Towards Robust and Efficient Cloud-Edge Elastic Model Adaptation via Selective Entropy Distillation

Yaofo Chen[1,2*], Shuaicheng Niu[3*], Yaowei Wang[2*], Shoukai Xu[1], Hengjie Song[1], Mingkui Tan[1,4,5†]

Presented by Yaofo Chen

South China University of Technology[1] Pengcheng Laboratory[2]
Nanyang Technological University[3]
Key Laboratory of Big Data and Intelligent Robot, Ministry of Education[4]
Pazhou Laboratory[5]

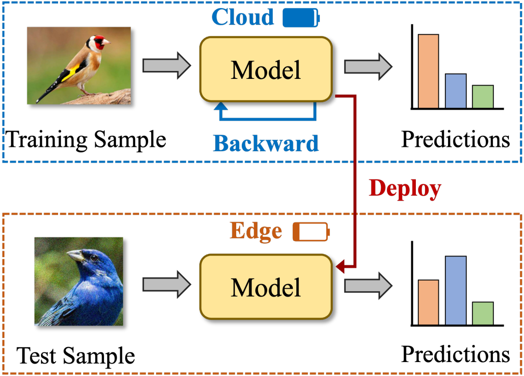
June 9, 2024

Outline

Outline

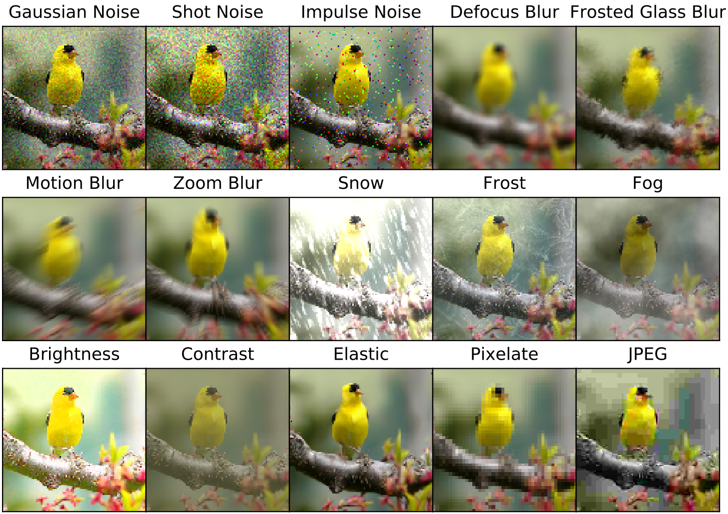
Background

In real-world applications, the typical DNN deployment pipeline involves **training a large model on a cloud** server and then distilling it into a smaller version for **deployment on edge devices**.



Background

In edge environments, **dynamic changes** and **differing test distributions** from training reduce model performance.



Challenges of Cloud-Edge Model Deployment

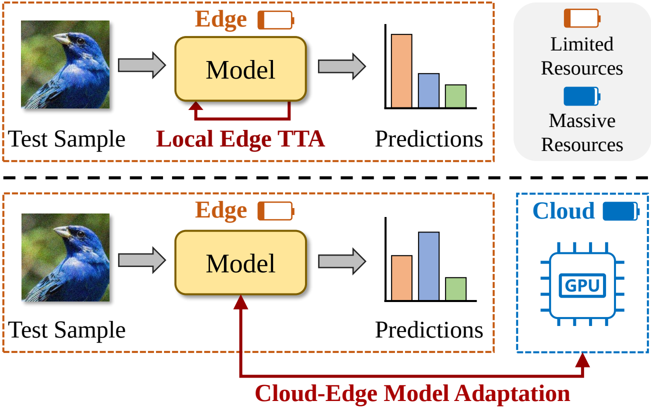
Challenges

- Due to the **high cost of adaptation** for resource-limited edge devices, the model usually **remains fixed**.
- It is **difficult** for the fixed model to **handle distribution-shifted data**.

Outline

Motivation

- **Local Test-time Adapttion** (upper): It locally performs adaptation only in the edge with **limited resources**.
- **Cloud-edge Style Adaptation** (lower): It conducts model adaptation **more efficiently** in the edge, which **offloads the heavy adaptation workloads to the cloud** with massive resources.



Challenges of Cloud-Edge Style Adaptation

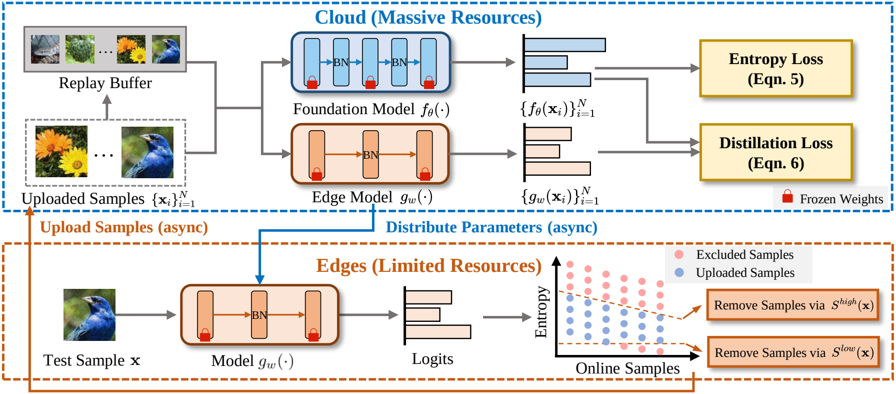
Challenges

- The **data communication cost may be heavy** if uploading all samples to the cloud.
- It is unclear how to **exploit the massive resource in the cloud** to enhance the performance.

Outline

Overview

- **Edge Side**: To reduce the communication cost, the edge uploads only reliable and informative samples to the cloud.
- **Cloud Side**: 1) adapt the foundation model $f_\theta(\cdot)$ and store uploaded samples into a replay buffer; 2) adapt the edge model $g_w(\cdot)$ by distilling from $f_\theta(\cdot)$ with samples from the edge and the replay buffer.



Dynamic Entropy-based Sample Filtration

The previous method (ETA, ICML 2022) only filters out test samples based on a **static and pre-determined** threshold. It suffers **two limitations**:

- Only a part of the negatively impacting samples can be excluded.
- Adaptation with extremely low-entropy samples is unnecessary.

To address the above issues, we propose to

- Dynamically exclude **unreliable** (high entropy) samples by adaptively adjusting the threshold based on the entropy of current samples.
- Exclude the **low-informative** (low entropy) samples.

We devise a binary score $S(\mathbf{x})$ to indicate whether a sample $\mathbf{x}$ should be uploaded ($S(\mathbf{x})=\mathbf{1}$ indicates uploading and $S(\mathbf{x})=\mathbf{0}$ indicates removal).

Dynamic Identification on Unreliable Samples

We exploit a entropy threshold $E_{\max}$ to filter out the high entropy samples

$$S^{high}(\mathbf{x}) = \mathbb{1}_{\{E(\mathbf{x};w)<E_{\max}\}}(\mathbf{x}), \tag{1}$$

where $E(\mathbf{x}; w)$ denotes the entropy of the outputs $g_w(\mathbf{x})$ for the sample $\mathbf{x}$.

Then, we seek to **lower** $E_{\max}$ **according to the average entropy of the test samples after every adaptation batch**. In the adaptation batch t, the entropy threshold $E^t_{\max}$ can be calculated by

$$E^t_{\max} \leftarrow \lambda \times E^{t-1}_{\max} \times \frac{E^t_{\text{avg}}}{E^{t-1}_{\text{avg}}}, \tag{2}$$

where E^t_{avg} denotes the average entropy of all test samples in past t batches, λ is a hyper-parameters.

Identification on Low-Informative Samples

Following the similar scheme above, we employ a threshold $E_{\min}$ to **discard samples with entropy lower** than $E_{\min}$. Formally, $S^{low}(\mathbf{x})$ can be written as

$$S^{low}(\mathbf{x}) = \mathbb{1}_{\{E(\mathbf{x};\theta) > E_{\min}\}}(\mathbf{x}). \tag{3}$$

The overall binary score $S(\mathbf{x})$ can be calculated by

$$S(\mathbf{x}) = S^{high}(\mathbf{x}) \cdot S^{low}(\mathbf{x}). \tag{4}$$

Replay-based Knowledge Distillation

Step 1: upon receiving uploaded samples $\hat{\mathcal{X}}=\{\mathbf{x}_i\}_{i=1}^{N}$, we put them into a **replay buffer** $\mathcal{B}=\mathcal{B}\cup\hat{\mathcal{X}}$.

Step 2: we adapt $f_\theta(\cdot)$ by minimizing the weighted entropy loss

$$\min_{\tilde{\theta}} H(\mathbf{x}) \sum_{y\in\mathcal{C}} f_\theta(y|\mathbf{x}) \log f_\theta(y|\mathbf{x}), \tag{5}$$

where $H(\mathbf{x})=1/\exp\left(E(\mathbf{x};\theta)-E_{\text{max}}\right)$ and $\mathcal{C}$ denotes the output space.

Step 3: we optimize $g_w(\cdot)$ by employing both **entropy minimization** and knowledge distillation as follows,

$$\min_{\tilde{w}} H(\mathbf{x})[\alpha\mathcal{L}_{KL}(g_w(\mathbf{x}), f_\theta(\mathbf{x})) + \beta\mathcal{L}_{CE}(g_w(\mathbf{x}), \hat{y}) + \mathcal{L}_{ENT}(g_w(\mathbf{x}))], \tag{6}$$

where α and β are factors for balancing the losses.

Outline

Comparisons on CNN-based Models

Our CEMA **achieves higher adaptation accuracy** than state-of-the-art adaptation methods on the ImageNet-C dataset.

	Noise			Blur				Weather				Digital				
Severity Level=3	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet18 (baseline)	21.6	19.9	18.7	29.9	15.8	28.7	27.6	27.6	23.8	35.5	62.7	38.1	51.8	41.6	53.0	33.1
• BN Adaptation†	42.3	39.8	40.0	37.5	31.4	45.1	44.3	40.8	36.2	53.9	65.0	58.2	60.2	58.0	57.7	47.4
• ONDA†	40.0	38.9	37.5	29.5	27.5	43.8	43.9	40.2	35.2	54.6	65.1	56.1	59.7	58.6	57.6	45.9
• LAME†	20.6	18.9	17.2	29.5	14.7	28.3	26.9	26.8	23.2	34.9	62.4	37.5	51.3	41.1	52.5	32.4
• PL	48.1	48.0	46.1	41.1	39.7	51.3	49.9	47.3	39.8	58.6	64.9	59.2	62.5	60.8	59.4	51.8
• Tent	47.2	47.1	45.1	40.0	38.2	50.4	49.4	46.7	40.1	58.1	64.9	59.0	62.5	60.5	59.2	51.2
• CoTTA	42.0	40.7	39.8	30.3	30.1	46.3	46.1	41.9	36.5	56.2	64.9	58.0	60.2	59.3	58.1	47.4
• ETA	50.1	50.2	48.6	44.0	42.7	52.9	51.4	49.9	43.5	59.5	**65.2**	60.9	**62.9**	**61.6**	**59.9**	53.5
• CEMA (Ours)	**51.1**	**51.2**	**49.8**	**45.2**	**44.1**	**53.7**	**52.0**	**50.8**	**44.2**	**60.1**	65.0	**61.1**	**62.9**	**61.6**	59.8	**54.2**
Severity Level=5	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet18 (baseline)	1.5	2.3	1.5	11.4	8.7	11.1	17.6	10.6	16.2	14.0	51.5	3.4	16.5	23.3	30.7	14.7
• BN Adaptation†	16.6	16.2	17.3	18.6	18.2	25.9	34.7	28.4	29.8	41.2	58.5	22.2	40.1	45.3	38.0	30.1
• ONDA†	13.7	15.0	14.1	12.3	13.2	23.7	34.2	29.4	28.6	40.9	58.5	12.3	39.3	44.6	37.5	27.8
• LAME†	0.9	1.1	0.6	11.2	8.2	10.8	17.0	8.7	15.6	12.4	51.1	3.3	14.9	22.5	30.1	13.9
• PL	24.8	26.8	24.6	20.3	21.3	33.6	41.8	39.0	32.4	49.9	59.5	11.4	47.9	51.5	47.0	35.4
• Tent	22.8	25.0	23.2	20.1	21.1	32.4	41.0	37.8	33.5	48.9	59.3	18.0	46.9	50.6	45.9	35.1
• CoTTA	15.2	16.2	15.7	11.8	14.9	26.9	36.9	31.2	29.9	43.6	59.2	17.0	40.9	47.2	39.3	29.7
• ETA	26.8	29.7	27.6	22.6	22.7	37.1	44.0	42.4	37.6	51.6	60.1	26.1	49.8	53.3	48.5	38.7
• CEMA (Ours)	**29.8**	**32.2**	**30.3**	**25.3**	**26.8**	**39.3**	**45.3**	**43.7**	**38.7**	**52.8**	**60.1**	**32.9**	**50.8**	**54.0**	**49.3**	**40.8**

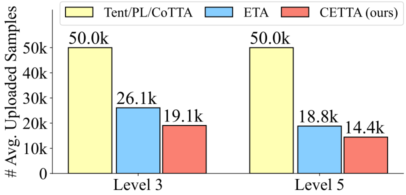
Comparisons on Transformer-based Models

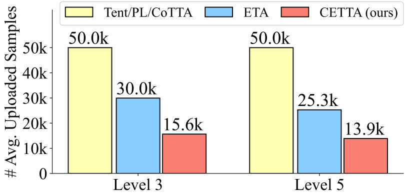
Our CEMA **outperforms state-of-the-art** adaptation methods in terms of adaptation performance on the ImageNet-C dataset.

	Noise			Blur				Weather				Digital				
Severity Level=3	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
DeiT-tiny (baseline)	49.1	48.0	48.6	38.1	20.5	43.8	31.6	44.9	44.2	47.0	66.7	60.6	55.5	47.6	56.8	46.9
• LAME†	48.9	47.7	48.3	37.5	19.2	43.5	30.8	44.3	43.8	46.2	66.4	60.3	55.1	47.0	56.4	46.3
• PL	52.7	52.8	53.1	46.1	35.6	53.3	42.4	49.8	46.9	58.4	67.9	63.7	62.3	58.4	59.6	53.5
• Tent	53.1	53.1	53.4	47.9	41.0	54.7	46.3	51.5	48.2	60.0	68.1	64.1	63.8	60.1	60.7	55.1
• CoTTA	49.8	48.8	49.4	39.0	20.9	45.1	32.1	46.0	45.4	49.0	67.0	61.6	56.5	49.0	57.5	47.8
• ETA	54.1	54.2	54.2	49.4	47.0	56.1	51.7	53.7	51.0	**61.5**	68.1	**64.6**	64.7	62.4	62.0	57.0
• CEMA (Ours)	**55.0**	**55.1**	**55.1**	**50.5**	**48.5**	**57.1**	**52.9**	**55.4**	**51.8**	60.2	**68.4**	64.3	**65.5**	**63.4**	**63.0**	**57.7**
Severity Level=5	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
DeiT-tiny (baseline)	17.0	18.2	17.4	19.2	12.6	22.9	20.9	32.6	37.6	32.9	59.6	23.9	23.5	10.3	38.5	25.8
• LAME†	16.5	17.9	17.0	18.6	11.4	22.4	19.9	31.4	37.1	29.6	59.3	23.4	21.3	10.1	38.1	24.9
• PL	1.0	2.3	1.1	17.8	2.4	35.9	3.6	9.4	15.2	0.8	62.8	38.6	3.9	35.9	46.2	18.5
• Tent	4.1	13.3	13.6	27.1	1.6	38.7	3.4	11.7	14.6	0.8	63.2	**41.3**	2.4	44.1	47.8	21.8
• CoTTA	17.6	18.8	18.1	19.7	12.7	23.9	21.0	33.7	38.7	34.8	60.4	24.4	24.1	10.6	39.3	26.5
• ETA	32.1	33.7	33.4	33.8	**35.0**	42.9	**43.4**	45.9	46.0	53.2	63.9	33.9	**50.2**	**50.6**	51.0	43.1
• CEMA (Ours)	**34.4**	**36.7**	**36.2**	**35.8**	34.4	**44.8**	43.0	**48.0**	**46.8**	**54.6**	**64.0**	37.0	49.9	50.1	**52.8**	**44.5**

Comparisons of #Uploading Samples

Our CEMA **requires much lower uploading samples** than existing adaptation methods, such as Tent, PL, and ETA, on both CNN- (left figure) and Transformer-based (right figure) models.





Outline

Conclusions

- We establish a **Cloud-Edge Elastic Model Adaptation (CEMA)** paradigm designed for efficient collaborative model adaptation. Our CEMA is a **general paradigm** that is applicable to online adapt edge models to new dynamically changing environments.
- We **reduce communication costs** by devising entropy-based criteria for excluding **unreliable and low-informative samples** from being uploaded. Experimental results show CEMA **lowers 60% communication cost** than SOTAs on ImageNet-C.
- We improve the adaptation performance of the edge model by performing a **replay-based entropy distillation**, which minimizes prediction entropy and the KL divergence between the edge model and the foundation model using a sample replay strategy.

Closing Remarks

Thank you for your attention. Scan for more details.

PDF

Code