NEURAL INFORMATION PROCESSING SYSTEMS

SNAP

Low-Latency Test-Time Adaptation with Sparse Updates





Hyeongheon Cha Dong Min Kim Hye Won Chung Taesik Gong* Sung-Ju Lee*

* Corresponding authors

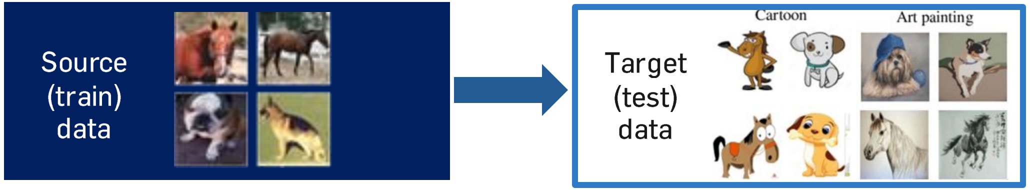
KAIST UNIST ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

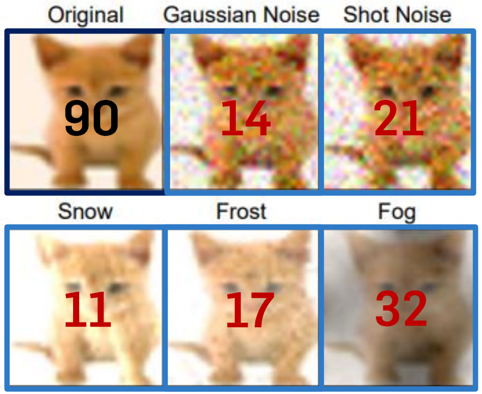
https://nmsl.kaist.ac.kr/projects/snap

Motivation

Test-Time Adaptation

Deep learning models often suffer from **domain shifts**

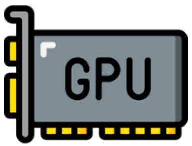


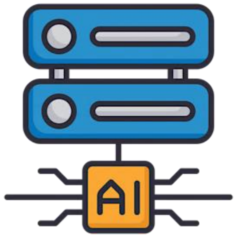


Test-time adaptation (TTA) adapts models **after deployment**,

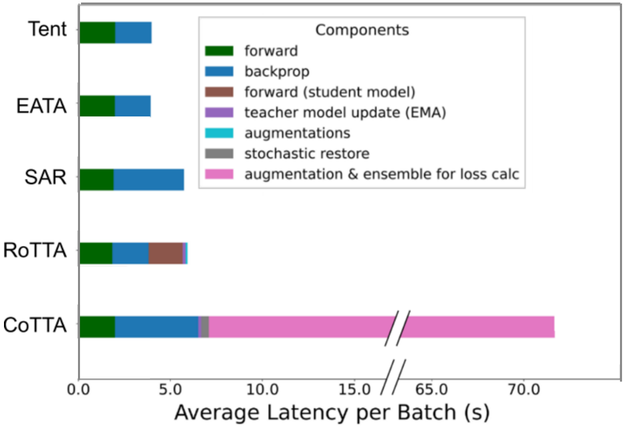
without **any source** or **labeled data.**

Motivation





SOTA TTA algorithms have been designed and evaluated mainly on GPU servers, focusing on **Improving accuracy in dynamic scenarios.**

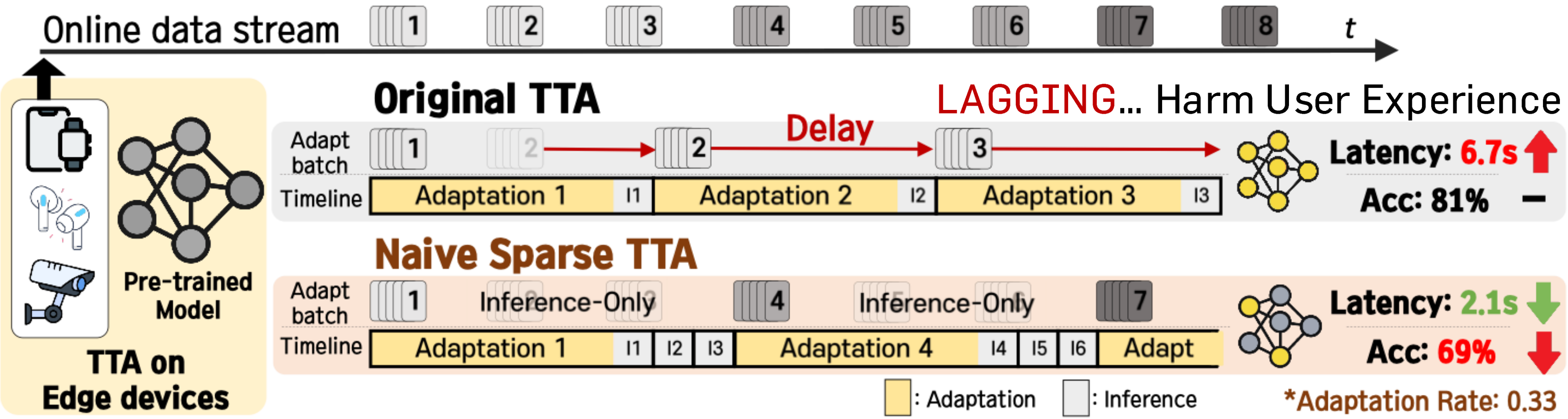


Inference-only	22.13
Tent [1]	73.66
EATA [2]	75.82
SAR [3]	73.52
RoTTA [4]	66.54
CoTTA [5]	71.95

Model Accuracy
(Tested on GPU)

[1] Wang, Dequan, et al. "Tent: Fully test-time adaptation by entropy minimization." International Conference on Learning Representations. ICLR, 2021.
[2] Niu, Shuaicheng, et al. "Efficient test-time model adaptation without forgetting." International Conference on Machine Learning. ICML, 2022.
[3] Niu, Shuaicheng, et al. "Towards stable test-time adaptation in dynamic wild world." International Conference on Learning Representations. ICLR, 2023.
[4] Yuan, Longhui, Binhui Xie, and Shuang Li. "Robust test-time adaptation in dynamic scenarios." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, 2023.
[5] Wang, Qin, et al. "Continual test-time domain adaptation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, 2022.

TTA with Sparse Update

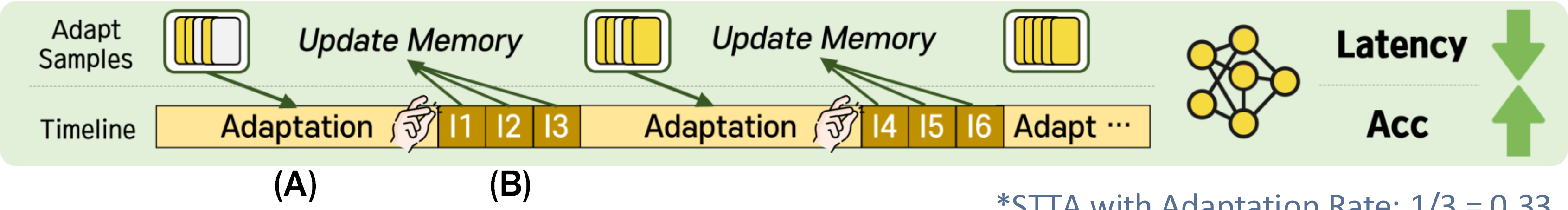


- ✓ Events unfold continuously, leading the model **to miss incoming samples** while processing previous ones.
- ✓ Limited number of **samples & updates significantly** reduces adaptation accuracy.

Practical.

SNAP: Sparse Network Adaptation for Practical TTA

Goal - Achieve **significantly lower latency** than original TTA while maintaining **comparable** or **superior** accuracy.



*STTA with Adaptation Rate: 1/3 = 0.33

Frequency of updates and determines how sparsely adaptation occurs

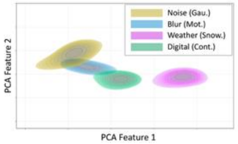
(A) Class and Domain Representative Memory for *extremely* efficient (e.g.,1%) sampling.

(B) Inference-only Batch aware **Memory Normalization** to correct normalization by blending stable, representative statistics from memory with recent inference batch data.

(A) CnDRM: Class and Domain Representative Memory

When the sampling ratio is low (<0.5)

- selecting easy and class-representative samples becomes more effective. [1]
- distance from the class center significantly impacts performance, with samples closer to the center being particularly effective in scenarios with high label noise. [2]

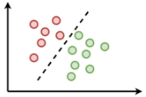


Criteria 1: **Class** representation

① Filter-out low-confidence samples.

"typically located near decision boundaries (hard) and are more likely to carry incorrect pseudo-labels"

$$C(\mathbf{x}) = \max_{y \in \mathcal{Y}} p(y|\mathbf{x};\Theta)$$



② Prediction-balanced manner.

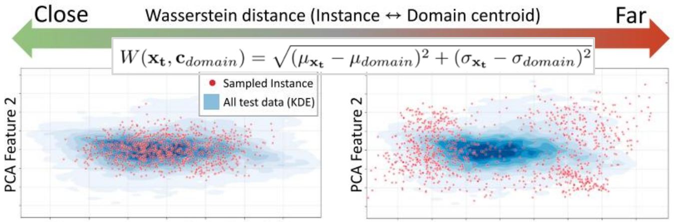
"prevents bias towards certain classes when only a few samples are available for single adaptation"

#Cat  #Dog

Samples that are diverse and reliable,
even without access to ground-truth labels.

Criteria 2: **Domain** representation

Supervised DL's class-centroid ≃ Unsupervised DA's domain-centroid



Accuracy: 26.65 % **Accuracy: 18.52 %**

✓ Early layers in DL models tend to retain domain-specific features[3,4,5].

✓ Utilize the hidden features statistics (mean and variance) of early layers to identify domain-representative samples.

$$\mathbf{c}_{domain} \quad \begin{array}{l} \mu_{domain} \leftarrow (1-\beta)\mu_{domain} + \beta\mu_t \\ \sigma^2_{domain} \leftarrow (1-\beta)\sigma^2_{domain} + \beta\sigma^2_t \end{array}$$

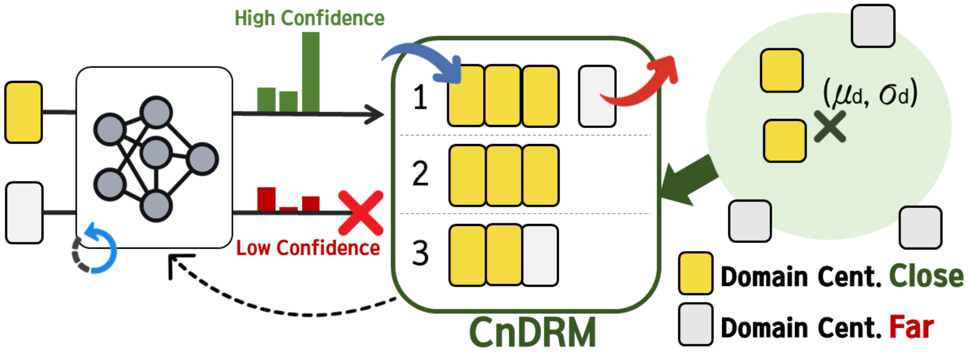
[1] Hoyong Choi, et al. "BWS: Best Window Selection Based on Sample Scores for Data Pruning across Broad Ranges." ICLR, 2024.
[2] Xiaobo Xia, et al. "Moderate coreset: A universal method of data selection for real-world data-efficient deep learning." ICLR, 2022.
[3] Matthew D Zeiler ea al. "Visualizing and understanding convolutional networks.", ECCV, 2014.
[4] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks." NeurIPS, 2018.
[5] Mattia Segu et al. "Batch normalization embeddings for deep domain generalization." Pattern Recognition, 2023

(A) CnDRM: Class and Domain Representative Memory

Adapt via CnDRM

Sparse adaptation via

Class-Domain Representative Memory



Algorithm 1 Class and Domain Representative Memory (CnDRM) Management

Require: test data stream x_t, memory M with capacity N, confidence threshold τ_{conf}, adaptation rate $1/k$

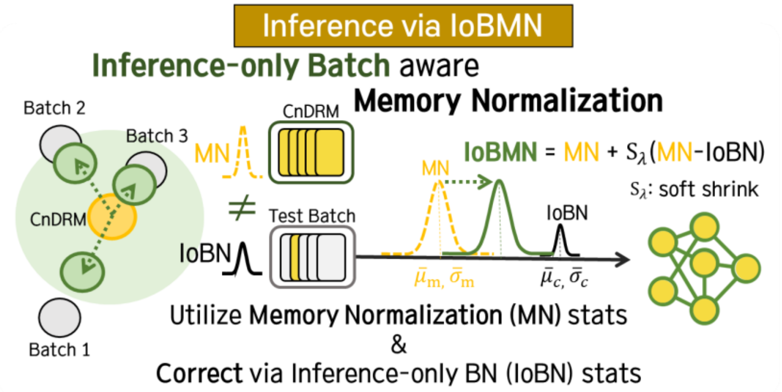
for batch $b \in \{1, \ldots, B\}$ **do**
  $\hat{Y}_b \leftarrow f(b; \Theta)$
  for each sample x_t in batch b **do**
    $\hat{y}_t \leftarrow \hat{Y}_b[t]$
    confidence $\leftarrow C(x_t; \Theta)$
    $\mathbf{c}_t(\mu_{\mathbf{x_t}}, \sigma_{\mathbf{x_t}}) \leftarrow$ mean & variance of early feature
    $w_{x_t} \leftarrow W(x_t, \mathbf{c}_{domain})$
    if confidence $> \tau_{conf}$ **then**
      Add $\mathbf{s}_t(x_t, \hat{y}_t, c_t, w_{x_t})$ to M ▷ Add class-representative samples
      if $|M| > N$ **then**
        $L^* \leftarrow$ class with most samples in M
        if $\hat{y}_t \notin L^*$ **then** ▷ Remove domain-centroid farthest sample
          $\mathbf{s}_{farthest} \leftarrow \arg\max_{\mathbf{s}_i \in M \wedge \hat{y}_i \in L^*} w_{x_i}$
        else
          $\mathbf{s}_{farthest} \leftarrow \arg\max_{\mathbf{s}_i \in M \wedge \hat{y}_i = \hat{y}_t} w_{x_i}$
        Remove $\mathbf{s}_{farthest}$ from M
  $\mathbf{c}_{domain} \leftarrow (1-\beta)\mathbf{c}_{domain} + \beta \mathbf{c}_t$ ▷ Update domain-centroid
  Recalculate w_{s_i} for all $\mathbf{s}_i$ in M
  if $b \mod k == 0$ **then** ▷ Adaptation occurs every k batches
    Update model Θ using samples in M

- Core component of SNAP that addresses the challenges of **efficient data sampling** for Sparse TTA.
- Adaptation rate directly impacts the number of samples used for model update, necessitating a careful sampling strategy to optimize performance with minimal data.
- Given this limited sampling ratio, CnDRM selects **the most** *class and domain-representative* (managed by distance **ranking**) samples to maintain model performance while minimizing overhead.

(B) IoBMN: Inference-only Batch aware Memory Normalization

After the model update via CnDRM,

- Have to follow the data distribution shift only through the normalization on Inference-only batches (Skip stages).
- Maintaining robust performance becomes challenging as the limited memory statistics may not fully align with each subsequent inference batches.
- This lead to a *potential mismatch between the model's stored statistics and the current data distribution*.
- Traditional normalization methods, which solely rely on test batches' statistics, struggle to address these *MISMATCH*.



① Basing normalization on class-domain representative statistics
② Dynamically adjusting statistics with recent inference data.

⇒ IoBMN efficiently corrects for potential distributional shifts, ensuring both robustness and adaptability in STTA conditions.

Ablation Study

CIFAR10C

Methods	Tent	CoTTA	EATA	SAR	RoTTA
CnDRM	77.46	77.69	77.17	76.85	75.64
CnDRM+EMA	78.02	72.19	77.05	76.84	76.18
CnDRM+IoBMN	**78.95**	**78.83**	**78.61**	**78.06**	**77.07**

CIFAR100C

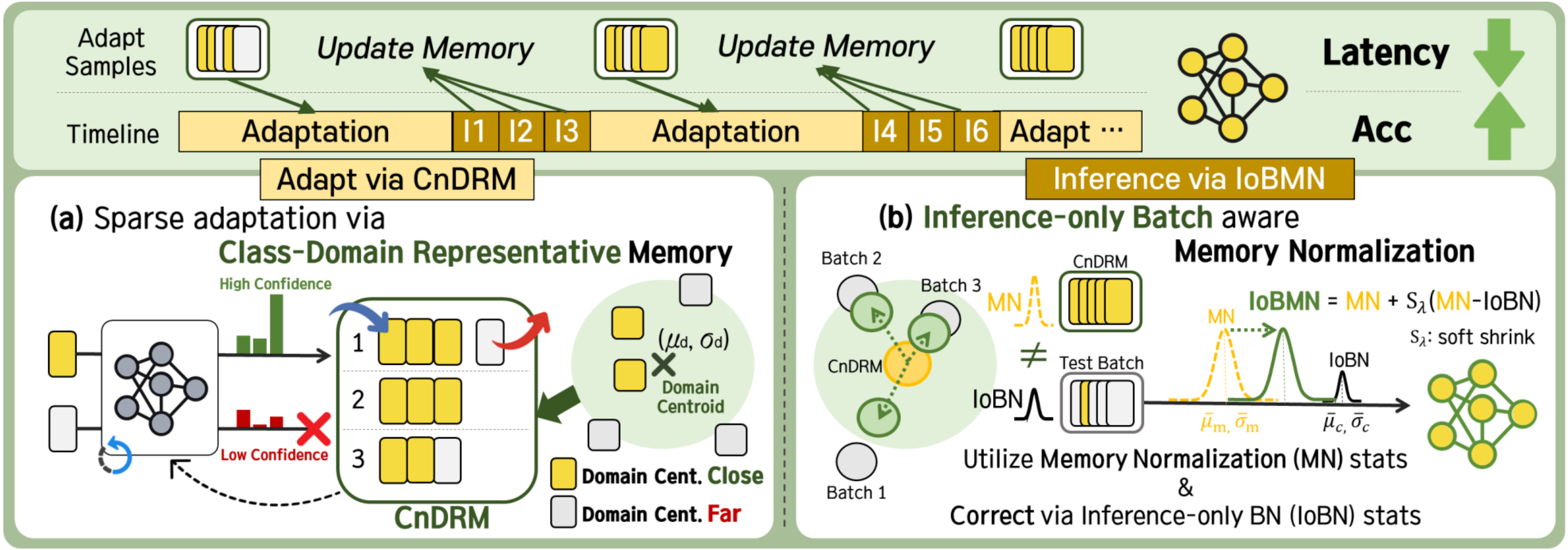
Methods	Tent	CoTTA	EATA	SAR	RoTTA
CnDRM	54.46	50.06	51.41	55.24	50.47
CnDRM+EMA	54.36	41.63	50.21	54.84	49.95
CnDRM+IoDMN	**55.84**	**50.52**	**52.35**	**55.76**	**51.33**

✓ Combination of CnDRM and IoBMN (inference using memory's double representative statistics corrected to match the test batch) consistently yields the highest accuracy.

✓ This trend is observed across all evaluated adaptation rates, suggesting that both components certainly contribute to enhancing performance.

Sparse Network Adaptation for Practical TTA

SNAP

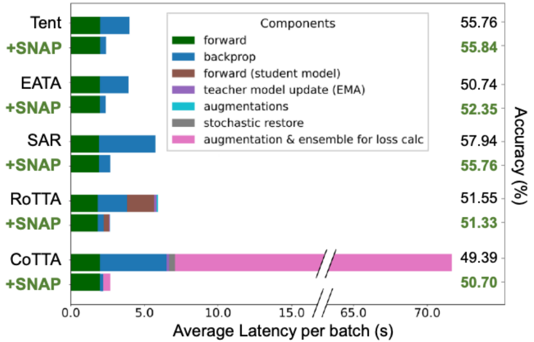


Evaluation

SNAP mitigates accuracy drops of sparse TTA

while retaining its latency benefits,

thereby **boosting efficiency**.



* Tested on RPi4, ResNet18, CIFAR100C

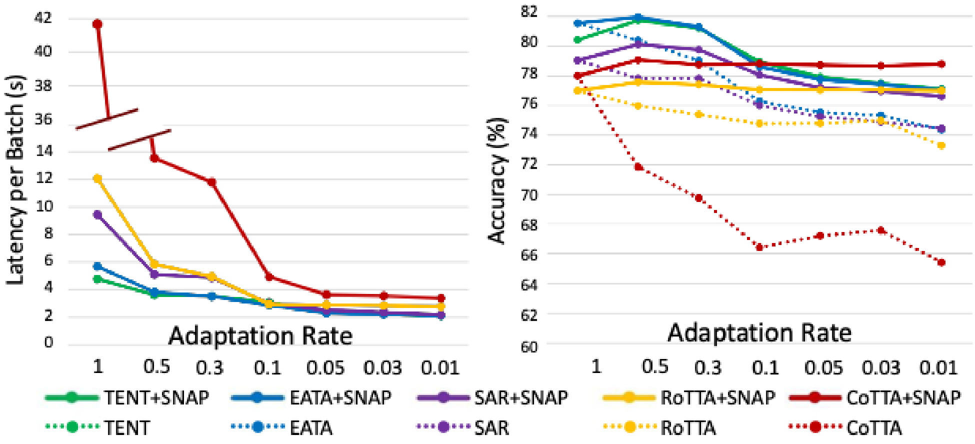
	Latency	Accuracy (Δ)
Tent [1]+SNAP	2.20 sec (-44.0%)	78.95 (-1.48)
EATA [2]+SNAP	2.18 sec (-44.6%)	78.61 (-2.95)
SAR [3]+SNAP	2.30 sec (-60.1%)	78.06 (-0.99)
RoTTA [4]+SNAP	2.25 sec (-62.0%)	77.07 (+0.07)
CoTTA [5]+SNAP	8.96 sec (-87.5%)	78.83 (+0.83)

* Tested on RPi4, ResNet18, CIFAR10C

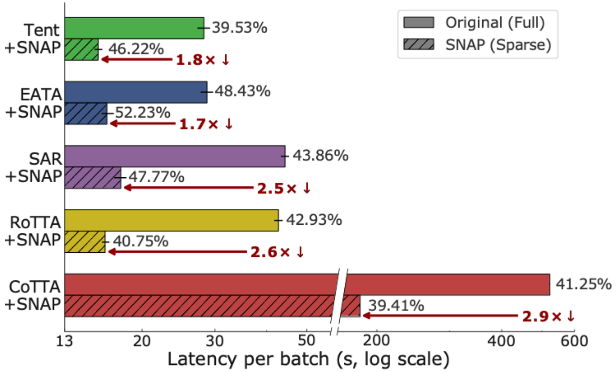
[1] Wang, Dequan, et al. "Tent: Fully test-time adaptation by entropy minimization." International Conference on Learning Representations. ICLR, 2021.
[2] Niu, Shuaicheng, et al. "Efficient test-time model adaptation without forgetting." International Conference on Machine Learning. ICML, 2022.
[3] Niu, Shuaicheng, et al. "Towards stable test-time adaptation in dynamic wild world." International Conference on Learning Representations. ICLR, 2023.
[4] Yuan, Longhui, Binhui Xie, and Shuang Li. "Robust test-time adaptation in dynamic scenarios." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, 2023.
[5] Wang, Qin, et al. "Continual test-time domain adaptation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, 2022.

Evaluation

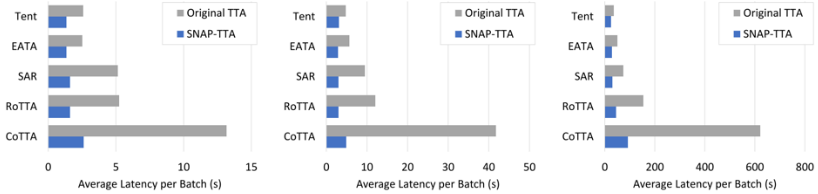
➢ Validation of SNAP across Various Adaptation Rates (0.01 to 0.5)



➢ Validation of SNAP on Vision Transformer (ViT) based Model (Layer Normalization)



➢ Validation across diverse Edge-devices

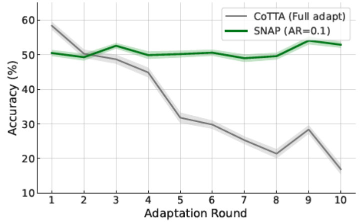


(a) NVIDIA Jetson Nano (b) Raspberry Pi 4 (c) Raspberry Pi Zero 2 W

Evaluation - additionals

- Validation of SNAP on Continuous / Long-term Domain Shift Scenario

AR	Method	Gau.	Shot	Imp.	Def.	Gla.	Mot.	Zoom	Snow	Fro.	Fog	Brit.	Cont.	Elas.	Pix.	JPEG	Avg.
0.1	Tent	24.68 ±0.45	19.65 ±1.27	5.12 ±1.22	0.63 ±0.05	0.43 ±0.02	0.40 ±0.04	0.44 ±0.06	0.41 ±0.03	0.30 ±0.03	0.33 ±0.04	0.42 ±0.05	0.24 ±0.04	0.32 ±0.02	0.31 ±0.05	0.31 ±0.04	3.60 ±0.23
	+ SNAP	**28.71 ±0.66**	**30.60 ±1.82**	**22.91 ±2.25**	**6.13 ±0.90**	**1.62 ±0.20**	**0.87 ±0.13**	**0.88 ±0.07**	**0.64 ±0.08**	**0.64 ±0.06**	**0.66 ±0.05**	**0.75 ±0.01**	**0.44 ±0.05**	**0.60 ±0.08**	**0.63 ±0.07**	**0.61 ±0.07**	**6.45 ±0.43**
	CoTTA	10.99 ±0.40	12.21 ±0.04	11.54 ±0.30	11.28 ±0.13	11.13 ±0.15	22.08 ±0.07	34.80 ±0.18	30.69 ±0.10	29.45 ±0.04	43.87 ±0.19	61.92 ±0.09	12.76 ±0.16	40.03 ±0.13	44.99 ±0.14	36.43 ±0.16	27.61 ±0.15
	+ SNAP	**15.19 ±0.17**	**15.97 ±0.11**	**15.91 ±0.02**	**13.94 ±0.04**	**14.18 ±0.03**	**24.76 ±0.07**	**36.50 ±0.23**	**32.61 ±0.04**	**31.76 ±0.06**	**46.14 ±0.10**	**63.60 ±0.14**	**15.60 ±0.04**	**42.17 ±0.02**	**46.77 ±0.06**	**38.08 ±0.12**	**30.21 ±0.08**



- Memory Overhead of SNAP / Integration of SNAP with Memory-Efficient TTA: MECTA[1]

Methods	Average Mem (MB)		Peak Mem (MB)		Mem Overhead (MB)
	Original TTA	SNAP	Original TTA	SNAP	SNAP - Original
Tent	764.24	**751.35**	**822.93**	828.46	5.52 (0.67%)
CoTTA	1133.52	**1099.64**	**1211.21**	1227.99	16.78 (1.13%)
EATA	816.69	**749.95**	**847.73**	862.51	14.78 (1.74%)
SAR	786.65	**753.69**	**863.77**	865.18	1.41 (0.02%)
RoTTA	933.23	**871.64**	**972.23**	983.94	11.71 (1.20%)

Methods	Accuracy (%)	Max Memory (MB)
Tent	35.21±0.09	6805.26
+MECTA	37.62±0.16	**4620.25 (-32.10%)**
+ SNAP	**39.52**±0.13	4622.12 (-32.08%)
EATA	35.55±0.19	6541.02
+MECTA	41.41±0.37	**4512.38 (-31.01%)**
+ SNAP	**42.86**±0.20	4535.44 (-30.66%)

- Impact of Memory Size on SNAP Performance

Memory Size	**Accuracy (%)**
16 (Base)	26.60 ±0.11
32	28.44 ±0.17
64	28.89 ±0.06
128	28.60 ±0.09

- Robustness in Single-sample (BS=1) adaptation scenario

Method	**Accuracy (%)**
SAR (single-sample)	52.21 ± 0.28
+ STTA	8.06 ± 0.12
+ SNAP	**51.80 ± 0.25**

- Effect of Learning Rate on naïve sparse, SNAP and full adaptation

Learning rate	**Tent**			**CoTTA**			**EATA**		
	Full	**naïve STTA**	**SNAP**	**Full**	**naïve STTA**	**SNAP**	**Full**	**naïve STTA**	**SNAP**
2×10^{-3}	2.31 ±0.10	18.06 ±0.14	27.41 ±0.12	13.31 ±0.08	10.93 ±0.07	14.80 ±0.11	0.36 ±0.03	1.86 ±0.06	9.59 ±0.15
1×10^{-3}	4.54 ±0.11	**25.46** ±0.13	**31.12** ±0.14	13.18 ±0.09	10.93 ±0.07	14.73 ±0.10	1.31 ±0.05	2.86 ±0.08	24.95 ±0.13
5×10^{-4}	10.22 ±0.12	24.71 ±0.14	28.01 ±0.11	13.15 ±0.07	10.92 ±0.06	**15.18** ±0.09	21.96 ±0.12	18.76 ±0.10	**28.09** ±0.11
1×10^{-4}	**27.03** ±0.10	22.00 ±0.12	26.21 ±0.13	13.12 ±0.08	**11.74** ±0.06	15.13 ±0.09	**29.42** ±0.11	**22.43** ±0.10	26.10 ±0.12
5×10^{-5}	26.34 ±0.09	16.72 ±0.13	19.31 ±0.12	**13.34** ±0.08	10.92 ±0.07	14.76 ±0.09	29.37 ±0.11	20.32 ±0.10	23.28 ±0.10

[1] Junyuan Hong, et al. "Mecta: Memory-economic continual test-time model adaptation." International Conference on Learning Representations. ICLR, 2023.

Contributions

SNAP

- ✓ Existing state-of-the-art TTA methods rely on frequent adaptation and high computational cost, making them unsuitable for practical use on edge devices, resulting in a latency-accuracy trade-off.
- ✓ Propose **SNAP**, a sparse TTA framework that significantly reduces adaptation frequency and data usage, delivering latency reductions proportional to adaptation rate, while preserving accuracy.
  - CnDRM identifies key samples that are both class-representative and domain-representative to facilitate adaptation with minimal data.
  - IoBMN leverages representative samples to dynamically refine normalization stats during inference, effectively aligning the model to distribution shifts.
- ✓ Evaluation on **real edge devices** with **five state-of-the-art TTA algorithms**, SNAP **reduces latency by up to 93.12%**, while keeping the **accuracy drop below 3.3%**, even across **adaptation rates ranging from 1% to 50%.**
- ✓ Plug-and-play and low-overhead design of SNAP, offering seamless integration with existing TTA methods and improving efficiency.

For further discussions, please visit our poster or reach out using the contact information.

San Diego Poster Session 2 (Exhibit Hall C,D,E)

Wed 3 Dec 4:30 p.m. PST — 7:30 p.m. PST

Website: https://nmsl.kaist.ac.kr/projects/snap

Code: https://github.com/chahh9808/SNAP

Contact: hyeongheon@kaist.ac.kr