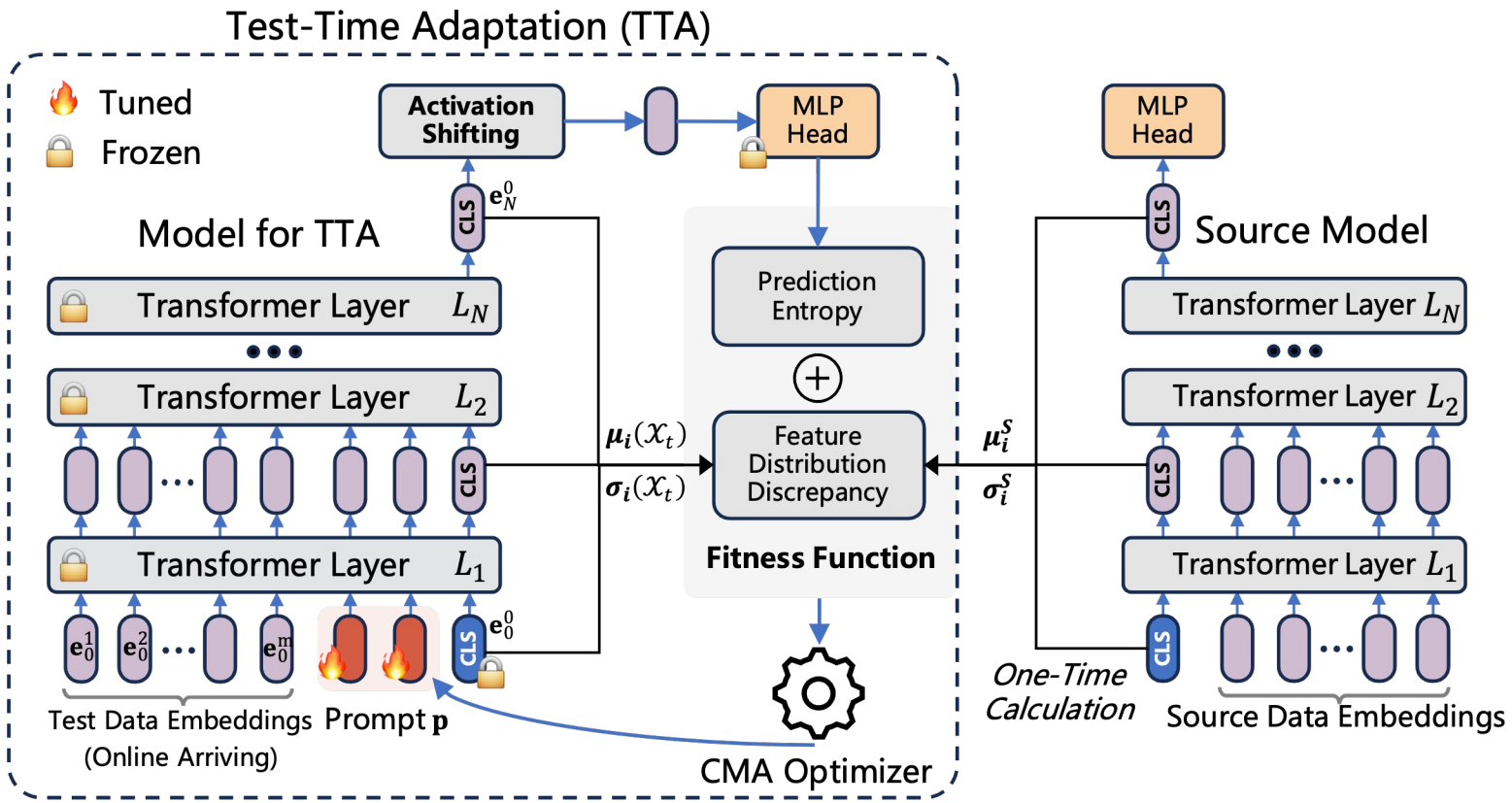
Forward-Only Prompt Adaptation



Taking only entropy as the fitness fails to stabilize the CMA learning, resulting **COLLAPSE**

Forward-Only Prompt Adaptation

Stabilize CMA learning via **Entropy + Feature Discrepancy Alignment**

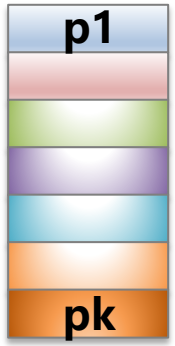


Note: Calculating source statistics only needs 64 unlabeled samples

Covariance Matrix Adaptation

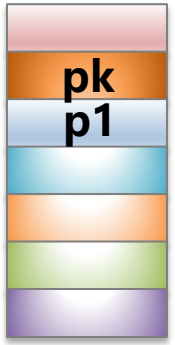
Instead of directly optimizing prompt **p**, CMA optimizes a distribution of **p**

$$\mathbf{p}_k^{(t)} \sim \mathbf{m}^{(t)} + \tau^{(t)}\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}^{(\mathbf{t})})$$



sampling *k* prompts

forward & calculate fitness



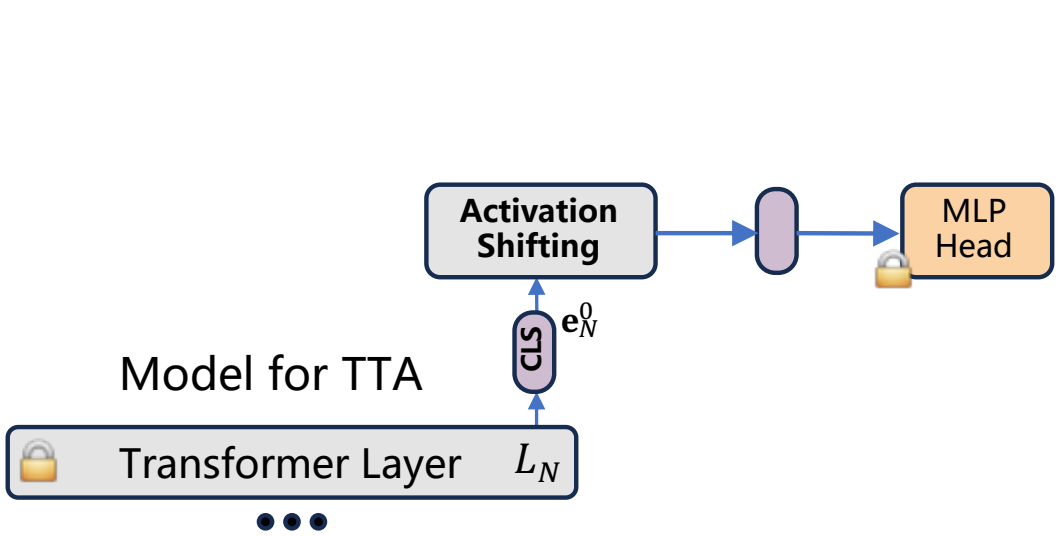
ranking prompts per fitness value

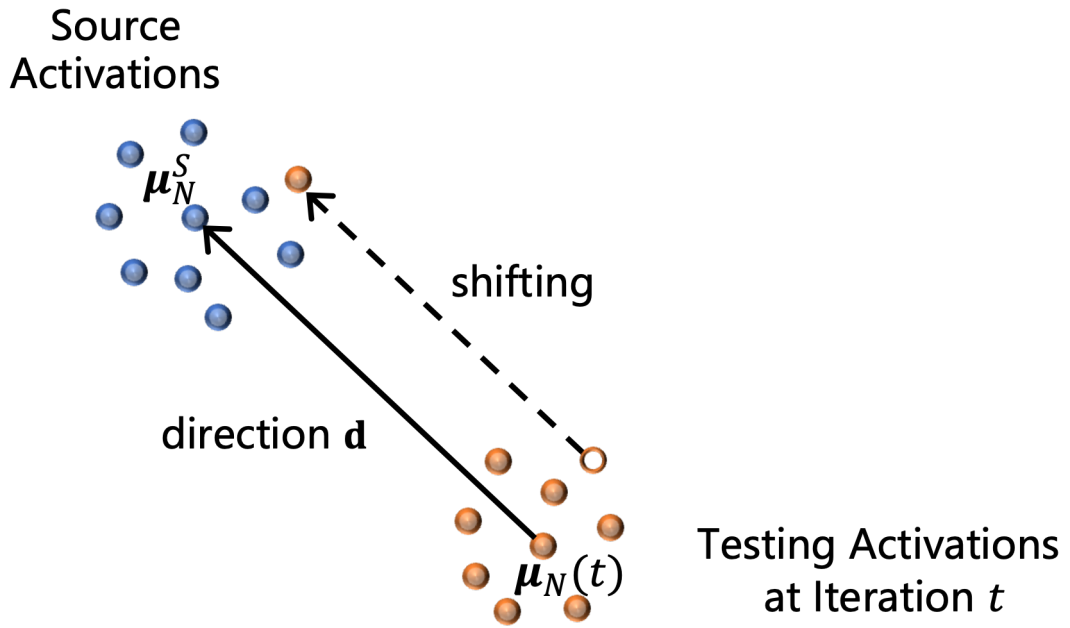
update **m** and **Σ** according to the ranking signals

Back-to-Source Activation Shifting

In cases of prompt adaptation is inadequate,

we directly modify the activations along the direction from OOD to ID





shifting: $\mathbf{e}_N^0 \leftarrow \mathbf{e}_N^0 + \gamma\mathbf{d}, \quad \mathbf{d}_t = \boldsymbol{\mu}_N^S - \boldsymbol{\mu}_N(t)$

target center (online updated): $\boldsymbol{\mu}_N(t) = \alpha\boldsymbol{\mu}_N(\mathcal{X}_t) + (1-\alpha)\boldsymbol{\mu}_N(t-1)$

Note: Calculating source activations center only needs 64 unlabeled samples

Comparisons with SOTA on Full-Precision Models

Table 2. Comparisons with SOTA methods on ImageNet-C (severity level 5) with ViT regarding **Accuracy (%)**. **BP** is short for **backward propagation** and the **bold** number indicates the best result. We only report average ECE (%,↓) here and put detailed ECEs in Appendix D.

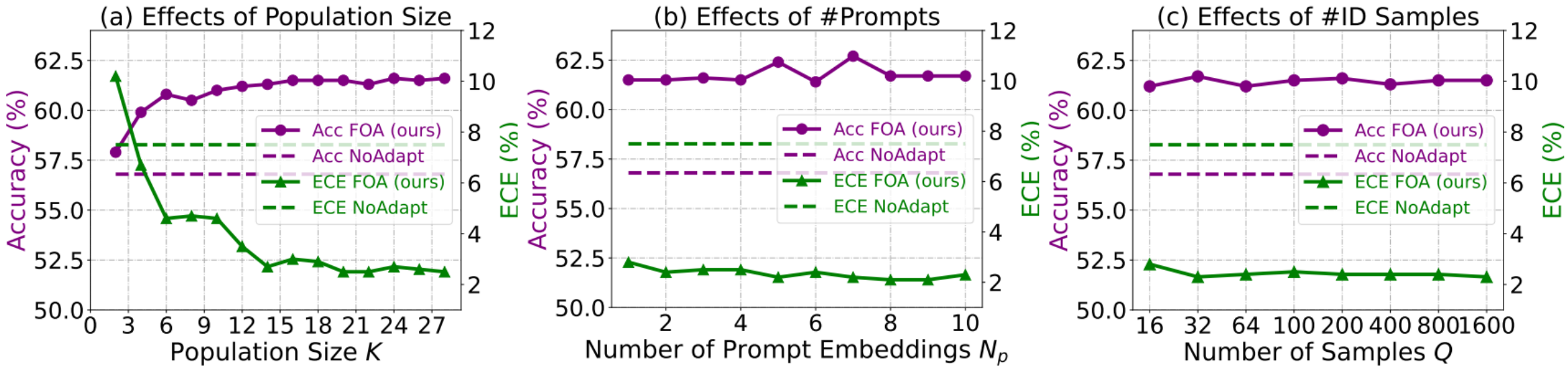
		Noise			Blur				Weather				Digital				Average	
Method	BP	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Acc.	ECE
NoAdapt	✗	56.8	56.8	57.5	46.9	35.6	53.1	44.8	62.2	62.5	65.7	77.7	32.6	46.0	67.0	67.6	55.5	10.5
LAME	✗	56.5	56.5	57.2	46.4	34.7	52.7	44.2	58.4	61.5	63.1	77.4	24.7	44.6	66.6	67.2	54.1	11.0
T3A	✗	56.4	56.9	57.3	47.9	37.8	54.3	46.9	63.6	60.8	68.5	78.1	38.3	50.0	67.6	69.1	56.9	26.8
TENT	✔	60.3	61.6	61.8	59.2	56.5	63.5	59.2	54.3	64.5	2.3	79.1	67.4	61.5	72.5	70.6	59.6	18.5
CoTTA	✔	63.6	63.8	64.1	55.5	51.1	63.6	55.5	70.0	69.4	71.5	78.5	9.7	64.5	73.4	71.2	61.7	6.5
SAR	✔	59.2	60.5	60.7	57.5	55.6	61.8	57.6	65.9	63.5	69.1	78.7	45.7	62.4	71.9	70.3	62.7	7.0
FOA (ours)	✗	61.5	63.2	63.3	59.3	56.7	61.4	57.7	69.4	69.6	73.4	81.1	67.7	62.7	73.9	73.0	**66.3**	**3.2**

Comparison on Quantized Models

Table 4. Effectiveness of our FOA on **Quantized ViT models**. We report the corruption **Accuracy** (%) and average ECE (%, ↓) on ImageNet-C (severity level 5). The **bold** number indicates the best result and see Appendix D for the detailed ECEs of each corruption.

		Noise			Blur				Weather				Digital				Average	
Model	Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Acc.	ECE
8-bit	NoAdapt	55.8	55.8	56.5	46.7	34.7	52.1	42.5	60.8	61.4	66.7	76.9	24.6	44.7	65.8	66.7	54.1	10.8
	T3A	55.6	55.7	55.7	45.8	34.4	51.1	41.2	59.5	61.9	66.8	76.4	45.5	43.4	65.6	67.5	55.1	25.9
	FOA (ours)	60.7	61.4	61.3	57.2	51.5	59.4	51.3	68.0	67.3	72.4	80.3	63.2	57.0	72.0	69.8	**63.5**	**3.8**
6-bit	NoAdapt	44.2	42.0	44.8	39.8	28.9	43.4	34.7	53.2	59.8	59.0	75.1	27.4	39.0	59.1	65.3	47.7	9.9
	T3A	43.3	41.3	42.7	29.1	23.4	38.9	30.0	49.4	58.3	60.2	73.8	31.0	36.3	58.0	65.2	45.4	30.1
	FOA (ours)	53.2	51.8	54.6	49.6	38.8	51.0	44.8	60.3	65.0	68.8	76.7	39.5	46.6	67.3	68.6	**55.8**	**5.5**

Hyper-parameter Sensitivity



Effects of Components and Design Choices

Table 5. Ablations of components in our FOA. *Entropy* and *Activation (Act.) Discrepancy* are the left/right item in Fitness Function (Eqn. 5) used for CMA-based prompt adaptation. *Act. Shifting* is the method proposed in Section 3.2. We report the average results over 15 corruptions on ImageNet-C (level 5) with ViT-Base.

Entropy	*Act. Discrepancy*	*Act. Shifting*	Acc. (%, ↑)	ECE (%,↓)
	NoAdapt		55.5	10.5
✔			44.9	36.8
	✔		63.4	9.4
		✔	59.1	12.7
	✔	✔	63.8	9.9
✔	✔		65.4	3.3
✔	✔	✔	**66.3**	**3.2**

all components work well

Table 9. Empirical studies of design choices w.r.t. learnable parameters, optimizer and loss function. We report the average results over 15 corruptions on ImageNet-C (level 5) with ViT-Base.

	Learnable Params	Optimizer	Loss	Acc. (↑)	ECE (↓)
NoAdapt	–	–	–	55.5	10.5
TENT	norm layers	SGD	entropy	59.6	18.5
exp1	prompts	SGD	entropy	50.7	18.4
exp2	norm layers	SGD	Eqn. (5)	70.5	7.9
exp3	prompts	SGD	Eqn. (5)	64.6	3.7
exp4	norm layers	CMA	Eqn. (5)	0.1	5.8
exp5	norm layers	CMA	entropy	0.1	99.0
exp6	prompts	CMA	entropy	44.9	36.8
Ours	prompts	CMA	Eqn. (5)	65.4	3.3

the fitness function boosts performance a lot when with SGD

Computational Efficiency

Table 8. Comparisons w.r.t. computation complexity. **FP/BP is short forward/backward propagation.** #FP and #BP are numbers counted for processing a single sample. Accuracy (%) and ECE (%) are average results on ImageNet-C (level 5) with ViT-Base. The Wall-Clock Time (seconds) and Memory Usage (MB) are measured for processing 50,000 images of ImageNet-C on a single RTX 3090 GPU. K is the population size in CMA and it works well with all $K \in [2, 28]$ and $K \in \mathbb{N}^+$, as shown in Figure 2 (a).

Method	BP	#FP	#BP	Average Acc.	Average ECE	Run Time (seconds)	Memory (MB)
NoAdapt	✗	1	0	55.5	10.5	119	819
T3A	✗	1	0	56.9	26.8	235	957
MEMO	✔	65	64	57.2	9.9	40,428	11,058
TENT	✔	1	1	59.6	18.5	259	5,165
SAR	✔	[1, 2]	[0, 2]	62.7	7.0	517	5,166
CoTTA	✔	3or35	1	61.7	6.5	964	16,836
Act. Shifting	✗	1	0	59.1	12.7	120	821
FOA (K=2)	✗	2	0	59.6	9.7	255	830
FOA (K=4)	✗	4	0	60.9	5.8	497	830
FOA (K=6)	✗	6	0	62.7	4.6	740	830
FOA (K=28)	✗	28	0	66.3	3.2	3,386	832

Table 7. Comparison w.r.t. run-time memory (MB) usage. Results obtained via ViT-Base (32/8-bit) on ImageNet-C (Gaussian, level 5). FOA-I V1/V2 denote storing features/images for interval update under batch size (BS) 1. The memory for 8-bit ViT is an ideal estimation by 0.25× memory of 32-bit ViT per Liu et al. (2021b).

	BP	BS=1	BS=4	BS=8	BS=16	BS=32	BS=64
NoAdapt	✗	346	369	398	458	579	819
TENT	✔	426	648	948	1,550	2,756	5,165
CoTTA	✔	1,792	2,312	3,282	5,226	9,105	16,836
FOA	✗	–	372	402	464	587	832
FOA (8-bit)	✗	–	93	100	116	147	208
		BS=1, but update prompt every I samples					
		I=1	I=4	I=8	I=16	I=32	I=64
FOA-I V1	✗	–	352	356	373	406	473
FOA-I V1 (8-bit)	✗	–	88	89	93	102	118
FOA-I V2	✗	–	351	353	358	368	388
FOA-I V2 (8-bit)	✗	–	88	88	89	92	97

- Act Shifting vs. NoAdapt: much better acc, same efficiency
- FOA (K=2) vs. TENT: same acc, but no BP needed, lower memory
- FOA (K=6) vs. CoTTA: better acc, higher efficiency and lower memory

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Thank You for Your Attention!

Code: https://github.com/mr-eggplant/FOA

Presenter: Shuaicheng Niu (牛帅程)
shuaicheng.niu@ntu.edu.sg

International Conference on Machine Learning (ICML) 2024

AI TIME Artificial Intelligence